IMS

Failure Analysis Structure Tables (FAST) for Dump Analysis

Version 9

LY37-3204-00

IMS

Failure Analysis Structure Tables (FAST) for Dump Analysis

Version 9

LY37-3204-00

Note

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 545.

First Edition (October 2004)

This edition applies to Version 9 of IMS (product number 5655–J38) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About This Book

This information is available as part of the DB2® Information Management Software Information Center for z/OS® Solutions. To view the information within the DB2 Information Management Software Information Center for z/OS Solutions, go to http://publib.boulder.ibm.com/infocenter/dzichelp. This information is also available in PDF and BookManager formats. To get the most current versions of the PDF and BookManager formats, go to the IMS Library page at www.ibm.com/software/data/ims/library.html. To view or download the PDF of this information on the Web, you must enter a valid IMS customer number in the Web form that appears after you click the PDF icon for this book on the IMS Version 9 Library page.

This book offers IMS system programmers a detailed analysis and explanation of the more common abends. Note that this book does not document all abends; all user abend codes are explained in some detail in *IMS™ Version 9: Messages and Codes, Volume 1*.

This book consists of:

- Chapter 1, “Introduction,” on page 1: overview, definitions, and descriptions of how to use the failure analysis structure tables (FAST).
- Chapter 2, “IMS Failure Analysis Structure Tables (FAST) 1 - 500,” on page 5, Chapter 3, “IMS Failure Analysis Structure Tables (FAST) 501 - 1000,” on page 145, and Chapter 4, “IMS Failure Analysis Structure Tables (FAST) 1001 - 4095,” on page 399: explanations, analyses, possible causes, and APAR processing instructions for each abnormal termination (abend). The abends are in ascending numerical order.
- “Bibliography” on page 549: list of non-IMS publications cited in this book.

With IMS Version 9, you can reorganize HALDB partitions online, either by using the integrated HALDB Online Reorganization function or by using an external product. In this information, the term *HALDB Online Reorganization* refers to the integrated HALDB Online Reorganization function that is part of IMS Version 9, unless otherwise indicated.

For definitions of additional terminology used in this book and references to related information in other books, see the *IMS Version 9: Master Index and Glossary*.

IBM Product Names Used in This Information

In this information, the licensed programs shown in Table 1 are referred to by their short names.

Table 1. Licensed Program Full Names and Short Names

Licensed program full name	Licensed program short name
IBM® Application Recovery Tool for IMS and DB2	Application Recovery Tool
IBM CICS® Transaction Server for OS/390®	CICS
IBM CICS Transaction Server for z/OS	CICS
IBM DB2 Universal Database™	DB2 Universal Database
IBM DB2 Universal Database for z/OS	DB2 UDB for z/OS
IBM Enterprise COBOL for z/OS and OS/390	Enterprise COBOL
IBM Enterprise PL/I for z/OS and OS/390	Enterprise PL/I
IBM High Level Assembler for MVS™ & VM & VSE	High Level Assembler
IBM IMS Advanced ACB Generator	IMS Advanced ACB Generator
IBM IMS Batch Backout Manager	IMS Batch Backout Manager
IBM IMS Batch Terminal Simulator	IMS Batch Terminal Simulator

Table 1. Licensed Program Full Names and Short Names (continued)

Licensed program full name	Licensed program short name
IBM IMS Buffer Pool Analyzer	IMS Buffer Pool Analyzer
IBM IMS Command Control Facility for z/OS	IMS Command Control Facility
IBM IMS Connect for z/OS	IMS Connect
IBM IMS Connector for Java™	IMS Connector for Java
IBM IMS Database Control Suite	IMS Database Control Suite
IBM IMS Database Recovery Facility for z/OS	IMS Database Recovery Facility
IBM IMS Database Repair Facility	IMS Database Repair Facility
IBM IMS DataPropagator™ for z/OS	IMS DataPropagator
IBM IMS DEDB Fast Recovery	IMS DEDB Fast Recovery
IBM IMS Extended Terminal Option Support	IMS ETO Support
IBM IMS Fast Path Basic Tools	IMS Fast Path Basic Tools
IBM IMS Fast Path Online Tools	IMS Fast Path Online Tools
IBM IMS Hardware Data Compression-Extended	IMS Hardware Data Compression-Extended
IBM IMS High Availability Large Database (HALDB) Conversion Aid for z/OS	IBM IMS HALDB Conversion Aid
IBM IMS High Performance Change Accumulation Utility for z/OS	IMS High Performance Change Accumulation Utility
IBM IMS High Performance Load for z/OS	IMS HP Load
IBM IMS High Performance Pointer Checker for OS/390	IMS HP Pointer Checker
IBM IMS High Performance Prefix Resolution for z/OS	IMS HP Prefix Resolution
IBM Tivoli® NetView® for z/OS	Tivoli NetView for z/OS
IBM WebSphere® Application Server for z/OS and OS/390	WebSphere Application Server for z/OS
IBM WebSphere MQ for z/OS	WebSphere MQ
IBM WebSphere Studio Application Developer Integration Edition	WebSphere Studio
IBM z/OS	z/OS

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:

- Go to the IMS Library page at www.ibm.com/software/data/ims/library.html and click the Library Feedback link, where you can enter and submit comments.
- Send your comments by e-mail to imspubs@us.ibm.com. Be sure to include the title, the part number of the title, the version of IMS, and, if applicable, the specific location of the text on which you are commenting (for example, a page number in the PDF or a heading in the Information Center).

Summary of Changes

Changes to This Book for IMS Version 9

This edition contains new and changed technical information and editorial changes for IMS Version 9.

Changed, Added, and Deleted Abends

The sections below list the abends that were either changed, added, or deleted for IMS Version 9:

Added Abends

These abends were added for IMS Version 9:

- ABENDU0369
- ABENDU0830
- ABENDU2496
- ABENDU3056
- ABENDU3476
- ABENDU3477

Changed Abends

These abends were changed for IMS Version 9:

- ABENDU0008
- ABENDU0009
- ABENDU0010
- ABENDU0012
- ABENDU0015
- ABENDU0102
- ABENDU0107
- ABENDU0166
- ABENDU0231
- ABENDU0437
- ABENDU0711
- ABENDU0820
- ABENDU0832
- ABENDU0833
- ABENDU0880
- ABENDU1026

Deleted Abends

These abends were deleted for IMS Version 9:

- ABENDU0118
- ABENDU0475
- ABENDU3999

Library Changes for IMS Version 9

Changes to the IMS Library for IMS Version 9 include the addition of one title, a change of one title, organizational changes, and a major terminology change. Changes are indicated by a vertical bar (|) to the left of the changed text.

The IMS Version 9 information is now available in the DB2 Information Management Software Information Center for z/OS Solutions, which is available at http://publib.boulder.ibm.com/infocenter/dzichelp. The DB2 Information Management Software Information Center for z/OS Solutions provides a graphical user interface for centralized access to the product information for IMS, IMS Tools, DB2 Universal Database (UDB) for z/OS, DB2 Tools, and DB2 Query Management Facility (QMF™).

New and Revised Titles

The following list details the major changes to the IMS Version 9 library:

- *IMS Version 9: IMS Connect Guide and Reference*

  The library includes new information: *IMS Version 9: IMS Connect Guide and Reference*. This information is available in softcopy format only, as part of the DB2 Information Management Software Information Center for z/OS Solutions, and in PDF and BookManager® formats.

  IMS Version 9 provides an integrated IMS Connect function, which offers a functional replacement for the IMS Connect tool (program number 5655-K52). In this information, the term *IMS Connect* refers to the integrated IMS Connect function that is part of IMS Version 9, unless otherwise indicated.
- The information formerly titled *IMS Version 8: IMS Java User's Guide* is now titled *IMS Version 9: IMS Java Guide and Reference*. This information is available in softcopy format only, as part of the DB2 Information Management Software Information Center for z/OS Solutions, and in PDF and BookManager formats.
- To complement the IMS Version 9 library, a new book, *An Introduction to IMS* by Dean H. Meltz, Rick Long, Mark Harrington, Robert Hain, and Geoff Nicholls (ISBN # 0-13-185671-5), is available starting February 2005 from IBM Press. Go to the IMS Web site at www.ibm.com/ims for details.

Organizational Changes

Organization changes to the IMS Version 9 library include changes to:

- *IMS Version 9: IMS Java Guide and Reference*
- *IMS Version 9: Messages and Codes, Volume 1*
- *IMS Version 9: Utilities Reference: System*

The chapter titled ″DLIModel Utility″ has moved from *IMS Version 9: IMS Java Guide and Reference* to *IMS Version 9: Utilities Reference: System*.

The DLIModel utility messages that were in *IMS Version 9: IMS Java Guide and Reference* have moved to *IMS Version 9: Messages and Codes, Volume 1*.

Terminology Changes

IMS Version 9 introduces new terminology for IMS commands:

type-1 command
: A command, generally preceded by a leading slash character, that can be entered from any valid IMS command source. In IMS Version 8, these commands were called *classic* commands.

type-2 command
: A command that is entered only through the OM API. Type-2 commands are more flexible than type-2 commands and can have a broader scope. In IMS Version 8, these commands were called *IMSplex* commands or *enhanced* commands.

Accessibility Enhancements

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products. The major accessibility features in z/OS products, including IMS, enable users to:

- Use assistive technologies such as screen readers and screen magnifier software

- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

User Assistive Technologies

Assistive technology products, such as screen readers, function with the IMS user interfaces. Consult the documentation of the assistive technology products for specific information when you use assistive technology to access these interfaces.

Accessible Information

Online information for IMS Version 9 is available in BookManager format, which is an accessible format. All BookManager functions can be accessed by using a keyboard or keyboard shortcut keys. BookManager also allows you to use screen readers and other assistive technologies. The BookManager READ/MVS product is included with the z/OS base product, and the BookManager Softcopy Reader (for workstations) is available on the IMS Licensed Product Kit (CD), which you can download from the Web at www.ibm.com.

Keyboard Navigation of the User Interface

Users can access IMS user interfaces using TSO/E or ISPF. Refer to the *z/OS V1R1.0 TSO/E Primer*, the *z/OS V1R5.0 TSO/E User's Guide*, and the *z/OS V1R5.0 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.

Chapter 1. Introduction

Failure analysis structure tables (FAST) help you 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

In FAST, the abend descriptions help identify the failed subroutines from the functions that were active when the failure occurred.

FAST can be used to provide precise information about problematic symptoms for the IBM Early Warning System (EWS) and RETAIN. FAST provides definitions to problems that are encountered in IMS and can be used 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.

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.

Using FAST

Use the FAST 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.

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

- Standard abnormal terminations (abends)
- Pseudoabends

Standard Abends

An abend macro is issued at the point of error detection or by branching to a subroutine that issues the abend macro. 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 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 if either the contents of important control blocks should be written to the system log or a memory dump should be created.

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 will be 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. The FAST explain 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 abends that result from errors in DL/I, it is necessary for the DL/I call trace to be active. Refer to the *IMS Version 9: Diagnosis Guide and Reference* for instructions on invoking the DL/I trace.

Information in Failure Analysis Structure Tables

Figure 1 shows an example of a failure analysis and structure table entry. The notes below this table describe each of the components in this example entry.

ABENDU0014[1]
DFSFLLG0[2]

Explanation[3]**:** *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*

Analysis[4]**:** *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Code	***Meaning***
X'04'	*xxxxxxxxxxxxxxxxxxxx*
X'08'	*xxxxxxxxxxx*
X'0C'	*xxxxxxxxxxxxxxxxxxxxxxxx*
X'10'	*xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
X'14'	*xxxxxxx*
X'20'	*xxxxxxxxxxxxxxxxxxxxx*

Key[5]	Label[6]	Description[7]
Reg15=X'04' Reg5=length or Reg6=A(DFSPRMLL)	LLRET4	*xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx* *xxxxxxxxxxxxxxxx*

Possible Cause[8]**:** *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*

APAR Processing[9]**:** *xxxxxxxxxxxxxxx*

Figure 1. Example of a Failure Analysis Structure Table

Notes to Figure 1:

1. Abnormal termination code. Tables are ordered in ascending sequence of this code.
2. One or more modules that detect the condition that triggers this abend.
3. A statement of why the failure occurred.
4. Information to be used as a guide to a unique label below. Typical information provided includes general register usage in the failing module, the general control flow before failure, and the names of pertinent control blocks.
5. Key conditions that can be used to isolate the error to a single cause.
6. Label of the routine that detected the error.

   Note: This is not necessarily the routine that issued the abend.
7. A synopsis of the function of this routine and an indication of the cause of failure.
8. Situations that have caused this type of failure.
9. Additional documentation that is required to complete an APAR for this specific abend

Each of the tables in this book consists of 3 columns: Key, Label, and Description (see notes 5, 6, and 7 in Figure 1 on page 2). The ″Key″ 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 and the ″Label″ and ″Description″ areas provide an explanation of how and why the subroutine failed.

Formatted Dump

Two of the aids that IMS makes available for problem determination are the IMS Interactive Dump Formatter and the IMS Offline Dump Formatter. These aids allow you to format and print the entire IMS dump, or only those control blocks and data areas that are needed to analyze the problem. You can format dumps online, even though the Interactive and Offline formatters are the recommended approach. The IMS library provides more information about the Interactive Dump Formatter and Offline Dump Formatter in the following books:

- *IMS Version 9: Installation Volume 2: System Definition and Tailoring*
- *IMS Version 9: Utilities Reference: Database and Transaction Manager*
- *IMS Version 9: Diagnosis Guide and Reference*

Chapter 2. IMS Failure Analysis Structure Tables (FAST) 1 - 500

The following topics provide additional information about abends 2 through 500.

ABENDU0002

DFSV4100

Explanation
An IMS control region has abnormally terminated, and has forced the termination of all active dependent regions.

Analysis
ABENDU0002 is a pseudoabend scheduled by module DFSV4100 to abend the dependent regions.

ABENDU0005

DFSFDLY0

Explanation
The log termination failed to close the IMS system log during emergency restart. Message DFS0738X is issued.

Analysis
ABENDU0005 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	Label	Description
Reg15=X'01' RLWABC=X'01' PDCB, SDCB	LTA1240 LTA1260	A WADS DCB could not be opened. PDCB is the primary WADS DCB; When dual WADS are being used, SDCB is the secondary WADS DCB.
Reg15=X'07' RLWABC=X'07'	LTA1290	Either the WADS does not consist of enough tracks for a track group or there was a TRKCALC failure.
Reg15=X'09' RLWABC=X'09' RLWP.PABLKCNT	LTA2660 LTA2670 LTA2680	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.
Reg15=X'B' RLWABC=X'B'	LTA330A	A track group contained segments from more than one non-residual block.
Reg15=X'C' RLWABC=X'C' RLWPBA	LTA350	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	LTA350	An invalid record length was found in the rebuilt block pointed to by RLWPBA.

Key	Label	Description
Reg15=X'E' RLWABC=X'E' RLWPBA	LTA340R	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	LTA3530	A log record sequence error was found in the rebuilt block pointed to by RLWPBA.

ABENDU0008

DFSXRPS0, DFSPMBR0, DFSRED20

Explanation
IMS was unable to build a table to contain PROCLIB member data.

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.

ABENDU0009

DFSXRPS0, DFSIILD0, DFSRED20

Explanation
IMS was unable to OPEN the data set with DDNAME=xxxxxxxx specified in message DFS0597W.

Analysis
This standard abend is 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.

ABENDU0010

DFSXRPS0, DFSIILD0, DFSRED20

Explanation
An abend occurred for one of the following reasons:

- The record format is not valid for the data set. Message DFS0604W 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.

Analysis

This 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.

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'.

ABENDU0011

DFSDNSC0

Explanation

The system console device module, DFSDNSC0, was called by module DFSICIO0 to handle a WRITE interrupt. This condition should not occur.

Analysis

ABENDU0011 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 DDM 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:

Code	***Meaning***
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'8000000B'	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.

ABENDU0012

DFSXRPS0, DFSIILD0, DFSRED20

Explanation

The system abended for one of the following reasons:

- The block size is not valid for the data set. Message DFS0605W 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.
DFSIILDO0	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.

ABENDU0013

DFSFDLS0

Explanation

IMS batch is unable to open the primary log data set.

Analysis

ABENDU0013 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) respectively. 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.

ABENDU0014

DFSFLLG0, DFSCMS00

Explanation

An invalid request has been submitted to the IMS logger.

DFSFLLG0

Analysis

ABENDU0014 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 ***Meaning***

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

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.

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.

APAR Processing
Copy of the log and system dump with the save area trace.

ABENDU0015

DFSXRPS0, DFSIILD0, DFSRED20

Explanation
The data set did not contain the member required by IMS. Messages DFS0579W and DFS0596W provide additional information. Messages DFS0596W, DFS0597W, DFS3652X, and DFS3659X identify the name of the member and provide 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.

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.

ABENDU0016

DFSIINV0, DFSIINU0, DFSIILD0

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 (DFSCLVyx) could not be initialized. (y is a value from 0 to 9 or A to F; x is the IMS nucleus suffix.) Issued by DFSIINV0 after sending message DFS1998.
- One of the following occurred:
  - The VTAM control block modules could not be loaded. 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. Issued by DFSIINU0 after sending message DFS1992. R3 = length of storage request. R5 = IMODULE return code.

- The LQB/RCNT control block modules (DFSCLCms, DFSCLSms, DFSCLRms, DFSCLIDs, where m is a value from 0 to 9 or A to F; s is the IMS nucleus suffix) could not be initialized. Issued by DFSIINV0 after sending message DFS1990.
- The LQB/RCNT control block modules (DFSCLCms, DFSCLSms, DFSCLRms, DFSCLIDs, where m is a value from 0 to 9 or A to F; s is the IMS nucleus suffix) could not be loaded. Issued by DFSIINV0 after sending message DFS1991. 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. Issued by DFSIILD0 after sending message DFS1993. R3 = length of storage request. 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. Issued by DFSIILD0 after sending message DFS1993. R4 = length of storage request. 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. Issued by DFSIINV0 after sending message DFS1996. R4 = length of storage request. 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.

ABENDU0017

DFSFDLB0

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

Analysis
ABENDU0017 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. LBBLKCNT 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.

ABENDU0019

DFSCVRB0

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).

Analysis
ABENDU0019 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.

ABENDU0020

DFSFCTT0

Explanation
The IMS system or the specified external subsystem was terminated by a MODIFY command task. IMS terminates abnormally.

Analysis
ABENDU0020 is a standard abend issued by DFSFCTT0.

Possible Cause
User entered a z/OS MODIFY command to abend all of IMS or a specific task representing an external subsystem connection, such as DB2.

ABENDU0021

DFSFDLG0, DFSFDLS0

Explanation
The IMS system log writer was passed an invalid request or detected an illegal condition.

DFSFDLG0

Analysis
ABENDU0021 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)	WRTE0960 WRTE1060	During OLDS write processing, register 9 points to a block whose block sequence number is not 1 greater than the previous block's sequence number (LDWRTCNT).

DFSFDLS0

Analysis

ABENDU0021 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.

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.

ABENDU0022

DFSCVRG0

Explanation

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

Analysis

ABENDU0022 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:

| ***Code*** ***Meaning***

- **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.

ABENDU0023

DFSSDLC0

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.

Analysis

The specification of the ACBLIB DDNAMES IMSACBA and IMSACBB needs to 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 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.

ABENDU0024

DFSRRA00

Explanation
Either a DL/I subordinate address space region or a DBRC region attempted to connect to the IMS control region. The job name 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.

ABENDU0025

DFSRRA00

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, for example, is caused by starting the DL/I subordinate address space region twice. This abend is issued from label TESTSAS.

ABENDU0026

DFSPCC20, DFSPCC30, DFSPR000

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

Analysis
ABENDU0026 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):

Code	*Meaning*
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 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).

DFSPCC20

Analysis

ABENDU0026 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 Reg1=completion code, X'8000001A'	PC00 PC01	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.

DFSPCC30

Analysis

ABENDU0026 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.

DFSPR000

Analysis

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

ABENDU0026 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.

ABENDU0028

DFSTERM0

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

Analysis
ABENDU0028 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 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.

ABENDU0029

DFSXDL00, DFSXDL10, DFSSDLC0

Explanation
ABENDU0029 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.

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	DFSV4200 was invoked to build the SSCTs 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.

Key	Label	Description
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 DFSRSMD0 failed.
Reg3=X'23'	ABCOD35	DFSRSMD0 call failed.

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	DFSIIND0 was called to perform database 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 user ID (see message DFS0841I in *IMS Version 9: Messages and Codes, Volume 2* for a description of the specific problem.)

DFSXDT10

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

Code	Meaning
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 can not 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.

DFSSDLC0

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

ABENDU0031

DFSULG20

Explanation

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

Analysis

ABENDU0031 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.

ABENDU0032

DFSULG40

Explanation
A bad log was read by the Log Recovery Utility.

Analysis
ABENDU0032 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.

ABENDU0034

DFSSCBT0

Explanation
Module DFSSCBT0 or DFSBCB30 was entered to get or release the 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
ABENDU0034 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 SAPACLLE. Refer to mapping macro DFSCLLE for a description of the formatted CLLE area.

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

Code	***Meaning***
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:

Code	***Meaning***

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.

ABENDU0035

DFSDSC00

Explanation

ABENDU0035 is issued if the DBCTL 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.

ABENDU0036

DFSCVCK0

Explanation

An unexpected posting of a VTAM request parameter list (RPL), resulting from either or both a VTAM and IMS logic error, was encountered.

Analysis

ABENDU0036 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.

ABENDU0037

DFSHTKR0

Explanation

During takeover, module DFSHTKRO called Internal Resource Lock Manager (IRLM) for takeover request processing, and IRLM returned a return code greater than 4.

Analysis
ABENDU0037 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 code information in *IMS Version 9: Messages and Codes, Volume 1*.

Possible Cause
Failure in the IRLM.

ABENDU0038

DFSPCCC0, DFSRST00

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 IRLM return code information in *IMS Version 9: Messages and Codes, Volume 1*
- The information about message DFS038I in *IMS Version 9: Messages and Codes, Volume 2*

Possible Cause
Failure in the IRLM.

ABENDU0039

DFSPCCC0, DFSRDSH0, DFSRST00

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. To determine the cause of the failure, see the IDENTIFY request explanation in the IRLM return and reason code information in *IMS Version 9: Messages and Codes, Volume 1*.

Analysis
ABENDU0039 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.

Possible Cause
Back out may be needed.

ABENDU0040

DFSPCCC0, DFSRDSH0, DFSRST00, DBFLHCK0

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

Analysis
ABENDU0040 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 code information in the *IMS Version 9: Messages and Codes, Volume 1*.

Possible Cause
Failure in the IRLM.

ABENDU0041

DFSPCCC0, DFSPCC30, DFSRST00

Explanation
A signon request was issued to DBRC and the request failed. The DBRC return code defines the error. To determine the cause of the failure, refer to:

- The DBRC request return code information in *IMS Version 9: Messages and Codes, Volume 1*
- The message DFS041I information in *IMS Version 9: Messages and Codes, Volume 2*

Analysis
ABENDU0041 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.

ABENDU0042

DFSRLP00

Explanation
Back out was required for this startup and the sign on 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
ABENDU0042 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.

ABENDU0043

DFSPCCC0, DFSRST00

Explanation
A sign on request for “recovery end” was issued to DBRC and the request failed. This request issued after the initial checkpoint at the end of emergency restart notifies DBRC of completion of an emergency restart. 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:

- The DBRC request return code information in *IMS Version 9: Messages and Codes, Volume 1*
- The message DFS0413I information in *IMS Version 9: Messages and Codes, Volume 2*

Analysis
ABENDU0043 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 PXPARMS. The parameter list address can be located in the checkpoint PST listing with all the information required to issue the signon request.

ABENDU0044

DFSRLP00, DFSXBAT0

Explanation
A normal or emergency restart was specified. DBRC was active during the previous execution, but is not active now.

DBRC=FORCE was specified on the IMSCTRL system definition macro statement, and an execution of IMS was requested with an execution-time parameter of DBRC=N.

Analysis
ABENDU0044 is a standard abend issued by DFSRLP00 and DFSXBAT0.

Possible Cause
DBRC=N was specified.

ABENDU0045

DFSRLP00

Explanation
The same IRLM was not used for this execution of IMS or was not present. If the IRLM was present for emergency restart during the previous execution, it is required that it be present for this execution. Message DFS045I precedes this abend.

Analysis
ABENDU0045 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.

ABENDU0046

DFSPCC20, DFSSBMP0

Explanation
Conflicting PROCESSING INTENTs were found between one of the DBPCBs and the ACCESS parameter. The DBPCBs are contained in the PSB (the third positional parameter on the EXEC control statement). The ACCESS parameter is defined in the DATABASE macro statement for the IMS online subsystem. The PROCESSING INTENT is derived from the PROCOPT operand specifications in the PSBGEN.

Analysis
ABENDU0046 is a pseudoabend set by module DFSSBMP0 and issued from module DFSPCC20. See message DFS046A in *IMS Version 9: Messages and Codes, Volume 2* 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.

ABENDU0047

DFSDBAU0, DFSPCC20, DFSSBMP0

Explanation
A database authorization request to DBRC failed, or the attempt to obtain a work area in which to build a database authorization request list failed.

DFSDBAU0

Analysis
ABENDU0047 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 message DFS047A in *IMS Version 9: Messages and Codes, Volume 2*.

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

DFSPCC20, DFSSBMP0

Analysis
ABENDU0047 is a pseudoabend set by module DFSSBMP0 and issued from module DFSPCC20. See message DFS047A in *IMS Version 9: Messages and Codes, Volume 2* 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.

ABENDU0048

DFSXDRC0, DFSXRIC0

DFSXDRC0

Explanation
An error occurred during initialization of a database recovery control.

Analysis
ABENDU0048 is issued for all abnormal conditions. Register 15 contains the following reason codes:

Code	*Meaning*
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.

DFSXRIC0

Analysis
ABENDU0048 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	*Meaning*
X'04'	IMODULE GETMAIN for the IMS-DBRC control block (DFSRCWK) failed.
X'08'	IMODULE LOAD of DFSRCQM0, DFSRCQR0, or DSPCRTR0 failed.
X'0C'	First call for initialization (INIT-0) had a nonzero return code.
X'10'	Second call for initialization (INIT-1) had a nonzero return code.

Take appropriate action according to the error code.

Code	*Meaning*
X'04'	Increase the IMS control region size.
X'08'	Correct the JOB/STEPLIB DD statement, or link-edit the DBRC modules into the correct library.
X'0C'	Determine the reason for the DBRC initialization failure by referring to the appropriate DBRC database recovery control documentation.
X'10'	Determine the reason for the DBRC initialization failure by the second call for DBRC initialization (INIT-1), which had a nonzero return code.

DFSXRID0

Explanation

Analysis
ABENDU0048 is issued for all abnormal conditions. Register 15 contains the following reason codes:

Code	*Meaning*
X'08'	IMODULE LOAD of DSPCRTR0 failed.
X'0C'	First call for initialization (INIT-0) had a nonzero return code.

X'10' Second call for initialization (INIT-1) had a nonzero return code.

X'1C' IMODULE LOAD for DFSRSMD0 failed.

X'20' DFSRSMD0 call failed.

ABENDU0049

DFSDBAU0, DBFDBAU0

Explanation
IMS has encountered an OS RDJFCB error or an out-of-storage condition in module DFSDBAU0. When given for an out-of-storage condition, this abend can only occur while using the IRLM and an IRLM failure or an IRLM communication failure has occurred.

Analysis
ABENDU0049 is a standard abend issued by module DFSDBAU0. The program status word (PSW) at entry-to-abend points to one of three locations, either after label ST307 or at label AB049 in module DFSDBAU0 or at label DBFSTATS in DBFDBAU0.

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).

ABENDU0050

DFSXTRA0

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.

ABENDU0056

DBFDBFI0, DFSDASI0, DBFINTE0, DBFINTT0

Explanation
An error occurred in a DBCTL environment while Fast Path was trying to access buffers for a Coordinator Controller (CCTL). 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, DBFDBF10 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 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, DFSDASI0 sets up pseudoabend U0056 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 identify 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	***Meaning***
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.

ABENDU0069

DBFIRC10, DFSCPY00, DFSDCPY0

Explanation

A recursive Fast Path IRC entry was detected by the IMS Fast Path inter-region communication SVC or PC.

Analysis

ABENDU0069 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.

Key	Label	Description
Reg15=X'04'		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'		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.

APAR Processing

Dependent region abend dump.

ABENDU0070

Several Modules

Explanation
A GETMAIN failure occurred during system initialization or subsequent execution.

Analysis
ABENDU0070 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/VS SVC (from the SCD) was just issued using an Execute instruction. The IMS/VS SVC that services GETMAIN is DFSMODU0. The execution library load module name is IGC14xxx, where xxx is the user-specified IMS/VS SVC number. The IMS/VS SVC (DFSMODU0) returns an error code in register 15 (bytes 0 and 1). For an explanation of the IMODULE and DFSBCB return codes, see the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

DFSSTKMG

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

Analysis
ABENDU0070 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.

DBFICI10

Analysis
ABENDU0070 is a standard abend that can be issued by the IMS/VS 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/VS/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.

DBFIFIX0

Analysis

ABENDU0070 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.

DFSIINF0

Analysis

ABENDU0070 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.

DFSIRST0

Explanation

An IMS IMODULE LOAD failed.

Analysis

ABENDU0070 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.

DFSTMOD0

Analysis

ABENDU0070 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.

DFSXBAT0

Explanation

A request for storage allocation failed.

Analysis

ABENDU0070 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.

DFSXCIC0

Analysis

ABENDU0070 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. • IMODULE LOAD for DFSSDLO0 failed. • AWE request (DL/I SAS INIT) failed. • QUICKSAVE request failed. • DPST request failed.

DFSXSTM0

Analysis

ABENDU0070 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		IMODULE GETMAIN for Latch Control Block area failed.
Reg14=BAL		IMODULE GETMAIN for Common Services ITASK work area failed.
Reg14=BAL		Creation of Master Services ITASK failed.
Reg14=BAL		Creation of Storage Compression ITASK failed.
Reg14=BAL		Request for AWE for Latch ITASK creation failed.

DFSXLGI0

Analysis

ABENDU0070 is a standard abend that can be issued by the logical/physical logger initialization module, DFSXLGI0. 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.

ABENDU0071

Several Modules

Explanation

During system initialization or subsequent execution, an error occurred in a lower-level module.

Analysis

ABENDU0071 is a standard abend issued by one of the following modules: DBFXFP00, DBFXFP10, DFSFDLB0, DFSFDLD0, DFSFDLG0, DFSFDLS0, DFSFLLG0, DFSHSRV0, DFSPCC30, DFSRCFS0, DFSRESP0, DFSXBAT0, DFSXCIC0, DFSXCTL0, DFSXCTX0, DFSXDLG0, DFSXLGI0, DFSXLIC0, DFSXRCF0, DFSXSL10, or DFSXSTM0.

Possible Cause

The lower-level module detected an internal IMS error.

DBFXFP00

Analysis

ABENDU0071 is a standard abend issued by DBFXFP00. DBFXFP00 invokes lower-level modules to perform specific initialization. The ABENDU0071 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.

DBFXFP10

Analysis

ABENDU0071 is a standard abend issued by DBFXFP10. DBFXFP10 invokes lower-level modules to perform specific initialization. The ABENDU0071 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=FIND return code=DFSCBTS FUNC=ALTER return code=DFSCWU FUNC=CWU return code=DBFICI10 return code=DBFINI20 return code=DBFIFIX0	ABEND1	DBFXFast Path10 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=FIND 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 DBFXFast Path10 results in an abend.

DFSFDLB0

Explanation

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

Analysis

ABENDU0071 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), respectively. 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.

DFSFDLD0

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

Analysis
ABENDU0071 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.

DFSFDLG0

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

Analysis
ABENDU0071 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), respectively. Register 14 contains the address at which the failure was detected. For a description of the IMODULE return codes, see *IMS Version 9: Messages and Codes, Volume 1*.

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.
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.

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 return code in register 15.

DFSFDLS0

Explanation
ABENDU0071 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).

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

DFSFLLG0

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

Analysis
ABENDU0071 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	LAWE0100	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.

DFSHSRV0

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

Analysis
ABENDU0071 is a standard abend that can be issued by the hot standby surveillance module, DFSHSRV0. DFSHSRV0 invokes lower level modules to perform 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 should be 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.

DFSPCC30

Analysis

ABENDU0071 is a standard abend issued by DFSPCC30. DFSPCC30 invokes lower-level modules to perform specific initialization. ABENDU0071 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.

DFSRCFS0

Explanation

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

Analysis

ABENDU0071 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.

DFSRESP0

Analysis

ABENDU0071 is a standard abend issued by DFSRESP0. DFSRESP0 invokes lower-level modules to perform specific functions. ABENDU0071 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 *IMS Version 9: Messages and Codes, Volume 2*.

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, and are explained below:

Code	***Meaning***
4	End of chain (SCAN).
8	Block not found.
12	Invalid function code.
16	Element type not set.
20	Element address not set.
24	Element to be dequeued not on chain.
28	Control block table (CBT) entry 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 an explanation of IMODULE return codes refer to the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

DFSRST00

Analysis

ABENDU0071 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.

DFSRLP00

Analysis

ABENDU0071 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.

DFSXBAT0

Explanation

A lower level initialization request failed.

Analysis

ABENDU0071 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
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.
Reg14=BAL Reg15=*0000000C*	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. *xxxx* indicates a reason code defined below, followed by the return code from the lower level service. **Code** **Meaning** **0004** IMODULE GETMAIN failed. *yyyy* is the IMODULE return code. **0008** DFSLOADL failed. *yyyy* is the DFSLOADL return code.

DFSXCIC0

Analysis

ABENDU0071 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 ABENDU0071. Register 14 in the abend SVRB should be used to isolate to the specific label below. Register 12 is the base register, and register 11 is the pointer to the SCD.

Key	Label	Description
Reg14=BAL	CICSTART	DFSIINM0 is called to load the PDIR and DDIR.
Reg14=BAL Reg15=X'04' or X'0C'	CICSTART	DFSIIND0 is called to initialize the PSB and DMB directories. If the return code in register 15 is X'04', one of the following messages was issued giving 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.
Reg3=X'10'	ABCOD16	DFSXRAC0 was unable to set RACF user ID (specific problem described in message DFS0841I).
Reg14=BAL	CIC4A	DFSIINB0 is called to initialize the communications facilities.
Reg14=BAL	CIC6	DFSIFIX0 is called to fix storage as directed by the user.
Reg6=STAE parmlist Reg7=SCDFLOS	(E)STAE	Unsuccessful completion of the (E)STAE SVC (SVC 60). Message DFS0406A is issued prior to this abend.
Reg14=BAL Reg15=Return code from IMODULE	NODLIS4	Unsuccessful IMODULE LOAD of module DFSCST00.
Reg14=BAL Reg15=return code from IMODULE	DBRCDONE	Unsuccessful IMODULE LOAD of module DFSLATE0.
Reg14=BAL Reg15=reason code	CICSTART	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: **X'0101'** CQSREG request failed. **X'0102'** CQSCONN request failed. **X'0103'** CQSCONN request failed with a message queue (MSGQ) structure error. **X'0104'** CQSCONN request failed with an expedited message handler queues (EMHQ) structure error. **X'0105'** CQSCONN request failed with MSGQ and EMHQ structure errors. **X'0106'** MSGQ structure attribute WAITRBLD is incorrect. **X'0107'** EMHQ structure attribute WAITRBLD is incorrect. **X'0108'** IMODULE GETSTOR request failed. **X'0109'** CQSUNLCK FORCE failed for MSGQ or EMHQ structure.

DFSXCTL0

Analysis

ABENDU0071 is a standard abend that can be issued by DFSXCTL0.

Key	Label	Description
Reg15=return code	CREAT400	DFSCWU macro failed to obtain a PST and returned a nonzero return code.

DFSXCTX0

Explanation

ABENDU0071 is a standard abend, issued by module DFSXCTX0.

Analysis
DFSXCTX0 invokes lower level modules to perform specific initialization. ABENDU0071 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.

DFSXDCC0

Analysis
ABENDU0071 is a standard abend issued by DFSXDCC0. DFSXDCC0 invokes lower-level modules to perform specific initialization tasks. The ABENDU0071 is the result of a nonzero return code passed back to DFSXDCC0 from one of the lower-level modules. Register 14 in the abend SVRB should be used to isolate to a specific label below. 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. *xxxx* indicates a reason code defined below, followed by the return code from the lower level service. **Code** **Meaning** **0004** IMODULE GETMAIN failed. *yyyy* is the IMODULE return code. **0008** DFSLOADL failed. *yyyy* is the DFSLOADL return code.

DFSXDLG0

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

Analysis
ABENDU0071 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 DFSXLGI0 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.

Key	Label	Description
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.

DFSXLGI0

Explanation

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

Analysis

ABENDU0071 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 member DFSVSMxx of IMS PROCLIB. Message DFS2205I was issued before the abend.

DFSXLIC0

Analysis

ABENDU0071 is a standard abend issued by DFSXLIC0. DFSXLIC0 invokes lower-level modules to perform specific initialization tasks. The ABENDU0071 is the result of a nonzero return code passed back to DFSXLIC0 from one of the lower-level modules. Register 14 in the abend SVRB should be used to isolate to a specific label below. 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:

Code ***Meaning***

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.

Key	Label	Description
Reg14=BAL Reg15=*xxxxyyyy*	XLIC0135	DFSFTIN0 encountered an error during initialization. *xxxx* indicates a reason code defined below, followed by the return code from the lower level service. **Code** **Meaning** **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 link-editing of DFSXLIC0.
Reg15=IDM1 Reg15=IDM2 Reg15=IDM3 Reg15=IDM4 Reg15=IDM5 Reg15=IDM6		DFSIIDM0 returns one of the following: • Allocation of DFSPOOL AOIP failed. • AOIP storage requests for control blocks failed. • Standard User Exit definition failed for DFSAOE00. • AOIP storage request for DFSAOE00 work area failed. • AOIP storage request for DFSAOE00 interface block failed. • Load of DFSAOSC0, ICMD security module, failed.
Reg14=BAL		DFSIIOC0 is called to process on-line change blocks.

DFSXRCF0

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

Analysis
ABENDU0071 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. See message DFS2930I in the *IMS Version 9: Messages and Codes, Volume 2* for more information on these subcodes.

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.

DFSXSL10

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

Analysis
ABENDU0071 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 ABENDU0071. Register 14 in the abend SVRB should be used to isolate to the specific label below. 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.

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.

DFSXSTM0

Analysis

ABENDU0071 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 to 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. Message DFS2930I in *IMS Version 9: Messages and Codes, Volume 2* describes these codes.

If DFSCSL00 detects an error during the common service layer (CSL) initialization, module DFSXSTM0 will issue message DFS2930I with reason code X'18' and ABENDU0071 will occur.

The following reason codes result from errors detected by DFSCSL00:

Code ***Meaning***

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:

Code ***Meaning***

2004 Problem with DFSCGxxx PROCLIB member. See message DFS3305E for additional information.

200C OLC=GLOBAL is specified but OLCST is not specified.

ABENDU0072

DFSXLGI0

Explanation
A DEVTYPE or TRKCALC macro was issued for the log data sets and resulted in a nonzero return code.

Analysis
ABENDU0072 is a standard abend that can be issued by 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 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.

ABENDU0073

DFSFDLS0, DFSXLGI0, DFSXLIC0

DFSFDLS0

Explanation
There was an insufficient number of log data sets given to IMS.

Analysis
ABENDU0073 is a standard abend that can be issued by the physical logger, DFSFDLS0, during a warm or emergency restart.

In order to protect IMS recovery resources, this abend does not generate a dump.

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

DFSXLGI0

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

Analysis
ABENDU0073 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. Prior to 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 statement(s) 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	There were fewer than 3 OLDS DD statements with the same DFSOLPxx on-line. There was no IEFRDER DD statement in the batch although the log was required.
Reg15=X'0C'	ABEND073	The OLDS block size was either too small or not a multiple of 2048.

DFSXLIC0

Explanation
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.

Analysis
ABENDU0073 is a standard abend issued by module DFSXLGI0 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 IEFRDER 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 if this output area has had any data moved into it after completion of the SVC 24. This 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 prior to 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.

ABENDU0074

DFSXLGI0

Explanation
The RDJFCB macro was issued for the system log or the monitor log and resulted in a nonzero return code.

Analysis
| ABENDU0074 is a standard abend that can be issued by DFSXLGI0. 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.

ABENDU0075

DFSXLGI0

Explanation

The log data set device is not acceptable for IMS.

ABENDU0075 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.

ABENDU0076

Explanation

ABENDU0076 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 14 points to the location where the error was detected. Register 15 contains the return code from the system service that found the error.

DFSTRA00

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

DFSXCFB0

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

ABENDU0077

DFSXBAT0, DFSXSTM0

Explanation
A DELETE failure occurred during initialization.

Analysis
ABENDU0077 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.

DFSXBAT0

Analysis
ABENDU0077 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.

DFSXSTM0

Analysis
ABENDU0077 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.

ABENDU0078

DFSXDBI0

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.

Analysis

ABENDU0078 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 LOAD return codes, see *IMS Version 9: Messages and Codes, Volume 1*.

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 *IMS Version 9: Installation Volume 2: System Definition and Tailoring*.

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.

ABENDU0079

DFSXNCL0

Explanation

This abend occurs if the IMS nucleus, load module DFSVNUCs or DFSCNUCs, where s is the nucleus suffix, could not be loaded during control region initialization.

Analysis

ABENDU0079 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. For a description of the IMODULE return codes, see *IMS Version 9: Messages and Codes, Volume 1*.

Key	Label	Description
Reg3=RGPARMS 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.

ABENDU0080

DFSAOSF0, DFSAOS10

Explanation

An error occurred in Overflow Sequential Access Method (OSAM) OPEN/CLOSE/EOV processing.

Analysis

ABENDU0080 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'20'	Open processing.
X'30'	EOV processing.
X'40'	Close processing.
X'50'	Termination processing.

Register 3 contains the DCB address for all types of processing except Entry and Termination processing. Register 8 contains the private area address.

Register 6 points to the private area; at offset X'68' in that area, you will 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)

Code	***Meaning***
X'01'	Incompatible operating system.
X'02'	Invalid parameter list passed.
X'03'	Invalid function code detected.
X'04'	Zero SCD address in vector.

Open Processing (Label OPNABxx)

Code	***Meaning***
X'01'	DCB is already open as DD DUMMY.
X'02'	This is not OSAM open calling.
X'03'	The DCB is already open.
X'04'	Not used.
X'05'	Reading the job file control block (JFCB) failed.
X'06'	OS OPEN failed, but DCB abend EXIT was not entered.

X'07' OS OBTAIN failed for DSCB.

X'08' Error occurred in OSAMDEB subroutine.

X'09' Unable to locate module DFSAOS60 or DFSAOS80.

X'0A' Not used

X'0B' Invalid return code from TTR convert routine.

EOV Processing (Label EOVABxx)

Code ***Meaning***

X'01' DCB/DEB check failure in COPYDCB subroutine.

X'02' Flag not set for end of volume (EOV) in progress.

X'03' Error occurred in OSPLIT. PRISVSCD contains an error code.

X'04' ERROR occurred in OSDEB subroutine.

X'05' Error occurred in OSAMDEB subroutine.

Close Processing (Label CLSABxx)

Code ***Meaning***

X'01' DCB is already closing.

X'02' DCB/DEB check failure in COPYDCB subroutine.

X'03' Error occurred in OSPLIT. PRISVSCD contains an error code.

X'04' Error occurred in OSDEB subroutine.

X'05' IMODULE delete error.

Termination Processing (Label OTRMABxx)

Code ***Meaning***

X'01' IMODULE GETMAIN failed for CSA DEB.

X'02' Page fix failed for CSA DEB.

X'03' Page fix failed for final DEB.

DFSAOS10

Explanation

This abend occurs when module DFSAOS10 has received a request to schedule EOV processing for a DASD data set but has found that EOV processing is already in progress for that data set.

Analysis

ABENDU0080 is a standard abend issued by the OSAM OPEN interface module DFSAOS10. The program status word (PSW) at entry-to-abend points to the instruction within module DFSAOS10 at label EOVERR2, from which the abend (SVC 13) is issued. The abending label is EOVERR1, which is branched to by macro EOVOS at which the error is detected.

The following registers in the abend SVRB register are noteworthy:

Register 3 is the DCB address.

Register 5 is the DECB address.

Register 12 is the base register.

Key	Label	Description
Reg6=the address of the EOV sync words	N0WRTFM	The EOV sync words are used to synchronize I/O operations with EOV processing. Register 6 points to the sync words and shows their contents at abend.

ABENDU0081

DFSXSTM0

Explanation
Unable to build or set the SCD address into the termination control block.

Analysis
ABENDU0081 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 *IMS Version 9: Messages and Codes, Volume 1*.

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.

ABENDU0085

DFSFDSC0

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
ABENDU0085 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.

ABENDU0088

DFSXLSM0, DFSKLSO0, DFSXLSD0, DFSKLSD0

Explanation
Initialization for the local storage option (LSO) failed.

Analysis
ABENDU0088 is issued by modules DFSKLSO0 and DFSXLSM0 for all abnormal conditions.

Register 15 contains the following reason codes.

Attention: ABENDU4095 is issued in DFSKETXR to propagate any LSO abends.

DFSXLSM0

Code ***Meaning***

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)

DFSKLSO0

Code ***Meaning***

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)

DFSXLSD0

Code	*Meaning*
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.)

DFSKLSD0

Code	*Meaning*
X'14'	The DFSBCB quick save obtain macro failed (Register 3 = DFSBCB return code).

ABENDU0090

DFSFFET0, DFSFPRF0

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).

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.

DFSFFET0

Explanation
ABENDU0090 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.

Analysis
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) Reg15=nonzero return code from IPOST	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.

DFSFPRF0

Explanation
ABENDU0090 is a standard abend that can be issued by module, DFSFPRF0, which prefetches blocks into storage. 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 if the return code from the IPOST macro is not zero.

Analysis
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.

ABENDU0092

DFSCRPQ0, DFSRLP00

Explanation
An IMODULE GETMAIN request for storage in the private/extended private area of the IMS control region could not be satisfied.

Analysis
The private/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. Refer to *IMS Version 9: Messages and Codes, Volume 1* for a complete explanation of the IMODULE GETMAIN request return codes.

ABENDU0095

DFSIILD0

Explanation
An unrecoverable error occurred while initializing the logon, user, and MSC descriptors.

Analysis
Message DFS3658, prior to the abend, contains the reason for the failure.

ABENDU0097

DFSPCC30

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. See *z/OS DFSMS Macro Instructions for Data Sets* for an explanation of the return codes from the VSAM DLVRP macro.

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

ABENDU0098

DFSRCP30

Explanation
Edit routine address not found in table. The descriptor was ignored and message DFS368OW was returned.

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.

ABENDU0101

DFSRCJB0, DFSPCJB0, DFSPCJM0, DFSRCJM0

Explanation
An error occurred during Java dependent region processing.

Analysis
The user should 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.

DFSRCJB0

Analysis
ABENDU0101 is 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.
		BPX1SDD failed. Reg15=X'6' Reg3=BPX1SDD return code Reg4=BPX1SDD reason code

DFSPCJB0

Analysis

ABENDU0101 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 master Java Virtual Machine (JVM).
Reg15=X'3' Reg3=Possible CEEPIPI return code		JVM unable to invoke application (CEEPIPI error).
Reg15=X'4'		JVM unable to invoke application.

Related Reading: For CEEPIPI errors, see *z/OS Language Environment Programming Guide*.

DFSPCJM0

Analysis

ABENDU0101 is a standard abend issued by DFSPCJM0.

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 worker JVM.
Reg15=X'3' Reg3=CEEPIPI return code		JVM unable to invoke application (CEEPIPI error).
Reg15=X'4'		JVM unable to invoke application.

Related Reading: For CEEPIPI errors, see*: z/OS Language Environment Programming Guide*.

DFSRCJM0

Analysis

ABENDU0101 is a standard abend issued by DFSRCJM0.

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 master JVM.
Reg15=X'3'		Required ENVIRON=member not specified.
Reg15=X'4'		Required JVMOPMAS=member not specified.
		BPX1SDD failed.
Reg15=X'6' Reg3=BPX1SDD return code Reg4=BPX1SDD reason code		

Related Reading: For CEEPIPI errors, see *z/OS Language Environment Programming Guide*.

ABENDU0102

DFSZSR00, DFSZSR10

Explanation

This is a standard abend that the program restart handler can issue in 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.

Analysis

| During online batch processing, IMS determines the OLDS position from the last complete checkpoint of
| the BMP or JBP application from an internal IMS table. The Residual Recovery Element (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, IMS tries
| to locate the last checkpoint records from the OLDS. If the OLDS has been overwritten since the last
| checkpoint was recorded and the IMSLOGR DD data has been allocated in the JCL of the BMP or JBP
| region, the application restart process attempts to access the checkpoint records from the IMSLOGR DD
| data set.

In a batch environment the checkpoint records are accessed from the IMSLOGR DD data set.

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 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 read of the OLDS.
Reg3=C4D3A004		An error was returned from the log read exit routine, LOGREXIT.

Key	Label	Description
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.
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 correct 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. • An incorrect checkpoint ID was specified. • An incorrect or insufficient number of volume serials was specified. • The program name on the execute statement does not match the program name in the type X'18' log records. • An incorrect jobname was used. The restarted jobname must match the original jobname. You must provide correct the correct or recovered IMSLOGR data set and rerun the job.
Reg2=E2D9000A	E2D9000A	An invalid call was made to the SVC interface. This 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.

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. You will need to rerun with a prior GEN.
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; that is, 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.

Key	Label	Description
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. • An incorrect checkpoint ID was specified. • An incorrect or insufficient number of volume serials was specified. 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.
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 may 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.

ABENDU0103

DFSZSC00

Explanation

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.
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.

ABENDU0106

DFSPCC20

Explanation

A nonzero return code was issued in response to an IMODULE LOAD request during dependent region initialization.

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. For an explanation of the IMODULE return codes, refer to the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

Key	Label	Description
Reg1=X'8000006A'	PCAB106	IMODULE error.

ABENDU0107

DFSXLIC0

Explanation

Either:

- Loading of one of the security modules failed to be completed.
- If RACF is used (z/OS only), then RACF is at the wrong level or it is not installed on the system.

Analysis

This is a standard abend issued by module DFSXLIC0.

Register 15 contains one of the following codes:

Code ***Meaning***

X'01' IMODULE LOAD failed for module DFSSCHR0.

X'02' IMODULE LOAD failed for module DFSISIS0.

X'03' IMODULE LOAD failed for module DFSRAS00.

| **X'04'** An attempt to define the DFSRAS00 user exit to IMS failed.

| **X'05'** An attempt to get storage for the initialization call of the DFSRAS00 user exit failed.

ABENDU0108

DFSASLT1

Explanation

A timer request failure occurred during an LU 6.2 shutdown. The LU 6.2 shutdown was not completed and was unable to proceed. DFSTIMER returned with a bad return code.

Analysis

ABENDU0108 is a user abend issued by DFSASLT1 to prevent loss of LU 6.2 messages during normal IMS shutdown.

ABENDU0111

DFSDASI0

Explanation

When a CCTL issues an INIT call to a DB/DC or DBCTL environment for connection, DBCTL builds a status index entry (SIDX) for this CCTL and queues it to the SIDX chain. The enqueue request failed and the CCTL connection process is terminated.

ABENDU0111 is a pseudoabend set in module DFSDASI0.

Analysis

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:

Code	***Meaning***
X'14'	No SIDX element for enqueue
Others	Internal errors

ABENDU0113

DFSASK00, DFSSDA20, DBFATRM0, DFSFUNL0

DFSASK00, DFSFUNL0

Explanation

An abend occurred in a message region or batch message region while DL/I or DC was processing a call in the message region.

Analysis

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 ABENDU0113.

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 ABENDU0113.

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 ABENDU0113 or the non-IMS formatted 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 dump formatting is chosen, the dump is taken during the dependent region abend processing and shows the dependent region abend code. The control region is then terminated with ABENDU0113, but no additional dump is taken unless the offline dump formatting was not successful. Analyzing dependent region SDUMPs is the same as analyzing other IMS SDUMPs.

DFSSDA20

Explanation

The IMS end of task (EOT) exit routine discovered a dependent region active in a DL/I or DC call. DFSABSAV does not contain ABENDU0113 diagnostic information. Instead, at the time of the abend register 4 contains the address of the PST of the original abend.

Analysis

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 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 ABENDU0002.

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.

DBFATRM0

Explanation

Fast Path detected an 'in Fast Path' condition for a dependent region that is abending. DFSABSAV does not contain ABENDU0113 diagnostic information. Instead, at the time of the abend, register 4 contains the address of the PST of the original abend.

Analysis
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 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 ABENDU0002.

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.

ABENDU0114

DFSSCHR0

Explanation
A nonzero return code was received from an IMODULE GETMAIN request.

Analysis
ABENU0114 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 an explanation of IMODULE return codes refer to the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

ABENDU0119

DFSTMS00

Explanation
This abend can occur when:

- An LU6.2 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 ABENDU0119 code was used to end this instance of the transaction. The program and transaction were not stopped.

ABENDU0120

DFSXNCL0

Explanation
DFSXNCL0 could not complete the address resolution between the IMS control region nucleus and the SCD or any SCD extension.

Analysis
ABENDU0120 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).

ABENDU0121

DFSXDBI0

Explanation

When the IMS control blocks module, DFSBLK00, was loaded, the SCD (CSECT DFSISCD) was not the entry point as required.

Analysis

ABENDU0121 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 below.

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.

ABENDU0123

DFSECP10, DFSECP20

Explanation

A non-retriable DEADLOCK condition was detected while processing an IMS database call for a modified application program. The application program had allocated at least one CPI-C conversation when the deadlock was detected. This abend is the same as ABENDU0777 but is not retriable and the input message is discarded. CPI communications driven application programs are always non-retriable.

Analysis

ABENDU0123 is a pseudoabend, set by module DFSTMR00 and issued by modules DFSECP10 and DFSECP20.

ABENDU0124

DFSECP10, DFSECP20

Explanation
A non-retriable U2478 condition was detected while processing an IMS database call for a Modified application program. The application program had allocated at least one CPI-C conversation when the number of PI waiters exceeded 63. This abend is the same as ABENDU2478 but is not retriable and the input message is discarded. CPI communications driven application programs are always non-retriable.

Analysis
ABENDU0124 is a pseudoabend, set by module DFSTMR00 and issued by modules DFSECP10 and DFSECP20.

ABENDU0125

DFSECP10, DFSECP20

Explanation
A non-retriable LOCK REJECT condition was detected while processing an IMS database call for a modified application program. The application program had allocated at least one CPI-C conversation when the lock reject occurred. This abend is the same as ABENDU3303 except it is not retriable and the input message is discarded. CPI communications driven application programs are always non-retriable.

Analysis ABENDU0125 is a pseudoabend, set by module DFSTMR00 and issued by modules DFSECP10 and DFSECP20.

ABENDU0126

DFSECP10, DFSECP20

Explanation
A non-retriable U2479 condition was detected for a modified application program. The program had allocated at least one CPI-CI conversation when insufficient storage for a buffer queue element (BQEL) was detected. This abend is the same as ABENDU2479 except it is not retriable and the input message is discarded. CPI communications driven application programs are always non-retriable.

Analysis
ABENDU0126 is a pseudoabend, set by module DFSTMR00 and issued by modules DFSECP10 and DFSECP20.

ABENDU0127

DFSPCC20

Explanation
An implicit commit for a CPI communications driven program failed. The implicit commit was attempted because the CPI communications driven program terminated without issuing either a CPI-RR COMMIT(SRRCMIT) or a BACKOUT(SRRBACK) call and IMS managed resources were in an uncommitted state. All of the uncommitted changes were backed out.

Analysis
ABENDU0127 is a user abend, issued by DFSPCC20.

ABENDU0128

DFSECP10, DFSECP20

Explanation
An APPC/MVS ASSOCIATE call failed. This call is used to inform APPC/MVS where the LU6.2 conversation is to be processed.

Analysis
ABENDU0128 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.

ABENDU0129

DFSECP10, DFSECP20

Explanation
A RACF call to create a copy of an access control environment element (ACEE) failed. This call is used to establish the security environment for the dependent region.

Analysis
ABENDU0129 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 will indicate 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 will indicate 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 will indicate 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 will indicate that both the transaction and program were stopped.

ABENDU0130

DFSASK00

Explanation
A message processing program (MPP) region attempted to connect to a DBCTL region. This is an invalid connection because DBCTL does not allow MPP regions to connect to it.

The MPP gets a U130 pseudoabend.

Do not start an MPP that will attempt to connect to an IMSID that is associated with an active DBCTL region.

Analysis
ABENDU0130 is a pseudoabend, issued by modules DFSASK00 during the SIGNON attempt by an MPP region.

ABENDU0131

DFSASK00, DFSDASI0

Explanation
In the identify function of DFSASK00, a nonzero return code was received following an IMODULE GETMAIN request for SP230/SP252 storage for DFSSRB.

Analysis
This is a pseudoabend issued by module DFSASK00 and DFSDASI0.

ABENDU0132

DFSASK00

Explanation
A nonzero condition code was received following a store clock (STCK) instruction during IDENTIFY. This is probably a hardware or a system control program (SCP) problem.

Analysis
This is a pseudoabend issued by module DFSASK00.

ABENDU0134

DFSASK00

Explanation
A dependent region issued an IDENTIFY request when a control region shutdown was in progress. This is normal when a shutdown is in process.

Analysis
This is a pseudoabend issued by module DFSASK00.

ABENDU0135

DFSASK00

Explanation
An invalid request code was passed to DFSASK00. This is probably an IMS system error.

Analysis
This is a pseudoabend issued by module DFSASK00.

ABENDU0136

DFSASK00

Explanation
A second IDENTIFY request has been issued from the same dependent region. This is probably an IMS system error.

Analysis
This is a pseudoabend issued by module DFSASK00.

ABENDU0139

DFSASK00

Explanation
An error return code was received from the SYSEVENT SVC. This is probably an SCP or hardware problem.

Analysis
This is a pseudoabend issued by module DFSASK00.

ABENDU0140

DFSCPY00

Explanation
An application making a DL/I call was not identified.

Analysis
This is a pseudoabend issued by DFSCPY00. This abend is actually issued by DFSECP10 (for an MPP), or DFSECP20 (for a BMP).

Possible Cause
An IMS system error may have occurred.

ABENDU0141

DFSASK00

Explanation
Signon token not valid. The requester could not keep track of the token.

Analysis
This is a pseudoabend issued by module DFSASK00.

ABENDU0142

DFSRRA00

Explanation
The work unit blocks 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 MAXPST is reached.

Analysis
ABENDU0142 is a standard abend issued by DFSRRA00. The signon failed for lack of resources.

Possible Cause
There were not enough BCB blocks in one of the following blocks:

```
LCRE      OSWA
DPST      GMQW
XPST      D1WA
EPST      D2WA
```

ABENDU0143

DFSASK00

Explanation
Insufficient virtual storage is available for dependent region initialization.

Analysis
This is a pseudoabend issued by module DFSASK00 during SIGNON or CREATE-THREAD processing.

ABENDU0144

DFSASK00, DFSDASI0, DFSDASS0, DFSDASR0

Explanation
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.

The operator can 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.

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.

ABENDU0145

DFSASK00

Explanation
The caller of DFSASK00 is not a valid caller.

Analysis
ABENDU0145 is a pseudoabend detected by DFSASK00 and issued by one of the following modules: DFSRRA00, DFSPCC20, DFSRRC10, or DFSRRC40. ABENDU0145 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 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.

ABENDU0147

DFSASK00

Explanation
A SIGNOFF request has been received while a thread is still active. TERMINATE-THREAD is issued by SIGNOFF. This window can be found 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.

ABENDU0148

DFSASK00

Explanation
A nonzero return code was issued in response to an IMODULE GETMAIN or a QSAV request.

Analysis
This is a standard abend issued by module DFSASK00 during dependent region termination (SIGNOFF).

ABENDU0149

DFSASK00

Explanation
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.

ABENDU0150

DFSFSTM0, DFSRCQM0, DFSXDL00, DFSFDLI0, DFSFSCT0

Explanation

When either the control region, DBRC region, or the DL/I subordinate address space abends the other remaining regions terminate with ABENDU0150.

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 ABENDU0150.
- 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 ABENDU0150. CALLRTM is issued to the job step if the CTL TCB has not yet initialized.

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 ABENDU0150. 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 ABENDU0150. 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.
- ABENDU0150 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.

ABENDU0151

DFSSBI00

Explanation

DFSSBI00 was unable to read records from the //DFSCTL data set because of invalid data set attributes.

Analysis

ABENDU0151 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.

ABENDU0152

DFSSBCR0

Explanation

The Sequential Buffering (SB) COMPARE subroutine detected that the buffer content that the SB buffer handler wanted to return to the OSAM buffer handler did not match the contents of the block as stored on DASD.

Analysis

ABENDU0152 is a pseudoabend detected by the COMPARE subroutine of module DFSSBCR0. After detecting the mismatch, DFSSBCR0 calls module DFSSBSN0 to create a SNAP 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 on how to print both IMS and CICS log records, see the File Select and Formatting Print utility (DFSERA10) in *IMS Version 9: Utilities Reference: System*. 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.

ABENDU0154

DFSSBIO0

Explanation

An interface error between the OSAM access method and DFSSBIO0 prevented the OSAM access method from starting I/O.

Analysis

ABENDU0154 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 //DFSCTL data set) either to the IMS log as a X'67EF' log record, or to a data set chosen by the user. For information on how to format and print log records, see *IMS Version 9: Utilities Reference: System*.

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 ICHECKOS 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).

ABENDU0155

DFSSBHD0

Explanation

The Sequential Buffering (SB) test program (DFSSBHD0) detected an error during execution.

Analysis

ABENDU0155 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 *IMS Version 9: Messages and Codes, Volume 2*.

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.
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.

Key	Label	Description
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.
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.
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.

ABENDU0156

DFSCPY00

Explanation
An application making a DL/I call was in supervisor state.

Analysis
ABENDU0156 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 occurred.

ABENDU0157

DFSCPY00

Explanation
An application making a DL/I call was in the wrong key. The key of the caller is different from the TCB key.

Analysis
ABENDU0157 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.

ABENDU0158

DFSZDI20

Explanation

An unexpected error occurred when issuing the type 13 RDJFCB macro to obtain DD statement information for a GSAM data set at region initialization time.

Analysis

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.

ABENDU0160

DFSCPY00, DFSCPY50, DFSCPY70, DBFIRC10

Explanation

An application was not executing in an IMS dependent region. 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

ASCB

Thread token

Analysis

This is a pseudoabend when issued by modules DFSCPY00, DFSCPY50, and DBFIRC10. This 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')

ABENDU0166

DFSIRAC0

Explanation
IMS RACF initialization failed. See the possible reasons listed below by reason code.

Analysis
This 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:

4 CIMS
5 TIMS
6 IIMS
7 LIMS

Reg 15	***Meaning***
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

ABENDU0168

DBFERST0, DBFNDC00, DBFNRST0, DFSRLP00

Explanation
A problem was encountered during restart.

DBFERST0, DBFNRST0

Explanation
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.

Analysis
ABENDU0168 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 9: Diagnosis Guide and Reference* 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 already 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'DBFR'	NRST4086	A buffer containing an in-doubt CI or the current sequential dependent segment (SDEP) butter was logged at checkpoint; the buffer was too large for a single log record, and 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.
Reg2=C'DMHR'	NRST4080	Buffer definition (number of buffers, buffer size) on the restarting system does not match checkpointed information.
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'	EROC5920	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); or, corresponding RCTEs do not have the same routing code (field RCTECODE).

DBFNDC00

Explanation

A FIND request that was done for a Communication Name Table (CNT) name that is contained in a checkpoint record was unsuccessful.

Analysis

ABENDU0168 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 *IMS Version 9: Diagnosis Guide and Reference*.

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.

DFSRLP00

Explanation

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.

Analysis

ABENDU0168 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.

Code ***Meaning***

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'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, or the length of the checkpointed CTB is not equal to the loaded CTB.

X'0B' The count of checkpointed CLBs is not equal to the loaded count, or the length of the checkpointed CLB is not equal to the loaded CLB.

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 count of the checkpointed link control blocks (LCBs) is not equal to the loaded count, or 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, or 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 Block (VTCB) is not equal to the loaded count; or 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'18' This code is issued while trying to recover a VTCB from a X'4021' log record.

ABENDU0171

DFSRST00

Explanation

This abend results if FORCE security is in effect. One or more of the security tables could not be loaded correctly.

Analysis

ABENDU0171 is a standard abend issued by DFSRST00. The program status word (PSW) at entry-to-abend points to the instruction within label SEC abend from which the abend (SVC 13) is issued.

Register 12 in the abend SVRB registers is the base register. Other pertinent SVRB registers are as indicated below.

Within routine LDSECRTN, a call (BALR) is made to security table initialization, module DFSISMI0, and register 15 is checked for the return code. If FORCE was specified to force the loading of any security tables, the return code is interrogated to determine if ABENDU0171 should be issued. Label LDSERTN is entered from several different paths, but, when entered for the first time, a switch is set (RSTCTL4=X'10').

Register 15 contains one of the following return codes from module DFSISMI0:

Code	***Meaning***
X'00'	Normal completion
X'01'	User ID verify failure
X'02'	Force TRANS AUTH failure
X'04'	Password table load failure
X'08'	Terminal matrix load failure

Key	Label	Description
Reg1=abend completion code, X'000000AB' Reg7=BAL Reg9=address of SCD SCDCFLG1=X'60' Reg15=Return Code	SYSBLK59	Register 15 contains the type of abend. The SCDCFLG1 is tested for security options. If the option specified matches the return code returned from DFSISMI0, ABENDU0171 is issued. If the return code in register 15 is nonzero, the DFS171A message is issued. Then the FORCE security table load indicators in SCDCFLG1 and SCDSECFN are tested to determine whether any table for which FORCE was specified was not successfully loaded; if yes, this abend is issued.

ABENDU0172

DFSRST00

Explanation

There is insufficient space in the work area pool (WKAP) to satisfy restart work space requirements.

Analysis

ABENDU0172 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.

ABENDU0175 and ABENDU0176

DFSFDLY0

Explanation

These abends are issued in module DFSFDLY0 in response to errors detected by the following modules:

DFSFDLN0 Allocate an OLDS or SLDS

DFSFDLO0 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 of restart-read

DFSFDLZ0 Used in emergency restart to switch between OLDS read and SLDS read

U0175: 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.

U0176: 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.

Analysis

ABENDU0175 and ABENDU0176 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 U0175 and U0176 abends are accomplished by calling the common ABEND subroutine in csect DFSFDLY2 in module DFSFDLY0. Register 14 contains the caller's return address, 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, RLWWORK. RLWWORK 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 ABENDU0175 and ABENDU0176:

Reg8 Second base register for module (if needed)

Reg9 RLWWORK - 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

RLWWORK is the key data area for all of the logger restart-read modules and is mapped by DSECT macro DFSFRLWA. The following fields within RLWWORK 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.

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.

DFSFDLN0

Explanation

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.

Analysis

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 DFS0737I.
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.

DFSFDLO0

Explanation

Module DFSFDLO0 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).

Analysis

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.
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.

Key	Label	Description
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.

DFSFDLP0

Explanation

Module DFSFDLP0 reads system log data sets (SLDS) during warm start or emergency restart.

Analysis

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.
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.

DFSFDLQ0

Explanation

Module DFSFDLQ0 contains all of 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.

Analysis

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.
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.

DFSFDLR0

Explanation

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 may or may 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.

Analysis

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.
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 a X'42' (or X'4301') log record, a zero length record was found.

DFSFDLU0

Explanation

Module DFSFDLU0 terminates the latest OLDS from the WADS.

Analysis

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.

Key	Label	Description
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.

DFSFDLX0

Explanation

Module DFSFDLX0 is the driver for the OLDS read process.

Analysis

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 will abend. 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.

DFSFDLY0

Explanation

Module DFSFDLY0 contains all of the subroutines commonly used by the restart-read modules.

Analysis

Key	Label	Description
Reg15=RLWTCF 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=RLWTCF	LDG200	The OLDSCALC subroutine received a nonzero return code from TRKCALC.
Reg15=RLWTCF	OLPNT	The OLPNT subroutine received a nonzero return code from TRKCALC.

Key	Label	Description
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.

DFSFDLZ0

Explanation
Module DFSFDLZ0, during warm start or emergency restart, performs either OLDS read or SLDS read.

Analysis

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.

ABENDU0182

DBFMSRB0, DBFVOCI0, DBFVSOW0, DBFMIOS0, DBFFORI0, DBFUMAI0, DBFNCBS0, DBFMFL10

Explanation
A condition that should not occur was detected in one of the Fast Path modules.

Analysis
Register 15 contains the diagnostic code: bytes 0-2 contain the module ID and byte 3 contains the subcode.

For module ID `FOR` (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 `MIO` (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 `VOC` (DBFVOCI0), 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.

07 Address of L56X block in DSML DSMEL56X is zero.

For module ID `VSO` (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.

ABENDU0195

DBFARD30, DBFHSRT0

Explanation
A condition that should not occur was detected in one of Fast Path modules.

Analysis
Register 15 contains the ABENDU0 195 subcode. 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

ABENDU0203

DFSERA10

Explanation
While running the File Select and Formatting Print Program utility, DFSERA10, the parameter list constructed from the OPTION control statements was found to have been modified.

Analysis
ABENDU0203 is a standard abend issued from the Format Print utility, DFSERA10. The registers in the abend SVRB are valid and, together with the program status word (PSW) at entry-to-abend, should be used to aid in identifying the problem.

Register 6 should contain a pointer to the first list element. If it does not, an error occurred during initialization of the option process.

Refer to *IMS Version 9: Utilities Reference: System* for the proper format of the OPTION statement in the information about utility control statements.

Key	Label	Description
Reg6=pointer to first list element Reg11=address of routines	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 ABENDU0203.

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 program's legitimate address space.

ABENDU0204

DFSERA50

Explanation
The output data set for the DL/I call trace exit routine from DFSERA10 could not be opened.

Analysis
This is a standard abend issued by module DFSERA50.

Possible Cause
User error in TRCPUNCH DD statement.

ABENDU0206

DFSPCC30

Explanation
The IMS.PSBLIB or IMS.DBDLIB library could not be opened.

Analysis
This is a standard abend issued by module DFSPCC30. Register 1 contains the completion code X'800000CE'. Register 2 contains the address of PXPARMS; label PCDCBADR in PXPARMS contains the DCB address.

Key	Label	Description
Reg1=X'800000CE'	BADOPEN	OPEN failure for PSBLIB or DBDLIB.

ABENDU0209

DFSDLBL0, DFSDPSB0

Explanation
A region type of ULU or UDR was specified but the DBD name specified was a logical DBD, or the access method called was invalid.

Analysis
ABENDU0209 is a standard abend issued by module DFSDLBL0, when module DFSDPSB0 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 DFSDPSB0, 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 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

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.

ABENDU0214

DFSDLBL0, DFSDPSB0

Explanation

The parm field specified a region type of ULU or UDR. The program name supplied is not authorized to use this region type.

Analysis

ABENDU0214 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 Reg8=authorized program list address RCPGM ¬= APLPGM APLPGM=X'FF'	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.

ABENDU0215

DFSFDLL0

Explanation

The IMS disk logical logger, DFSFDLL0, has encountered an internal error. Register 15 contains the following return codes:

Code	*Meaning*
X'04'	Invalid buffer address
X'08'	Incorrect post code

Analysis
ABENDU0215 is a standard abend issued by module DFSFDLL0.

The following indicates the contents of the registers:

Register	Contents
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.

ABENDU0216

DFSULG10

Explanation
The DFSULTR0 utility detected a bad return code from DBRC and DBRC=yes was requested.

Analysis
ABENDU0216 is a standard abend issued by module DFSULG10.

The following indicates the contents of the registers:

Register	Contents
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.

ABENDU0219

DFSICLD0

Explanation
The display command processor, DFSICLD0, detected an unrecoverable error.

Analysis
ABENDU0219 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.

ABENDU0219 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 list below 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 or LTERM	DFSIDP40
X'14'	/DISPLAY DB	DFSIDP50
X'18'	/DISPLAY LINE	DFSIDP60
X'1C'	/DISPLAY ASSIGNMENT	DFSIDP70
X'20'	/DISPLAY CONVERSATION	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
X'60'	/DIS TRACE	DFSIDPK0
X'64'	/DIS HSSP	DBFPDHS0
X'68'	/DIS TIMEOVER	DFSIDPL0
X'6C'	/DISPLAY APPC/DESCRIPTOR	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	DFSYDPD0
X'88'	/DISPLAY FDR	DFSIDPZ0

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.

ABENDU0220

DFSIIRS0

Explanation

Initialization module DFSIIRS0 could not locate a control block. This is an unrecoverable error.

Analysis

When system definition is performed using LGEN, the communication name table (CNT), communication line block (CLB), and local link name block (LNB) are searched to resolve block pointers. Pointer resolution occurs between the following blocks:

- BTAM CTBs and CLBs
- SPQBs and CNTs
- RCNTs and LNBs

ABENDU0225

DFSUDUI1

Explanation

The Database Image Copy 2 utility (DFSUDMT0) issues this abend when it determines that invalid information has been received from DFSMSdss™. Register 15 contains a reason code that further identifies the problem.

The utility also issues message DFS3144A which identifies the database data set for which the problem occurred. 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 will be 8 or higher.

Programmer Response

Check ADRnnn messages issued by DFSMS™ to aid in diagnosis. See *z/OS System Messages, Volume 1* for a description of ADR messages. If you cannot determine the solution, contact the IBM Support Center. Have the abend documentation available.

Analysis

ABENDU0225 is a standard abend issued by module DFSUDUI1. Register 15 contains the reason code for the termination. The reason codes have the following meanings:

4 The output volume serial number passed by DFSMSdss on the exit option 26 invocation is invalid.

ABENDU0230

DFSRLP00

Explanation

ABENDU0230 is issued by module DFSRLP00 for two reasons.

1. An /ERE BLDQ command has been issued and the log used for restart reached EOF before an end of checkpoint (4099) record was read.
2. During restart an attempt was made to get a work area from the MAIN (WKAP) storage pool, and storage could not be obtained.

Analysis
ABENDU0230 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.

ABENDU0231

DFSRLP00

Explanation
IMS was not able to resynchronize with the CQS subsystem.

Analysis
ABENDU0231 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 *IMS Version 9: Diagnosis Guide and Reference*.

01 The operator replied `ABORT` to message DFS3909.

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 start or emergency restart 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 start or emergency restart IMS without CQS, buy 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 start or emergency restart 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 start or emergency restart 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 start or emergency restart 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 ABENDU0231 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.

- 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.

ABENDU0233

DFSRLP00

Explanation

IMS was not able to resume with the current resource structure usage.

Analysis

The abend subcode provides the reason for the failure:

01 An attempt was made to warm start 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.

ABENDU0240

DFSPCC20, DFSECP10

Explanation

A message processing application program exceeded the allowable execution time (set at IMS system definition) in a message processing region. In the case of a BMP, this abend indicates that the value specified in ″CPUTIME=″ has been exceeded.

Analysis

ABENDU0240 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 backchain ends in the eyecatcher 'F1SA'. Use the Linkage Stack created by the BAKR instruction to obtain the application registers and PSW at the time of abend.

ABENDU0242

DFSASK00

Explanation

This abend is issued for the following reasons:

- The DIRCA work-size parameter on the dependent region EXEC statement is not large enough 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.

Analysis

The error was detected by module DFSASK00.

ABENDU0249

DFSDDLT0

Explanation
This abend is issued when any status code other than a STATUSGA or status blank is returned to the DL/I test program, DFSDDLT0, during its internal end-of-job status calls.

Analysis
ABENDU0249 is issued by DFSDDLT0, the DL/I test program, in a batch environment. After all DFSDDLT0 control statements are processed, DFSDDLT0 issues internal end-of-job status calls. DFSDLA00, the DL/I Call Analyzer, calls DFSDVBH0, the buffer handler, for OSAM and VSAM buffer statistics. The return code (register 15) from DFSDVBH0 is non zero.

DFSDLA00 sets DBPCBSTC to 'GE' (STATUSGE) and passes it back to DFSDDLT0. DFSDDLT0 checks DBPCBSTC for blanks or 'GA'. When DBPCBSTC is not blanks, ABENDU0249 is issued, if the ABU249 option was coded. If the ABU249 option was not coded, printing of the output for the status call is bypassed.

In an online environment, the job terminates with ABENDU0479. DFSDLA00 issues the abend and does not return to DFSDDLT0 to finish the status calls.

Possible Cause
The user coded ABU249 on the DFSDDLT0 option statement, and an invalid status code was received during the DFSDDLT0 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 ABENDU0249 to be issued from DFSDDLT0, if the ABU249 option is coded.

ABENDU0250

DFSDDLT0

Explanation
The DL/I test program issued a conditional GETMAIN macro for an area to be used for the segment I/O area. If running online, the size requested was 32,732 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 DFSDDLT0.

ABENDU0251

DFSDDLS0, DFSDDLT0

Explanation
IMS is 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
ABENDU0251 is a standard abend issued from DFSDDLS0 (named DFSDDLT0 after being link-edited 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.

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 DDxxxxxxxx” 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 DDxxxxxxxx” to the print data set. A branch is then taken to RETURN at label EPILOGUE to abend.

ABENDU0252

DFSDDLS0, DFSDDLT0

Explanation
An abend control statement requesting an abnormal termination has been read.

Analysis
While running the DL/I test program for batch, DFSDDLS0 (named DFSDDLT0 after being link-edited with the language interface module, DFSLI000), a control statement requesting an abend has been encountered.

ABENDU0252 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'800000FC' 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.

ABENDU0253

DFSDDLS0, DFSDDLT0

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

ABENDU0253 is a standard abend issued from the DL/I test program for batch, DFSDDLS0 (DFSDDLT0 after being link-edited 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'800000FD' 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 "DBDxxxxxxxx DOES NOT EXIST—ABEND 253 'PRINTDD'". A branch is taken to RETURN to label EPILOGUE to abend.

Possible Cause

The user has coded a PCB name in the status control statement that is not one of the PCBs in the PSB named on the //EXEC statement of the JCL.

ABENDU0254

DFSDDLS0, DFSDDLT0

Explanation

The DL/I test program 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

ABENDU0254 is a standard abend from the DL/I test program for batch, DFSDDLS0 (named DFSDDLT0 after being link-edited 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'800000FE' 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.

ABENDU0255

DFSDDLT0

Explanation
A nonzero return code was received from the buffer handler.

Analysis
ABENDU0255 is a standard abend issued by module DFSDDLT0. 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
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

Possible Cause
An invalid value was specified for RCF, SGN, TRN or ISIS. Message DFS0255I was returned along with this abend.

ABENDU0256

DFSCRPV0

Explanation
ABENDU0256 is a standard abend issued by module DFSCRPV0.

Analysis
This abend is a standard abend issued from the IMS/VS communication restart processor (DFSCRPV0) 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.

ABENDU0257

DFSCRPV0

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.

Analysis
ABENDU0257 is a standard abend issued by module DFSCRPV0.

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.

ABENDU0258

DFSICLH0, DFSRCP30

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 (DFSRCP30).

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.

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.

ABENDU0260

DFSECP10, DFSECP20, DFSPR000

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).

Analysis
ABENDU0260 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.

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.

DFSECP20

Analysis
ABENDU0260 is a standard abend that can be issued by the BMP environment controller, DFSECP20. If issued here, 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.

DFSPR000

Analysis
ABENDU0260 is a standard 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 U0260 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
```

ABENDU0261

DFSCDLI0, DFSECP10, DFSECP20, DFSPR000

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.

Analysis
ABENDU0261 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 U0261 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.

DFSCDLI0

Analysis
ABENDU0261 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

DFSECP10, DFSECP20

Analysis
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.

DFSPR000

Analysis

ABENDU0261 is a standard abend that can be issued by the batch application program request handler, 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 U0261 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.

ABENDU0262

DFSECP20

Explanation

A batch message processing (BMP) program issued a DL/I call after issuing a DL/I checkpoint call or synchronization 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.

ABENDU0263

DFSDPRH0, DFSDCPY0

Explanation

This pseudoabend is issued when a Coordinator Controller (CCTL) thread in a DB Control (DBCTL) environment issues an invalid call.

Analysis

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.

ABENDU0265

DFSIAF20

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.

Analysis
ABENDU0265 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 above (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
Coding change or a user modification.

ABENDU0271

DFSZDC00

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 in *IMS Version 9: Messages and Codes, Volume 2* for details about the data set.

ABENDU0272

DFSZDC00

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.

ABENDU0273

DFSZDC00

Explanation
An error was detected while repositioning a GSAM data set. GSAM is abnormally terminated because position in the affected data set is unpredictable.

Analysis
Print the GSAM control block DSECTs (member name IGLI in IMS.SDFSMAC) for use in analyzing the GSAM control blocks. Use the “GSAM Control Blocks Dump” on the IMSERR or SYSPRINT output to determine which GSAM PCB has a nonblank status code.

ABENDU0274

DFSSBIO0

Explanation
An unexpected interface error occurred between DFSSBIO0 and its caller.

Analysis
ABENDU0274 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.)

ABENDU0275

DFSSBSN0

Explanation
IMS was not able to acquire a work area needed to create a SNAP dump of the Sequential Buffering (SB) control blocks and areas.

Analysis
ABENDU0274 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.

ABENDU0300

DFSUCF00, DFSUCP40

Explanation

ABENDU0300 is issued in the utility control facility (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 an ABENDU0300 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 an ABENDU0300 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 an ABENDU0300, 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 ABENDU0300 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.

ABENDU0302

DFSURDB0

Explanation

An unidentifiable error occurred during execution of the Database/Data Set Recovery Utility program, DFSURDB0.

Analysis

ABENDU0302 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 U0302 abend, an IMS message will be issued prior to 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.
Reg1=Completion code Reg10=BAL to Message Generator	RDBDBDLD #@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.

Key	Label	Description
Reg1=Completion ode Reg10=BAL to Message Generator	RDBDDBDLD #@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	RDBDDBDLD #@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 X'04' 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 IDCUSTRT 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	RDBCUOPN	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	RDBDUHDR (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	RDBDUHDR #@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.
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.

Key	Label	Description
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.
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.

Key	Label	Description
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.

ABENDU0303

DFSERA10

Explanation

The parameter list constructed from the OPTION statements has been modified. (See *IMS Version 9: Installation Volume 2: System Definition and Tailoring* for the modified format). A possible cause is an error in the user exit routine that caused it to address and modify storage outside the program's legitimate address space.

Analysis

ABENDU0303 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, "ELEMENT LIST ERROR," 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.

Refer to *IMS Version 9: Utilities Reference: System* for the format of the OPTION statement in the information about utility control statements.

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 of these tests fail, a branch is taken to the routine at label SEL04 to abend.

ABENDU0305

DFSURIO0

Explanation

An incomplete spanned record was detected while reading a VBS sequential data set. 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

ABENDU0305 only occurs for read requests.

ABENDU0305 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

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.

ABENDU0306

DFSUC350

Explanation

An error was detected by Change Accumulation utility while trying to build the output change accumulation record. The length of data found in an element exceeds the maximum allowable database block 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	MERGPUT2	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.

ABENDU0310

DFSPRERR

Explanation
An error was detected by Partial Database Reorganization.

Analysis
This 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. (Refer to *IMS Version 9: Messages and Codes, Volume 2* for an explanation of the message.) Register 5 contains the address of the PDBR common area (COMAREA), which contains addresses of other PDBR blocks.

ABENDU0311

DFSPRSER

Explanation
An error was detected by Surveyor.

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. (Refer to *IMS Version 9: Messages and Codes, Volume 2* for an explanation of the message.) Register 5 contains the address of the Surveyor common area (CMAREA), which contains addresses of other Surveyor blocks.

ABENDU0315

DFSIRST0, DFSRLP00, DFSXSTM0

Explanation
IMS received a nonzero return code from IMSAUTH SVC function.

Analysis
Register 15 contains the IMSAUTH return code. For a description of these return codes, see the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

ABENDU0316

DFSDMAW0

Explanation

An unexpected or unrecoverable error was detected during coupling facility services initialization or subsequent execution. 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.
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.

DFSDCFC0

Key	Label	Description
Reg15=X'11'		IOSB integrity error.

ABENDU0317

DFSDCFR0

Explanation

One of the following failures occurred on the XRF alternate system:

- Coupling facility connection failure
- Structure rebuild failure

Analysis

ABENDU0317 is issued by the GOSTPDS routine in DFSDCFR0. Error messages issued before this abend indicate the reason for the failure.

ABENDU0318

DFSDVBI0

Explanation

IMS coupling facility services initialization failed because either the required z/OS, DFSMS, or both release levels were not used.

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.

ABENDU0322

DFSSBTD0

Explanation

An unexpected condition occurred in the OSAM sequential buffering routine 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.

ABENDU0347

DFSURGU0

Explanation

A severe error condition was detected during Hierarchical Direct Access Method (HDAM) Reorganization Unload, and a SYSUDUMP or a 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

Analysis

ABENDU0347 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 ABENDU0347. 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.

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.

Key	Label	Description
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.
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.

ABENDU0355

DFSURGL0

Explanation
A severe error condition was detected during execution of the HDAM Reorganization Reload utility, and a SYSUDUMP or a SYSABEND DD statement was provided. This is possibly an internal system error.

Analysis
ABENDU0355 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 U0355 abend occurs.

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.

ABENDU0359

DFSURUL0, DFSURRL0

Explanation

The abend option was selected for the HISAM Reorganization Unload utility or Reload utility, and a severe error occurred.

Analysis

ABENDU0359 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 U0359 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.

ABENDU0369

DFSDVBH0

Explanation

The buffer handler module, DFSDVBH0, has detected an internal error. The caller provided the buffer handler with an invalid RBA for either the A-J data set or the M-V data set of a HALDB.

Analysis

At the time of the abend, both register 4 and register 5 will indicate what the internal error might be. Both registers will contain either a 0 or 1. The number 0 indicates that the I/O is for one of the A-J data sets. The number 1 indicates that it 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 zero. The RBA is always an odd value for the M-V data sets; its flag byte, DMBORFL2, of the corresponding DMB (M-V DMB), set is one.

If a 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.

ABENDU0390

DFSXLUM0

Explanation

Module DFSXLUM0 received a nonzero return code from a lower-level system service.

Analysis

ABENDU0390 is a standard abend issued by module DFSXLUM0 during control region initialization. SVRB 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:

Code	*Meaning*
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 (CSVxxxI) for details.

X'28' The data set SYS1.CSSLIB is not APF-authorized.

ABENDU0391

DFSXRM00

Explanation
Module DFSXRM00 received a nonzero return code from a lower level system service.

Analysis
ABENDU0391 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:

Code	*Meaning*
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.

ABENDU0392

DFSXTMC0

Explanation
Module DFSXTMC0 received a nonzero return code from a lower level system service.

Analysis
ABENDU0392 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:

Code	*Meaning*
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.

ABENDU0393

DFSXALM0

Explanation
Module DFSXALM0 received a nonzero return code from a lower level system service.

Analysis
ABENDU0393 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:

Code	*Meaning*
X'04'	DFSBCB GET QSAV failed. Register 8 contains the DFSBCB return code.

ABENDU0394

DFSXALC0

Explanation
Module DFSXALC0 received a nonzero return code from a lower level system service.

Analysis
ABENDU0394 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:

Code	*Meaning*
X'04'	DFSBCB GET QSAV failed. Register 8 contains the DFSBCB return code.

ABENDU0396

DFSXXCF0

Explanation
Module DFSXXCF0 received a nonzero return code from a lower level system service.

Analysis
ABENDU0396 is a standard abend 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:

Code	*Meaning*
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.

ABENDU0402

DBFIRC10, DFSDLPR0, DFSRRA00

Explanation

This abend is issued when the intersubsystem interface module encountered a problem.

DBFIRC10

Analysis

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).

DFSDLPR0

Analysis

ABENDU0402 is a standard abend set at label AB402 in module DFSDLPR0.

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.

DFSRRA00

Analysis

ABENDU0402 is a standard abend set at label RAAB0402 in module DFSRRA00 on return from DFSISI00.

Key	Label	Description
Reg1=X'C0000192'	RAAB0402	
SSOB+X'4C'='SVC#'	(ISIIDF in DFSISI00)	The identifying region's TYPE 2 SVC number does not match the IMS control region.

Key	Label	Description
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)	DFSCPY00 PC return code is non-zero.

ABENDU0403

DFSAOS70

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.

ABENDU0407

DFSAOS70, DFSILTA0

Explanation

The Log Transaction Analysis utility is unable to obtain sufficient storage for its queues.

Analysis

ABENDU0407 is a standard abend issued by the log transaction analysis module, DFSILTA0. The program status word (PSW) at entry-to-abend point to the instruction within label GETQUE from which the abend (SVC 13) is issued. Message DFS0407I—REGION TOO SMALL—accompanies this abend.

Several inline routines branch to GETQUE in order to set up the queues or a Q1ENTRY. If the GETMAIN supervisor call (SVC) issued within label GETQUE is unsuccessful, and no storage can be freed, the abend results.

Register 14 through register 8 are stored on entry to the routine at GETQUE at label GETQSV1. In this instance, register 1 contains the address of the GETMAIN parameter list. Abend processing overlays the contents of register 1 with the abend completion code, X'80000197'.

Key	Label	Description
Reg4=primary base register Reg5=secondary base register	GETQUE	DFSILTA0 has been unable to obtain sufficient storage for its queues, either using a GETMAIN (SVC 4), or by freeing storage elsewhere, so the DFS0407I message is sent and the abend is issued.

Possible Cause
The message region size may be too small.

ABENDU0411

DFSILTA0

Explanation
An OPEN operation failed for a data set that is defined by the DDNAME that is noted in message DFS0411I.

Analysis
ABENDU0411 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 U0411 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' Reg11=address of DCB for which OPEN failed	FNDSTART and FNDLOG	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.

ABENDU0413

DFSILTA0

Explanation
The limit of the internal table of 255 application programs running simultaneously has been exceeded. This is an IMS system error.

Analysis
ABENDU0413 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 log, 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.

ABENDU0415

DFSILTA0

Explanation
An invalid parameter was detected on an EXEC statement.

Analysis
Message DFS0408I or DFS0409I accompanies this standard abend, and identifies the invalid parameter.

ABENDU0427

DFSDVSM0, DFSDVBH0

Explanation
A logical error occurred while processing a VSAM database.

Analysis
ABENDU0427 is a pseudoabend issued by DFSDVSM0 and DFSDVBH0. This is a standard abend issued when the user codes DUMP=YES on the DL/I buffer options statement.

Module DFSDVBH0 intercepts the pseudoabend ABENDU0427, and changes it to a standard abend.

A pseudoabend dump of 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 buffer handler trace table in *IMS Version 9: Diagnosis Guide and Reference*. 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 byte equal to X'0427'. This entry is formatted as follows:

Disp.	ABENDU0427		Trace Entry
X'00'		DLTRFCTN	X'AB'
X'02'	ENTRY02		X'0427'
X'04'	ENTRY1	PSTFNCTN	Buffer handler function code.

Disp.	ABENDU0427		Trace Entry
X'05'	ENTRY1+2	TRACK SEQ. NO.	Offset to abend
X'08'	ENTRY2		
X'0A'	ENTRY2+2	DSGINDA	
X'0B'	ENTRY2+3	DSGINDB	
X'0C'	ENTRY3		RPL Request
X'0D'	ENTRY3+1	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 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.

Use *z/OS DFSMS Macro Instructions for Data Sets* for a description of the VSAM return and error codes. Use *IMS Version 9: Diagnosis Guide and Reference* for a description of the buffer handler function codes.

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 two 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, you should add the parameter DUMP=YES to the VSAM OPTIONS statement, which is described in *IMS Version 9: Installation Volume 2: System Definition and Tailoring*. This causes an abend to be issued instead of a pseudoabend.

Key	Label	Description
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	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 using coupling facility services. This caused the abend to be issued.
Reg14=BAL	GET5030	An invalid VSAM buffer prefix was found while testing for buffer validity using coupling facility services. This 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.

Key	Label	Description
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.

ABENDU0428

DFSSBMP0

Explanation
A BMP or a Fast Path region (IFP) step could not be initiated because the PSB was not found.

Analysis
ABENDU0428 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.

ABENDU0430

DFSDVBI0

Explanation
The DL/I database buffering services function cannot be initialized. Message DFS0430I has been issued; the reason code in the message further defines the reason for failure.

Analysis
ABENDU0430 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.

Code	***Meaning***
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:

Code ***Meaning***

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 ¬= 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 *requested.* If this switch is not set, but the switch is on indicating VSAM subpools are *required,* 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 ¬= X'80'	BLDVP100	The DVBISW switch is tested—if the buffer handler pool was *not* built, and if VSAM subpools were requested but not built, a branch is taken to label MDLOD040, and then to label ABEND430 to abend. If the buffer handler pool was *not* built, and if VSAM subpools were requested *and* built, a VSAM DLVRP macro is issued to delete the pools, and a branch is taken to label MDLOD040, then to label ABEND430 to abend. In both cases, IMS message DFS0430I (reason code 09) accompanies the abend.
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.

Key	Label	Description
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.

ABENDU0432

DFSSBMP0

Explanation

A batch message processing (BMP) program step could not be scheduled. The requested PSB was defined as a teleprocessing program with the parallel option specified, while the BMP program step is message driven.

Analysis

ABENDU0432 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.

ABENDU0436

DFSSBMP0, DFSPCC20

Explanation

A BMP step could not be scheduled because an input symbolic queue name was invalid.

Analysis

ABENDU0436 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.

DFSPCC20

Explanation
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.

ABENDU0437

DFSASK00

Explanation
If security maintenance utility (SMU) is active, then either the application group name (AGN) or the resources specified are not valid for this dependent region. If resource access security (RAS) is used instead of SMU, then there are IMS resources (transaction, PSBs, and LTERMs) that are not valid for this dependent region. For more information on AGN security and RAS, see *IMS Version 9: Administration Guide: System*.

Analysis
ABENDU0437 is a pseudoabend issued by module DFSASK00.

Possible Cause
User error.

ABENDU0438

DFSDASP0

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.

Possible Cause
User error.

ABENDU0440

DFSSBMP0

Explanation
A BMP step could not be scheduled because the input symbolic queue name was not a transaction code.

Analysis
ABENDU0440 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.

ABENDU0442

DFSDVSMO

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 ABENDU0442.

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
ABENDU0442 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 MSGDFS0442A, 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, ABENDU0442 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 ABENDU0442 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 ABENDU0845 is issued and a return to caller is made.
	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.

ABENDU0444

DFSSBMP0

Explanation
A BMP step could not be scheduled because an output symbolic queue name was invalid.

Analysis
ABENDU0444 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.

ABENDU0448

DFSSBMP0

Explanation
A BMP step could not be scheduled because the input transaction code is a remote SMB.

Analysis
ABENDU0448 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.

ABENDU0451

DFSDBH40, DFSDISM0, DFSDVBH0, DFSDVSM0

Explanation
A physical I/O error has been detected on a database data set.

Analysis
Message DFS0451I or DFS0451A is printed, 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 ABENDU0451, 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 ABENDU0451 is issued from module DFSDVBH0 at label PURGE090. This happens even though the operator responds “CONT” to message DFS0451A.

ABENDU0452

DFSSBMP0

Explanation
A BMP step could not be scheduled because the transaction code specified in the PARM field's fourth positional operand, which is on the EXEC control statement, has been stopped or locked by either a command or by a prior program failure.

Analysis
ABENDU0452 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.

ABENDU0453

DFSSBMP0, DFSPCC20

Explanation
The IMS scheduler has determined that a BMP region cannot be scheduled because the transaction code specified in the fourth positional operand of the PARM field on the EXEC statement is a conversational transaction or an LU 6.2 dynamically built transaction.

Analysis
ABENDU0453 is a pseudoabend issued by DFSPCC20 and DFSSBMP0.

Key	Label	Description
Reg1=X'800001C5'	BMPRSPSB	A conversational transaction was scheduled in a BMP.

ABENDU0454

DFSPCC20

Explanation
A BMP or IFP that was prestarted on the alternate IMS in an XRF environment has been terminated either by a `/STOP BACKUP` command on the alternate IMS, or by a `/CHE FREEZE` command on the active IMS. Scheduling did not complete.

Analysis
ABENDU0454 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.

DFSRRA00

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.

Analysis
ABENDU0454 is a pseudoabend passed by module DFSASK00 to module DFSRRA00, which issues the abend.

ABENDU0456

DFSSBMP0

Explanation
A batch-message processing (BMP) program step or a Fast Path (IFP) region, could not be initiated: 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.

  Examples of messages to look for during initialization are DFS563I and DFS830I.
- The PSB referenced was defined as DOPT and was found in the first concatenation of ACBLIB.
- A Fast Path PCB referenced a DBD that could not be found in the system.
- A 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.

Analysis
ABENDU0456 is a pseudoabend issued from module DFSSBMP0.

Message DFS554A accompanies this abend.

Key	Label	Description
	BMPTYPOK 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 usually occurs when compressing the ACBLIB while online IMS is running.

ABENDU0457

DFSSBMP0

Explanation
A BMP or Fast Path (IFP) region was started for a PSB already scheduled in another region.

Analysis
ABENDU0457 is a pseudoabend issued from module DFSSBMP0.

Key	Label	Description
	BMPTYPOK	The PSB is already scheduled in another region, and no parallel scheduling was specified.

ABENDU0458

DFSSBMP0

Explanation
A BMP or Fast Path IFP region step could not be scheduled, because one of the databases used by the PSB has been stopped by a prior program failure. The PSB was specified in the third positional operand of the PARM field on the EXEC control statement.

Analysis
ABENDU0458 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=” Field (that is, the second positional parameter) has been stopped or has not been defined as a DEDB in the DBDGEN.

Key	Label	Description
	BMPTYPOK	The PSB failed scheduling because the database was stopped.

ABENDU0462

DFSASK00

Explanation
An application program was scheduled in a message region. The program terminated without successfully issuing a GET UNIQUE (GU) for an input message.

Analysis
ABENDU0462 is a standard abend detected by module DFSASK00 in the normal thread termination routine.

Correct the application program so that a GET UNIQUE is issued.

Key	Label	Description
Related log records	Type X'08'	The transaction and program were scheduled.
	Type X'07'	Application termination record: • If the field copied from DLRGUMES is greater than zero, a GU was issued by the application program. • If the field copied from DLRGU1 is greater than zero, a GUs was successful. ABENDU0462 indicates that this field was equal to zero at termination time.

ABENDU0474

DBFSYN00, DBFIRC10, DFSASK00, DFSCPY00, DFSDASC0, DFSDCPY0, DFSDLTR0, DFSDVBH0, DFSTMAD0, DFSUICC0

Explanation

An application program was terminated because the operator entered the /STOP REGION ABDUMP command.

Attention: ABENDU0474 can also occur if you issue a /STOP REGION CANCEL command that is preceded by a /STOP REGION ABDUMP command.

DBFSYN00

Analysis

ABENDU0474 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.

DBFIRC10, DFSASK00, DFSCPY00, DFSDASC0, DFSDCPY0, DFSTMAD0

Analysis

ABENDU0474 is a pseudoabend issued by DBFIRC10, DFSASK00, DFSCPY00, DFSDASC0, DFSDCPY0, and DFSTMAD0 after detecting a /STOP REGION ABDUMP command.

DFSDLTR0, DFSUICC0

Analysis

ABENDU0474 is an abend issued by DFSDLTR0 and DFSUICC0 after detecting a /STOP REGION ABDUMP command.

DFSDVBH0

Analysis

ABENDU0474 is an abend issued by DFSDVBH0 after detecting a /TERMINATE OLREORG OPTION(ABORT) command.

ABENDU0476

DBFIRC10, DFSCDLI0, DFSCPY00, DFSDLA00, DFSECP10, DFSECP20, DFSPR000

Explanation

An invalid PCB or AIB address, or no address, was passed in a DL/I call parameter list.

DBFIRC10

Analysis

ABENDU0476 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 user-written application program is in error.

DFSCDLI0

Analysis

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 user-written application program is in error.

DFSCPY00

Analysis

ABENDU0476 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.

DFSDLA00

ABENDU0476 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.

Key	Label	Description
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''	PSEUDOCB	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.

DFSECP10

Analysis

ABENDU0476 is also a pseudoabend detected by the interregion copy facility module, DFSCPY00, 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.

ABENDU0476 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.

DFSECP20

Analysis

ABENDU0476 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
A user-written application program made a GSAM call, but passed an invalid GSAM PCB address. When DFSPR0X0 detects the abend, a user-written application has issued a DL/I call pointing to a parameter list with a PCB address equal to zero.

DFSPR000

Analysis
ABENDU0476 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
A user-written application program made a GSAM call, but passed an invalid GSAM PCB address.

ABENDU0477

DFSDLA00, DFSDLAS0

Explanation
Since the primary area to build FLD blocks was not large enough, an IGETBUF macro was issued to obtain an 8192-byte area for FLD blocks. The area was not available. In batch processing, a GETMAIN is done to obtain the space; for online processing, the space is obtained from the database working pool.

Analysis
ABENDU0477 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'000001DD'.

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.

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 PSTRTCDE, when compared, indicates that the RBN was beyond the data set. The abend is issued.

Possible Cause
For batch, the address space was not large enough. For online, the database working pool was too small.

ABENDU0478

DFSDLAS0

Explanation
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.

Analysis
ABENDU0478 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.

ABENDU0479

DFSDLA00

Explanation
This abend is issued when a nonzero return code is returned from a STAT call to the buffer handler to get buffer pool statistics.

Analysis
ABENDU0479 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).

ABENDU0481

DFSDRIS0, DBFDBLS0, DBFDRIS0

Explanation
This abend indicates that a problem prevented IMS from building a recoverable in-doubt structure (RIS). An RIS identifies and saves information about data in the in-doubt state. The system tries to build an RIS whenever a thread abends while owning in-doubt data.

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.

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.

ABENDU0484

DFSDRID0, DBFDLOG0

Explanation

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.

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

ABENDU0499

DFSCPY00, DBFHSRT0

Explanation

The maximum insert-call count was included and the application received an X'A7' status code. The application terminated abnormally when it attempted to issue another insert.

Analysis

This 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 references to “status code” in.

ABENDU0500

DFSASV20

Explanation

IMS was unable to find the DEB pointed to by the 7770-3 DCB by following the TCB-DEB chain.

Analysis

ABENDU0500 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.

Chapter 3. IMS Failure Analysis Structure Tables (FAST) 501 - 1000

The following topics provide additional information about abends 501 through 1,000.

ABENDU0501

DFSDN130, DFSDN140, DFSDS060

Explanation
The device-dependent module for the 3270 remote or the 3275 dial terminal detected an undetermined problem.

Analysis
ABENDU0501 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 U0501 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.

DFSDN130

Analysis
ABENDU0501 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 DDM 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.

DFSDN140

Analysis
ABENDU0501 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 DDM 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.
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.

DFSDS060

Analysis

ABENDU0501 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.

ABENDU0502

DFSDN070

Explanation
The IMS graphics attention handler entry point (DFSILAH0) 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.

ABENDU0503

DFSCPIN0

Explanation
The CTB you are trying to stop is not on the same line as you are currently dispatched on.

Analysis
ABENDU0503 is a standard abend issued by the communications module, DFSCPIN0, 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.

ABENDU0504

DFSICLA0

Explanation
The IMS switched line connect/disconnect processor module has detected an error in a get unique (GU) or a dequeue (DEQ) call to the queue manager while attempting to retrieve or delete messages belonging to an inquiry logical terminal on a switched line. This is an IMS system error.

Analysis
ABENDU0504 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.

ABENDU0505

DFSASV20

Explanation

The 7770-3 DEB builder, module DFSASV20 (z/OS only) was unsuccessful in an attempt to obtain storage.

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.

ABENDU0506

DFSBACK0

Explanation

A logic error was detected by DFSBACK0. There are two possible reasons for this error:

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 equals 0:

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.

For an input log from an IMS DB/DC or DBCTL subsystem, this abend should not occur.

If register 6 does not equal 0:

For dataspace management problems, register 6 will be the address in DFSBACK0 where the error was detected; register 15 will contain the return code from DFSRVSP0, the dataspace management module. DFSRVSP0 return codes 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

ABENDU0507

DFSBACK0

Explanation

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, DFS3278I, DFS3278A, or DFS3289A is issued.

Analysis

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 +GOSUB2	LOGRSRCH	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.

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 Causes for Message DFS894I

- Improperly closed log data set
- Corrupted log input (bad tape/drive)
- I/O read error

- Bad log records

Analysis

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 for Message DFS896A

- Incorrect or incomplete log input

Analysis

Message DFS3278I applies to the following table:

Message	Subcode	Key	Label	Description
DFS3278I	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.

Possible Causes for Message DFS3278I

- Improperly closed log data set
- Corrupted log input
- Invalid log input

Analysis

Message DFS3278A applies to the following table:

Message	Subcode	Key	Label	Description
DFS3278A	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 for Message DFS3278A

JCL error on IMSLOGR input

Analysis
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 Causes for Message DFS3289A
- Improperly closed log data set
- Corrupted log input
- Invalid log input
- JCL error on IMSLOGR input

ABENDU0509

DFSCLMO0

Explanation
The message generator module (DFSCLMO0) found an invalid SYSID while building a response message for an IMS command.

Analysis
ABENDU0509 is a standard abend issued by the communications message generator overlay segment processor, DFSCLMO0. 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'.

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.

ABENDU0511

DFSCONM0, DFSCON00, DFSCON10, DFSCON20

Explanation
A conversational processing error has occurred. Either an invalid buffer request has been specified, or a request to free a conversation was made when no conversation was active. This is probably an IMS system error.

Analysis
ABENDU0511 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 in FAST.

DFSCONM0

Analysis

ABENDU0511 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 ABENDU0511 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.
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.

DFSCON00

Explanation

ABENDU0511 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 ¬= 0	LOGERR	Nonzero return code trying to log the start of a conversation.
Reg15 ¬= 0	ISRTFLG1	Nonzero queue manager return code trying to do an 'insert prefix' call.
Reg15 ¬= 0	ISRTFLG2	Nonzero queue manager return code trying to do a 'put locate' call.
Reg15 ¬= 0	FNDBREAD	Nonzero queue manager return code trying to do a reposition call.
Reg15 ¬= 0	ISRTISRT	Nonzero queue manager return code trying to insert move message prefix call of APPC prefix.

DFSCON10

Explanation

ABENDU0511 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 *series* 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	

DFSCON20

Analysis

ABENDU0511 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.

Key	Label	Description
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.
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.

ABENDU0513

Several Modules

Explanation

A suspected overlay of one of the Data Communication (DC) buffer pools (I/OP, CWAP, HIOP, or LUMP), or of the Automated Operations buffer pool (AOIP), occurred. If the overlay is a DC pool type, the error might be caused by one of the user's DC exits or edit routines, an access method (such as BTAM or 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 IMS code.

Analysis

ABENDU0513 is a standard abend issued by modules DBFCMP10, DFSCFEI0, DFSCMCP0, DFSCMS00, DFSCOFC0, DFSCON20, DFSCRTU0, DFSDMIF0, DFSDMIQ0, DFSICIO0, DFSIIDM0, DFSLIEE0,

DFSNCS10, DFSQUEI0, and DFSSIST0 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 ABENDU0513 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 follows for each module:

DBFCMP10

Explanation

A potential storage overlay of the Fast Path Segment Compression work area (SEG1 or SEG2) occurred.

Analysis

ABENDU0513 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)

DFSCFEI0

Analysis

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.

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.

DFSCMCP0

Analysis

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).

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).

DFSCMEI0

Analysis

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.

Register 4 contains the address of the invalid buffer.

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.

DFSCMS00

Analysis

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 (DDM) 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.
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.

DFSCOFC0

Analysis

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 (DDM) at entry 6 (DD6M or DD6S).
Reg15='8C'	CALLDD1	A buffer overlay was detected before calling the device dependent module (DDM) at entry 1 (DDM1).

DFSCON20

Analysis

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
```

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.

DFSCRTU0

Analysis

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.

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.

DFSDMIF0, DFSDMIQ0

Analysis

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.

DFSICIO0

Analysis

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.

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 (DDM) 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.

Key	Label	Description
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).
Reg15=X'3C' Reg14=address of the invalid buffer	RECEIVE	A buffer overlay was detected while returning a RECANY 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).

Key	Label	Description
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.

DFSIIDM0

Analysis

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.

DFSLIEE0

Analysis

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.

DFSNCS10

Analysis

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.

DFSQUEI0

Analysis

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.
Reg15=X'64'	UEXIT	A buffer overlay was detected after calling AO exit routine.

DFSSIST0

Analysis

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.

ABENDU0516

DFSCON00, DFSICLH0

Explanation

More than one input and one output logical terminal are in conversation on the same physical terminal.

Analysis

ABENDU0516 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.

DFSCON00

Use register 5 in the abend SVRB to isolate to the specific label.

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.

DFSICLH0

Use register 4 in the abend SVRB to isolate to the specific label.

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.

ABENDU0517

DFSCONM0, DFSCON20, DFSECP10

Explanation

The conversational scratch pad 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, will either abend the control region or issue a pseudoabend to abend the message region.

Analysis

ABENDU0517 is a standard abend issued by three different modules: DFSCON20, DFSCONM0, DFSECP10 (set by DFSCPY00 and others).

The packed SPA will have to be located and examined for its validity. The format of a packed SPA is:
PPZZBBTRANCODEUUPP...Packed Data...

The following defines the value of PPZZBBTRANCODEUUPP:

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.

DFSCONM0

Analysis

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.

DFSCON20

Analysis

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, ABENDU0517 is issued.

The input message (containing the SPA segment) will have to be located 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 DFS2146I and the MSC link aborts 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. CTBPREFX 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). Refer to message DFS2146I in *IMS Version 9: Messages and Codes, Volume 2* for additional information.

DFSECP10

Analysis

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 user ABENDU0517 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.

ABENDU0519

DFSCMT10, DFSCLMR0, DFSICLR0, DFSCLMO0

Explanation

The message router, DFSICLR0, was called with an invalid enqueue request.

Analysis

ABENDU0519 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 will contain an indicative reason code. Register 12 is the base register. Register 5 contains the invalid queue number that was computed.

Code ***Meaning***

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.

ABENDU0525

DFSTIME0

Explanation

This abend is issued for either of the following reasons:

- The hardware clock is inoperable.
- The UTC offset value is outside the valid range.

Analysis

ABENDU0525 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 Causes

- A hardware failure occurred.
- An internal program error occurred.

ABENDU0545

DFSRELP0

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. The IMS alternate system abends. You must restart the IMS alternate system.

ABENDU0551

DFSPCC20

Explanation

GSAM PCBs were present, but the dependent region was not a batch message processing region (BMP).

Analysis

ABENDU0551 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.

ABENDU0552

DFSCON10, DFSCON20

Explanation
A conversational MPP terminated abnormally while in multiple systems. The SPA cannot be sent to the input terminal system, where the conversation is controlled, because no MSNAME exists for the input SYSID, which is used for responses.

Analysis
ABENDU0552 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.

DFSCON10

Analysis
ABENDU0552 is a standard abend that can be issued by the conversational processor inserting a SPA, module DFSCON10. The abend is issued from a common routine at label ABND.

ABENDU0522 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 DFSCONI0 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.

DFSCON20

Analysis

ABENDU0552 is a standard abend that 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 U0552 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.

ABENDU0553

DFSCMM20

Explanation

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).

Analysis

ABENDU0553 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.

ABENDU0554

DFSCMM20

Explanation
The main storage-to-main storage post handler received an illogical post code of read attention.

Analysis
ABENDU0554 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' Reg10=A (ECB(LXB)	CMM20B01	Read attention is posted in the ECB(LXB). This is an invalid post.

Possible Cause
Probable logic error in the main storage-to-main storage access method (DFSMTMA0).

APAR Processing
Dump from both systems with the access method trace invoked.

ABENDU0555

DFSCMM20

Explanation
The main storage-to-main storage post handler (DFSCMM20) received an invalid ECB(LXB) posting, and the CLB(LLB) is currently posted or dispatched.

Analysis
ABENDU0555 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' Reg9=A(CLB(LLB)) Reg10=A(ECB(LXB))	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.

Possible Cause
IMS logic error.

APAR Processing
Dump from both systems with the access method trace invoked.

ABENDU0556

DFSCMM20

Explanation
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
ABENDU0556 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.

APAR Processing
Dump from both systems with the access method trace invoked.

ABENDU0557

DFSCMR00

Explanation
During a restart of a Multiple Systems Coupling (MSC) system, the LCB/LLB resolution module (DFSCMR00) failed to resolve the LCB/LLB relationship.

Analysis
ABENDU0557 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.

ABENDU0560

DFSCLMR0

Explanation
An attempt to acquire a buffer or cancel a message resulted in an invalid return code from the queue manager (DFSQMGR0).

Analysis
ABENDU0560 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:

Code ***Meaning***

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.

ABENDU0561

DFSCMM20

Explanation
The main storage-to-main storage post handler (DFSCMM20) was called to service a post request, but the target CLB(LLB) link number was zero.

Analysis
ABENDU0561 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.

APAR Processing
Output of system definition and linkage editor listing.

ABENDU0562

DFSCMM20

Explanation
In a main storage-to-main storage link the data received exceeds the input buffer length.

Analysis
ABENDU0562 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.

APAR Processing
Dumps of both systems.

ABENDU0563

DFSCMR00

Explanation
During the reestablishment of the relationship between the LLB and the CLB at restart, an invalid condition has been detected. The LCB shows that it was assigned to an LLB, but the link number in the LCB is zero.

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
User error at system definition time.

ABENDU0564

DFSICIO0

Explanation
A device-dependent module (DDM) has called the communication analyzer at entry DFSCIOC0 to obtain an extra input/output work buffer, but an extra buffer already exists. This situation occurs because of a system error.

Analysis
ABENDU0564 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 ABENDU0564, 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.

ABENDU0566

DFSCMI00

Explanation
An internal logic error was detected while processing an internal and remote command. A message queue buffer may have been destroyed.

Analysis
ABENDU0566 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 refetched, 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.

ABENDU0567

DFSCINB0

Explanation
The communications analyzer detected an I/O error condition in sending the IMS ready message to the system console.

Analysis
ABENDU0567 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 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 DFSICI00 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.

ABENDU0568

DFSCINB0

Explanation
The communication initialization module received a nonzero return code from the queue manager while trying to acquire a buffer to transmit the IMS ready message.

Analysis
ABENDU0568 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:

Code ***Meaning***

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.

ABENDU0572

DFSCLMR0

Explanation
The length of an inner segment of a multisegment preedit message is a negative value.

Analysis
ABENDU0572 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.

ABENDU0573

DFSCLMR0

Explanation
The length of an inner segment of a multisegment message is larger than the total length of the message.

Analysis
ABENDU0573 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.

ABENDU0577

DFSICIO0

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 ABENDU0577.

ABENDU0578

DFSIINB0

Explanation

This abend occurred because the CTT device type (CTTDEVIC) was either in binary zeros or the value was less than X'33'(system console). CTTDEVIC is picked up and examined for each defined terminal during data communications initialization (control TCB). IMS terminal device types begin with number one.

Both of these conditions are obvious errors and probably occur because of one of the following conditions:

- The stage 2 assembly of DFSCLL0x had an error.
- The linkedit 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 linkedit 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).

ABENDU0579

DFSIINB0

Explanation
The control region initialization was unable to open successfully any line groups. This is an internal IMS error. A nonswitched device-type code (CTTDEVIC) was greater than X'37', which is invalid.

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.

ABENDU0580

DFSIINB0, DFSDINB0

DFSIINB0

Explanation
An IMODULE GETMAIN failed to obtain CSA storage for the CSACLB (ECB) for DFSCMTI0; this is an IMS internal error.

Analysis
This 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:

Code	*Meaning*
X'04'	Invalid function or option.
X'08'	Execution of z/OS not consistent with assembler z/OS for SVC routine.
X'14'	Insufficient storage in subpool, or requested length is zero.
X'18'	Program check or other internal error.

DFSDINB0

Explanation
An IMODULE GETMAIN failed during DB Control (DBCTL) initialization.

Analysis
Message DFS697I is issued along with this abend. The EPLOC field in the message will contain 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.

ABENDU0581

DFSIIMS0, DFSIINB0, DFSIINU0, DFSIINV0

Explanation
The destination-find module (DFSICLF0) was unable to locate a CNT/LNB/RCNT, defined for a VTAM/BTAM/MSC terminal.

Analysis
This is a standard abend issued by modules DFSIIMS0, DFSIINB0, DFSIINU0, and DFSIINV0.

DFSIINB0

Explanation
The destination-find module (DFSICLF0) was unable to locate a system control block DFSMTCNT-CNT or DFSRMCNT-CNT; this is an internal IMS error.

Analysis
This is a standard abend issued by module DFSIINB0. Register 2 contains a pointer to the destination to be located.

DFSIINV0

Explanation
The destination find module (DFSICLF0) was unable to locate a system control block (CNT) related to a set of VTAM terminal blocks (VTCB).

Analysis
This is a standard abend issued by module DFSIINV0. 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.

ABENDU0582

DFSIINB0

Explanation
The DFSBCB operation failed to obtain storage for a VTAM Receive Any buffer. This is an internal IMS error.

Analysis
ABENDU0582 is a standard abend issued by module DFSIINB0. For a description of the DFSBCB return codes, see the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

ABENDU0583

DFSIINB0

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 user link-edited IMS/VTAM control blocks with a non-IMS/VTAM nucleus.

ABENDU0584

DFSCLMA0

Explanation
An error situation was detected by the automated operator interface message generator.

Analysis
This is a standard abend issued by the AOI message generator module, DFSCLMA0. Register 5 points to the code in the module where the error was detected. Register 2 contains an error code indicating the reason for the abend, as follows:

Code	Meaning
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.

ABENDU0585

DFSIINB0

Explanation
An error occurred during XRF initialization.

Analysis
The subcode in register 15 indicates the specific problem.

Key	Label	Description
Reg15=X'01'	HOTI01	The HSBID parameter value is invalid.

Key	Label	Description
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

- 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 the IBM support center for assistance.

ABENDU0586

DFSHINT0

Explanation

An error was detected during XRF initialization.

Analysis

ABENDU0586 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

- An invalid parameter was specified.
- An invalid HSBMBR parameter was specified.

ABENDU0587

DBFERST0, DBFERDB0

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' or X'24' 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:

Code	Meaning
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 *IMS Version 9: Messages and Codes, Volume 2*.

X'1C' Unable to get storage for a buffer during processing of a 5947 log record.

X'24' Unable to get storage needed to process a 5950 private buffer log record.

ABENDU0589

DFSHREQ0, DFSHCI00

DFSHREQ0

Explanation

A FIND BLOCK request to CBT Services failed.

Analysis

ABENDU0589 is a standard abend issued from module DFSHREQ0 by an XRF alternate system during takeover. DFSHREQ0 creates I/O Toleration 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 XRF active system generated an incorrect log record.
- The XRF alternate system incorrectly processed a log record.

DFSHCI00

Explanation

A GET or RELEASE control block request (DFSBCB FUNC=GET/REL) for an asynchronous work element (AWE) from the AWE pool failed.

Analysis

Register 14 points to label HCIGAWE for a GET AWE, or to label HCIRAWE 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.

ABENDU0590

DFSHTRM0

Explanation
Either IRLM or VTAM terminated abnormally on the central processor complex (CPC) where the alternate system resides.

Analysis
ABENDU0590 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 ABENDU0590. No dump is taken.

ABENDU0591

DBFERS20

Explanation
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 is an internal IMS error.

Analysis
ABENDU0591 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 *IMS Version 9: Messages and Codes, Volume 1* 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.

ABENDU0592

DBFEACL0, DBFHCIR0, DBFENIS0, DBFE2CI0, DBFERS10

Explanation
A Fast Path XRF module was unable to obtain an Area lock while preparing to refresh the DMAC from the 2nd 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. IRLM codes are explained in *IMS Version 9: Messages and Codes, Volume 1*.

ABENDU0593

DFSRDBP0, DFSRESP0, DFSRST00

DFSRDBP0

Explanation
A database block failure occurred during restart processing.

Analysis
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:

Code **Meaning**

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.

Key	Label	Description
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(log record) Reg14=A(return SCHD 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: **X'01'** PSB is stopped or locked. **X'02'** Database stopped or locked. **X'03'** I/O error in reading PSB/DMB.DMB not found, or prior DMB error (DDIRBAD set). **X'04'** Intent conflict. **X'05'** PSBW/DMB pool too small to hold PSB/DMB. **X'06'** EPCB/DMB/PSB/PSBW pool storage is temporarily not available or DFSDBAU could not get needed space. Refer to PSTCODE1 for more information. **X'07'** Invalid PCB PROCOPT L or LS. **X'08'** FP buffer page fix error. **X'09'** PCB processing intent not compatible with database access. **X'0A'** Database authorization failure. **X'0B'** Database not available. **X'0C'** 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 SCHD 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: **X'40'** Intent conflict **X'20'** PST on DMB wait queue **X'08'** Type 3 batch region **X'04'** PST on PSB wait queue **X'x2'** 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. **X'00'** 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.

DFSRESP0

Explanation

A database or restart block failure occurred during restart.

Analysis

ABENDU0593 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.

Code	Meaning
X'08'	Block 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=FIND or GET, BLK=EQEL, DDIR, EQEL, or RRE request.

DFSRST00

Explanation
A database block failure occurred during restart processing.

Analysis
ABENDU0593 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:

Code	*Meaning*
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.

ABENDU0594

Several Modules

Explanation
A request to the RPST Block Management Services failed.

Analysis
ABENDU0594 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 ABENDU0594. 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
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 XRF active system generated an incorrect log record.
- The XRF alternate system incorrectly processed a log record.

ABENDU0595

DFSHDAI0, DFSHPTK0

Explanation
An OS GETMAIN request for storage failed.

Analysis
ABENDU0595 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.

ABENDU0596

DFSHPTK0

Explanation
A Buffer Tracking failure occurred during the tracking or takeover phase in an XRF complex.

Analysis
ABENDU0596 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
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 XRF active system generated an incorrect log record.
- The XRF alternate system incorrectly processed a log record.

ABENDU0597

DFSHLTK0, DFSHRAL0

Explanation
A Lock Tracking or Lock Reacquire failure occurred.

Analysis

ABENDU0597 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.

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.

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.
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 XRF active system generated an incorrect log record.
- The XRF alternate system incorrectly processed a log record.

ABENDU0598

DFSRST00, DBFNRST0

DFSRST00

Explanation

The DSNAMEs and VOLIDs of the system data sets in the active and alternate systems are inconsistent for XRF synchronization or for restart.

Analysis

ABENDU0598 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:

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.

DBFNRST0

Explanation

An MSDB data set name or VOLID is inconsistent for restart or XRF synchronization.

Analysis

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 prior to 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 DBFMSDSN.

Key	Label	Description
Reg3=C'MSDB'	NRST4080	One of the following occurred: 1. During normal restart, emergency restart, or XRF backup synchronization, a logged MSDB DBD name was not found in the system. 2. During XRF synchronization, a discrepancy was found between an MSDB VOLID on the active and alternate systems.

Possible Cause

- IMS was started with an incorrect procedure.
- The data set was migrated to a different volume.
- An internal program logic error occurred.

ABENDU0599

DFSRCP00, DFSRLP00

Explanation

An error occurred in a page fix or page free call.

Analysis

ABENDU0599 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. For an explanation of the IMSAUTH PGFIX/PGFREE return codes, see the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

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.

ABENDU0600

DFSRCP00, DFSRLP00

Explanation

A DIAGNOSE O instruction failed.

Analysis

ABENDU0600 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.

ABENDU0601

DBFDBFM0, DBFERS20, DBFTOPU0, DFSIDC00, DFSTODI0, DFSTOPR0

Explanation
An IMODULE GETMAIN or IMODULE DELETE failure occurred attempting to get or free storage.

Analysis
ABENDU0601 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 will point to the instruction from which the abend (SVC 13) is issued.

Register 15 contains a return code. For an explanation of the IMODULE return codes, see the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

ABENDU0602

DBFTOFNO, DBFTOPUO, DFSTOBHO, DFSTODIO, DFSTOPRO, DFSTORSO

Explanation
An IMS logic error occurred.

DBFTOFNO, DBFTOPUO

Analysis
ABENDU0602 is a standard abend that can by issued by DBFTOPUO. 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 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

Key	Label	Description
Reg15=X'66'	ABNLKREL	Lock release failed (RO = IRLM rc)
Reg15=X'67'	ABNOEPST	Get EPST failed

DFSTOBHO, DFSTORSO

Analysis

ABENDU0602 is a standard abend that can be issued by DFSTORSO. 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. Message DSF0612I 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

DFSTODI0

Analysis

ABENDU0602 is a standard abend that can be issued by DFSTODIO. 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. 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

DFSTOPRO

Analysis

ABENDU0602 is a standard abend that can be issued by DFSTOPRO. 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. Message DFSO612I 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

Key	Label	Description
Reg15=X'15'	ABND602	Invalid EEQE found
Reg15=X'1F'	ABNOEPST	EEQE COPY found an invalid EEQE

ABENDU0603

DBFHLOD0, DFSRLP00

Explanation

An error was detected while loading the main storage databases (MSDBs) from the system log.

Analysis

ABENDU0603 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' ESCDDBFA=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'=ESCDDBFL Reg15=X'04'	HLOD4079	The total length of page-fixed MSDB data does not match the length in ESCDDBFL.
Reg15=X'05' ESCDDBPA=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'=ESCDDBPL Reg15=X'06'	HLOD4079	The total length of pageable MSDB data does not match the length in ESCDDBPL.
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.

Key	Label	Description
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

- An internal program logic error occurred.
- The active and the alternate system are not the same system.

ABENDU0604

DFSICA20, DFSTRM00

Explanation

This abend is the expected response to the /SWITCH SYSTEM command, with either the keywords FORCE, ACTIVE, or both. Thus any of the following forms of the command will cause this abend: `/SWITCH SYSTEM FORCE` or `/SWITCH SYSTEM ACTIVE` or `/SWITCH SYSTEM FORCE ACTIVE`.

These commands force an ABENDU0604 in the active system, causing the alternate system to take over.

ABENDU0605

DFSBML00, DFSDBLR0

DFSBML00

Explanation

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.

DFSDBLR0

Explanation

An unexpected condition occurred in the attempted release of the ACBLIB read 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.

DFSBML00, DFSDBLR0

Analysis

ABENDU0605 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.

ABENDU0606

DFSSUSX0

Explanation

An unexpected condition occurred in the IMS IRLM suspend exit routine. This may be an IMS system error, or an unauthorized application program may be trying to use the IMS CROSS-MEMORY option.

Analysis

ABENDU0606 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.

ABENDU0608

DFSCFE80, DFSCFE90, DFSCFEI0, DFSCFEO0

Explanation: DFSCFE80, DFSCFE90, DFSCFEI0

While the /TRACE SET ON TRAP 2 command was enabled, IMS detected an overwrite of the MFS blocks.

Explanation: DFSCFEO0

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

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

Code	*Meaning*
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)

Code	*Meaning*
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)

Code	*Meaning*
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.

ABENDU0611

DFSCFE80, DFSCFE90

Explanation

During processing of a device output format (DOF) field descriptor element (FDE), Message Format Service (MFS) detected that an invalid request was made.

Analysis

ABENDU0611 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.

Key	Label	Description
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.
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 may have been modified.

ABENDU0616

DFSFDLB0

Explanation
As a result of a permanent I/O error, there was no batch log data set (SLDS).

Analysis
ABENDU0616 is a standard abend that can be issued by the physical logger - buffer post process routine, DFSFDLB0. 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, IEFRDER (IEFRDER2 if dual), has an I/O error. The buffer prefix, LBUFFER, contains the BSAM DECB and LDSET contains the status flags.

DFSFDLS0

Explanation
For batch, it means 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.

Analysis
ABENDU0616 is a standard abend that 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).
Reg8=A(LDSET)	CLOSEBAD	A CLOSE error was encountered on the batch log data set.

ABENDU0622

DFSRRA40, DFSRRA70

Explanation
In z/OS, IMS attempted to initialize a noncontrol region with a protect key in the range of 0 to 7.

Analysis
ABENDU0622 is a standard abend issued by DFSRRA40 (online) or DFSRRA70 (batch). Message DFS0622I is written to the IMS master console prior to 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 will abend with a dump.

Key	Label	Description
	RAMPCOM RADSVC1	The TCB protect key field (TCBPKF) is tested for a key in the range 0 to 7. A noncontrol region must execute 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's LPALIB.

ABENDU0623

DFSRRA00

Explanation
In z/OS, IMS attempted to initialize a control region with a protect key in the range 8 to 15.

Analysis
ABENDU0623 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 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

1. 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.
2. Some data sets in JOBLIB are not authorized.

ABENDU0624

Various Modules

Explanation

IMS received a nonzero return code from DFSBCB when trying to obtain an AWE.

Analysis

ABENDU0624 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.

ABENDU0630

DFSXCB00

Explanation

This abend is issued 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.
6. User exit routine initialization call to DFSUSRXI failed.

Analysis

ABENDU0630 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. The possible contents of register 15 are first summarized and then described in more detail.

'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.
'G' 00 00 nn	IMODULE GETMAIN failed for DFSCBTHD. The IMODULE return code is nn.
'I' nnnnnn	IMODULE GETMAIN failed for DFSXCB01. The IMODULE return code is nnnnnn.
'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.

'LUS' nn IMODULE LOAD failed for DFSUSRXI. The IMODULE return code is nn.

'L60' nn IMODULE LOAD failed for DFSBCB60. The IMODULE return code is nn.

'Q' nnnnnn DFSBCB GET failed for QSAV. The DFSBCB return code is nnnnnn.

'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 nnnn.

'SRX'nn Initialization call to DFSUSRXI failed. The return code from DFSUSRXI is nn.

'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 for 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 For an explanation of the IMODULE macro return codes, see the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

2. QSAV, BXQE, AWE pool failure. A probable link-edit error. DFSCBT10 does not include one of the above control block definitions. DFSCBT10 assembly had an error but the output was still link-edited 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 link-edit error. DFSCBT10 does not include one of the above control block definitions. DFSCBT10 assembly had an error but the output was still link-edited 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 desired 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 needs to 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. For an explanation of the IMODULE macro return codes, see the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

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. For an explanation of the IMODULE macro return codes, see the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

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. For an explanation of the IMODULE macro return codes, see the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

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 needs to

be increased by at least 8K bytes. Refer to the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1* for additional return code information.

ABENDU0631

DFSIINB0, DFSIINV0

Explanation

During DC initialization, a call was made to the IMS dispatcher using the DFSCIR macro to initialize an ECB. The DFSCIR macro returned with a nonzero return code in register 15 indicating that the request could not be processed.

Code	*Meaning*
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.

Analysis

ABENDU631 is a standard abend issued by DFSIINB0 or DFSIINV0. The following registers are useful in problem diagnosis:

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.

ABENDU0632

DFSRRA20

Explanation

The EXEC statement PARM field contains too many positional parameters.

Analysis

| ABENDU0632 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. For more information refer to *IMS Version 9: Installation Volume 2: System Definition and Tailoring*.

ABENDU0633

DFSRRA70

Explanation

Parameter DBRC=C was specified, but the program specified was not Batch Backout (DFSBBO00).

Analysis
ABENDU0633 is a standard abend issued by DFSRRA00. Message DFS0633I is sent to the z/OS console prior to this abend.

Key	Label	Description
	RAIR	DBRC=C and RCPGM do not equal Batch Backout (DFSBBO00).

ABENDU0634

DFSRRA20

Explanation
A positional parameter in the EXEC statement's PARM field has one or more leading blanks.

Analysis
ABENDU0634 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. For more information refer to *IMS Version 9: Installation Volume 2: System Definition and Tailoring*.

ABENDU0636

DFSRRA30, DFSRRA40, DFSRRA70

Explanation
The last fixed-length parameter value in the PARM field of the EXEC statement is too short.

Analysis
ABENDU0636 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.

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. Refer to *IMS Version 9: Installation Volume 2: System Definition and Tailoring* for more information.

APAR Processing
Region dump and input JCL.

ABENDU0638

DFSRRA30, DFSRRA40, DFSRRA70

Explanation
A comma was embedded in a fixed-length parameter.

Analysis
ABENDU0638 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 *IMS Version 9: Installation Volume 2: System Definition and Tailoring* for the correct format.

APAR Processing
Region dump, and input JCL.

ABENDU0640

DFSRRA30, DFSRRA40, DFSRRA70

Explanation
A required parameter was omitted for the kind of execution specified by the first three characters of the EXEC statement PARM field.

ABENDU0640 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 *IMS Version 9: Installation Volume 2: System Definition and Tailoring* for required parameters for the different regions specified as the first three characters.

APAR Processing
Region dump and input JCL.

ABENDU0641

DFSRRA80

Explanation
The region parameter in the PARM field of the EXEC statement is incorrect.

Analysis
ABENDU0641 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.

ABENDU0642

DFSRRA30, DFSRRA40, DFSRRA70

Explanation
A parameter value exceeds the maximum allowable length.

Analysis
ABENDU0642 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.

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.

APAR Processing
Region dump and input JCL.

ABENDU0643

DFSRRA30, DFSRRA40, DFSRRA70

Explanation
A nonnumeric value was specified for a numeric parameter.

Analysis
ABENDU0643 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.

APAR Processing
Region dump and input JCL.

ABENDU0644

DFSRRA30, DFSRRA40, DFSRRA70

Explanation
The internal destination list is invalid.

Analysis
ABENDU0644 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.

APAR Processing
Abend dump with the program check trap installed in DFSRRA20.

ABENDU0646

DFSRRA00

Explanation
The SPIE option in the PARM field of the EXEC statement is invalid.

Analysis
ABENDU0646 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:

- stand-alone batch
- message processing
- batch message processing

Refer to *IMS Version 9: Installation Volume 2: System Definition and Tailoring* for an explanation of the correct SPIE option settings, in the information about procedures.

Key	Label	Description
	RADLNMVM	This routine is entered for a stand-alone batch region. A BAL is taken to label RAOPTR 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.

ABENDU0648

DFSRRA00

Explanation
The validity check option in the PARM field of the EXEC statement is invalid.

Analysis
ABENDU0648 is a standard abend issued by DFSRRA00. Message DFS648I is issued to the IMS master console prior to the abend. The region from which the abend was issued will terminate with a 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 a 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.

ABENDU0650

DFSPCC20, DFSRRA00

DFSPCC20

Explanation
| An invalid class number has been specified in the PARM field of the EXEC statement. The class number
| specified must not exceed the number of classes defined during system definition.

Analysis
ABENDU0650 is a pseudoabend issued by DFSPCC20, which was set up by DFSSMSC0.

The EXEC statement that starts a message processing region must contain a positional parameter specifying four 3-digit decimal numbers indicating which classes of messages this message region will handle.

DFSSMSC0 validates that the class number does not exceed the number of unique transaction code classes specified at generation time. The maximum number of transaction code classes is defined by the MAXCLASS operand of IMSCTRL macro. The definition results in the generation of a like number of Transaction Class Tables (TCTs).

If an error is detected, DFSSMSC0 places pseudoabend ABENDU0650 into the PSTABTRM parameter and returns. DFSPCC20 will check if an error is detected during create thread and will issue the abend.

Key	Label	Description
	PSABEND	The class assignment is compared to the maximum transaction code class value. This is the number of TCTs in the system as defined by the MAXCLASS operand.

Possible Cause
A control block generation has been performed that altered the maximum number of TCTs in the system.

DFSRRA00

Explanation
An invalid class number has been specified in the PARM field of the EXEC statement. The class number specified must not exceed the number of classes defined during system definition.

Analysis
ABENDU0650 is a standard abend issued by DFSRRA00. Message DFS650I is issued to the IMS master console.

The EXEC statement that starts a message processing region must contain a positional parameter specifying four 3-digit decimal numbers indicating which class of messages this message region will handle.

DFSRRA00 validates that the class number does not exceed the maximum allowed which is 255.

If an error is detected, a branch is taken to label ABND where register 1 is loaded with the message/abend code offset. A branch is then taken to label RAMSGAB to output the message and terminate with a dump at label RAABND.

Key	Label	Description
	CON	The class number is compared against the maximum allowed, X'FF' (255). A value greater than X'FF' results in abend.

Possible Cause
JCL error in coding the PARM field of the EXECUTE statement.

ABENDU0652

DFSRRA00

Explanation
The required PARM field of the EXEC statement has been omitted.

Analysis
ABENDU0652 is a standard abend issued by the EXEC statement PARM analysis program, DFSRRA00. Message DFS652I is issued to the IMS master console prior to 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 9: Installation Volume 2: System Definition and Tailoring*or *IMS Version 9: Utilities Reference: System* 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
User JCL error.

ABENDU0654

DFSRRA00

Explanation
The PARM field specified in the EXEC statement was fewer than three characters in length.

For control region types, the first two parameters in the PARM field of the EXEC statement are required, and they must each 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, and it must be exactly 3 characters in length. If this is not true, this abend occurs.

See IMS SDRM for descriptions of parameters used in the PARM field of the EXEC statement for all region types.

Analysis
ABENDU0654 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. Refer to *IMS Version 9: Installation Volume 2: System Definition and Tailoring* for examples of the PARM field format.

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
User JCL specification error.

ABENDU0657

DFSZDI00, DFSZDI20

Explanation
This 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.

Code	***Meaning***
C4C900xx	DFSZDI00 detected the error.
C4C920xx	DFSZDI20 detected the error.

Key	Label	Description
R2=C4C90001	C4C90001	Error opening IMS DD statement. Explanation: Opening of an IMS DD statement for PSBLIB and DBDLIB failed. Operator Response: Ensure that the step contains an IMS DD statement for PSBLIB and DBDLIB and that it names the proper libraries.

Key	Label	Description
R2=C4C90002	C4C90002	Inconsistent PSBs. Explanation: The PSB in ACBLIB indicates that the PSB contains a GSAM PCB, but the PSB in PSBLIB does not contain any GSAM PCBs. Operator Response: Correct the appropriate PSB and rerun ACBGEN or PSBGEN.
R2=C4C90003	C4C90003	Additional PCB denied. Explanation: An error was encountered in a BMP region. The GSAM PCBs were being added to the PCB list. This is an internal IMS error. Operator Response: Rerun the job. If the situation persists, contact the IMS system programmer.
R2=C4C92001	C4C92001	Invalid DBD (not a GSAM DBD). Explanation: 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. Operator Response: Correct the appropriate PSB or DBD and rerun PSBGEN, DBDGEN, or ACBGEN.
R2=C4C92002	C4C92002	R15 contains a nonzero return code from the RDJFCB macro. Explanation: 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. Operator Response: Provide the appropriate DD statement for the GSAM database, if required, and rerun the job.
R2=C4C92003	C4C92003	On return from the RDJFCB macro, the ARLAREA (address of the ARA area obtained by RDJFCB) is zero. In this case, ARLCODE (described below) is probably = 08. Operator Response: See appropriate ARLCODE (shown below in C4C9004).
R2=C4C92004	C4C92004	On return from the RDJFCB macro, ARLCODE at REG5 + X'1C' is nonzero. Explanation: **If ARLCODE = 4** The ARL was not initialized properly prior to issuing the RDJFCB macro. **If ARLCODE = 8** 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.) Operator Response: Increase maximum virtual storage available for the region and reexecute step.
R2=C4C92005	C4C92005	A GSAM or BSAM DD statement has greater than 255 data sets in the concatenation.

ABENDU0658

DFSRRA00

Explanation

The program-name parameter was omitted from the PARM field.

Analysis

This is a standard abend issued by module DFSRRA00.

Possible Cause
User error.

ABENDU0662

DFSRRA00

Explanation
The first positional parameter in the PARM field is invalid.

Analysis
ABENDU0662 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.

ABENDU0684

DFSRRA00, DFSXBAT0

Explanation
IMS type 2 SVC detected an error during SVC or batch initialization.

Analysis
ABENDU0684 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 prior to the abend of the control region.

ABENDU0684 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. Refer to *IMS Version 9: Messages and Codes, Volume 2* for an explanation of message DFS686W.

DFSRRA00

Key	Label	Description
	SVCABEND	SVC initialization failed for control or batch region.

DFSXBAT0

Key	Label	Description
Reg14	CLBCIC	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 of DFS627W is not received at IMS termination, then DFSMRCL0 is not gaining control to clean up IMS resources.

APAR Processing

Abend dump

ABENDU0688

DFSRRA00

Explanation

The operator replied CANCEL or 'C' in response to message DFS690A.

Analysis

ABENDU0688 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.

ABENDU0689

DFSRRA00

Explanation

The IMS control region was not active while trying to initialize the DBRC or DL/I subordinate address space region. The IMSID of the IMS issuing the START DBRC PROC command does not correspond with the IMSID in the DBRC procedure.

Analysis

ABENDU0689 is a standard abend issued by the EXEC statement PARM analysis program, DFSRRA00. Message DFS689I is issued to the IMS master console prior to 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 is caused by one of the following conditions:

1. The DBRC or DL/I subordinate address space region was started by the operator 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.

Key	Label	Description
	RAENQWT1	A branch is taken to label RAMSGAB to issue the message and abend.

ABENDU0701

DFSIIEN0*

*The IMS format dump diagnostic area shows the calling module ID as the module that abended, instead of DFSIIEN0.

Explanation

IMS attempted to enqueue using an invalid queue control or queue element.

Analysis

ABENDU0701 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.

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.

Key	Label	Description
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.

ABENDU0702

DFSIIDE0

Explanation
IMS attempted to dequeue using an invalid queue control block or queue element.

Analysis
ABENDU0702 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.

ABENDU0704

DFSISMN0

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 ABENDU0704, 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.

ABENDU0705

DFSISMN0

Explanation
An IDESTROY call was issued to free a buffer, but a buffer with the same name specified could not be found.

Analysis
For ABENDU0705, 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.

ABENDU0707

DFSISMN0

Explanation
ABENDU0707 is issued from DFSISMN0 because the user is trying to allocate a buffer from a non-existent pool.

Analysis
For a U0707 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
The user called DFSISMN0 to get a buffer from a pool that was never created.

ABENDU0708

DFSISMN0

Explanation
ABENDU0708 is issued from DFSISMN0 because the user tried to return a buffer to a nonexistent pool.

Analysis
For ABENDU0708, 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.

ABENDU0709

DFSISMN0

Explanation
ABENDU0709 was issued from DFSISMN0 because the user is requesting a variable-length buffer with a length greater than the pool size.

Analysis
For ABENDU0709, find label IGCHKREQ, then look for the instruction B IGVARERR. Label IGVARERR loads register 1 with the abend code and branches to ABNDEXIT where the abend is issued.

Key	Label	Description
Reg1=abend code	IGCHKREQ	The requested buffer size found in register 3 is greater than the total pool size. The pool name is in register 2. Register 6 points to the pool header.

Possible Cause
The caller has called DFSISMN0 to get a buffer from a variable-length pool with a length greater than maximum length.

ABENDU0710

DFSISMN0

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
ABENDU0710 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 ABENDU0710 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.

ABENDU0711

Several Modules

Explanation

A system error was detected while processing in an RRS environment.

DFS0613I may be received indicating an ABENDU0113 has occurred due to an ABENDU0711. This situation can occur when RRS terminates and an IMS dependent region ends with an ABEND058 while in Fast Path syncpoint processing. The ABEND058 is converted to an ABENDU0711 RC2D for diagnostic purposes. To verify that this is the 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 ABENDU0113 dump to find the DPST for that dependent region. Flag SAPX1AFP will be on and EPSTW1SY=1.
4. An IRC START entry in the IMS Scheduler trace will indicate an ABEND058 completion code for that dependent region.

DFSASK00, DFSCPY00, DBFHGU10, DFSPCC20, DFSECP10

Analysis

ABENDU0711 is 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 ABENDU0711. 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. ABENDU0711 ceases when registration is complete, which is indicated by message DFS06531.

The application terminates abnormally.

DFSRRC10, DFSRRSI0, DFSTMS00, DFSRRSB0, DFSAERG0

Analysis

ABENDU0711 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 reason codes listed below 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	Description
X'2D'	The dependent region was processing a call to ATRCMIT or ATRBACK when RRS terminated, resulting in an ABEND058. The ABEND058 is converted to an ABENDU0711 RC2D. Register 1 reflects the ABEND058 completion code.

DFSRRSB0:

Reason Code	Description

X'23' IMS failed to register as a resource manager with RRS. Register 2 contains the return code from the CRGGRM service.

X'24' IMS was unable to switch the private context to another batch TCB. Register 2 contains the return code from the CTXSWCH service.

X'26' IMS was unable to create a private context. Register 2 contains the return code from the CTXBEGC service.

X'2B' IMS was unable to load all the RRS callable stubs.

DFSRRSI0:

Reason Code	Description
X'00'	The caller requested an invalid DFSRRSI function.
X'01'	IMS failed to obtain storage for DFSRRSIB. Register 2 contains the return code from IMODULE GETMAIN.
X'03'	Retrieval of the IMS and RRS log names failed. Register 2 contains the return code from the ATRIRLN service.
X'04'	The length of the IMS log name returned by RRS is invalid. Register 2 contains the length of the log name.
X'05'	The IMS log name returned by RRS is invalid. Register 2 contains the address of the log name.
X'06'	IMS's 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 and it causes an ABENDU0113.
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.

X'1B' IMS's notification exit failed to set IMS's 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 IMS's 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 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.

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 may 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	Description
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 syncpoint 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 abort the sync point for the application. This return code can also occur if the front-end RRS is cancelled and restarted for synchronous APPC/OTMA transactions that are processed by the back-end IMS.

X'20' An application program, whose input is part of a protected conversation, issued a ROLB call.

X'21' Disassociation 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 abort call was received for the unit of work. This error can also occur if either a /STOP REGION *xx* ABDUMP command was issued for a region with at 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 is an IMS internal or RRS error. The pseudodump (67FF log records) will need to be analyzed.

DFSAERG0:

Reason Code	Description
X'33'	ODBA could not successfully validate the application call. Register 5 contains the return code from the CTXRDTA service or other diagnostic information. Register 14 contains the address where the error was detected.

ABENDU0712

DFSIMBE0, DFSIMBD0

Explanation

The IMS transaction enqueue or dequeue service encountered a system error. The reason code in register 15 indicates the error.

DFSIMBE0

Analysis

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.

Key	Label	Description
Reg15=X'14' Reg14=BAL Reg3=A(TCT)		The pointer (TCTPRIOR) to the highest priority transaction within a class was zero.

DFSIMBD0

Analysis

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.
Reg15=X'10' Reg14=BAL Reg10=Latch Manager return code Reg9=A(TCT)		A request to release the TCTB latch in exclusive mode failed.

ABENDU0713

DFSSMSC0, DFSIMBE0

Explanation

The IMS MPP scheduler or the IMS transaction enqueue service encountered a system error. The reason code in register 15 indicates the error.

DFSSMSC0

Analysis

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.

Key	Label	Description
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.
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.

DFSIMBE0

Analysis

This is a standard abend issued by module DFSIMBE0.

Key	Label	Description
Reg15=X'04' Reg14=BAL Reg10=A(Current 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).

ABENDU0714

DFSSBMP0, DFSTMAD0

Explanation

The IMS BMP region scheduler or the IMS APSB/DPSB call processor encountered a system error. The reason code in register 15 indicates the error.

DFSSBMP0

Analysis

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 RELPDRBL 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 RELPDRBL Reg9=A(PST) Reg7=A(PDIR) Reg2=Latch manager Return code		A request to release the APSB latch in shared mode failed.

DFSTMAD0

Analysis

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.

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.

ABENDU0716

DFSIINQ0

Explanation

During initialization of an IMS control region, a 'NO BUFFERS' condition was detected.

Analysis

ABENDU0716 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 will point 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 abend subcode.

Code ***Meaning***

X'04' IMODULE GETMAIN failed while attempting to allocate private storage for the queue buffer address list, DFSPQBFA. Register 3 contains the requested storage length and register 15 contains the IMODULE return code. For an explanation of IMODULE return codes, refer to the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*. 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 is an internal logic error. Refer to problem determination.

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. This is an internal logic error. Refer to problem determination.

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. This is an internal logic error. Refer to problem determination.

X'18' IMODULE GETMAIN failed while attempting to allocate private storage for the optional DCB DD statements. Register 15 contains the IMODULE return code. For an explanation of IMODULE return codes, refer to the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*. 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 done 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. This is an internal logic error. Refer to problem determination.

X'1C' DFSCBTS FUNC=ALTER failed while attempting to set the block size and number for the QMBA block. Register 15 contains the DFSCBTS return code. This is an internal logic error. Refer to problem determination.

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.

ABENDU0717

DFSIINS0, DFSDOBI0

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 DFSDOBI0.

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
ABENDU0717 is a standard abend issued from module DFSIINS0 and DFSDBOBI0.

When issued from DFSDBOBI0, ABENDU0717 is issued from subroutine GETERR. At the time of abend, register 10 contains the SCD address, register 2 contains the total size of the buffer pool, and register 15 contains IMODULE GETMAIN return code.

When issued from DFSIINS0, register 11 contains the SCD address and register 8 contains one of the following abend subcodes.

Code ***Meaning***

X'04' A DFSBCB GET request was made to get QSAV storage in order to have enough save areas for use by those 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 request 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.

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.

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.

X'14' All storage pools manipulated by the storage pool managers using the IGETBUF/IFREEBUF or DFSPOOL macros must have a unique 4-character storage pool name. The pool name is used to find the storage pool header in the storage manager hash table. DFSIINS0 attempted 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.

ABENDU0718

Several Modules

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

| ABENDU0718 is a standard abend issued by one of these modules: DFSIINB0, DFSPLDR0, DFSPCC30,
| DFSPLPP0, DFSXBAT0, DFSXCTL0, DFSDINB0, DFSXLCI0, DFSXSL10, DFSXSTM0, and DFSXTRA0.
| The program status word (PSW) at entry-to-abend should be used to determine the failing module.

DFSPLDR0 is invoked 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 issues ABENDU0718.

DFSPLDR0

Analysis

ABENDU0718 is issued by DFSPLDR0 when the attempted load of required modules failed because (1) z/OS failed, or (2) the IMS module manager, DFSMODU0, rejected the load request. Message DFS677I or DFS678I and DFS679I is issued to the system console prior to the abend. 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

1. Hardware failure
2. PDS directory damaged
3. Required modules not on the PDS

DFSPLPP0

Analysis

ABENDU0718 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 prior to 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 may be invalid.

X'08' A permanent I/O error was detected.

Register 15 should be used to isolate to the specific label below.

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.

DFSXBAT0

Analysis

ABENDU0718 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.

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.

DFSXCTL0

Analysis

ABENDU0718 is a standard abend issued by DFSXCTL0.

Key	Label	Description
Reg15=return code	IMOL0219	SVC for IMODULE LOAD of DFSXCIC0 returned a nonzero return code. The load failed.

DFSXLIC0

Analysis

ABENDU0718 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.

DFSDINB0

Explanation

An IMODULE LOAD failed during DB Control (DBCTL) initialization.

Analysis

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

DFSXSL10

Explanation

An IMODULE LOAD or IMODULE GETMAIN in DFSXSL10 failed during IMS initialization.

Analysis

ABENDU0718 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.
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.

DFSXSTM0

Analysis

ABENDU0718 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.

DFSXTRAO

Explanation

The system was unable to load needed external trace modules or control blocks.

Analysis

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.

DFSIINB0

Analysis

ABENDU0718 is issued by DFSIINB0 CSECT LOADXITS when the attempted load of DFSCCMD0 fails and ICMD security is implemented using RACF (or equivalent), the Command Authorization exit routine, or both. No error messages are issued before this abend.

Key	Label	Description
Reg15=Return code	LOADXITS	The SVC for IMODULE LOAD of DFSCCMD0 returned a nonzero 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.

ABENDU0719

DFSIINB0

Explanation

The control region initialization was unable to open successfully any line groups. This is an internal IMS ERROR: 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.
```

ABENDU0720

DFSMINI0, DFSXIOB0

Explanation
IMS was unable to fix pages during initialization.

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. Please 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.

DFSMINI0

Analysis
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 locate DFSIDSP0 entry point.
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.

DFSXIOB0

Explanation
IMS was unable to page fix the OSAM I/O module (DFSAOS70) during OSAM I/O pool initialization.

Analysis
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.

ABENDU0721

DFSISMN0

Explanation
ABENDU0721 is issued from DFSISMN0 because an IFREEBUF call was issued to free a buffer that was not found in the pool.

Analysis
For ABENDU0721, find label IFSCAN. Several instructions after it you find instruction BZ IFNOBUFX. IFNOBUFX 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
The user has called DFSISMN0 to free a buffer, but the buffer is not in the pool. Program may be trying to free the buffer for the second time.

ABENDU0722

DFSISMN0

Explanation
ABENDU0722 is issued from DFSISMN0 when, after issuing an IWAIT, the ECB was not posted by storage management.

Analysis
ABENDU0722 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 722 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.

ABENDU0723

DFSXDCC0

Explanation
IMS was unable to locate the subsystem control table (SSCT) during IMS batch initialization.

Analysis
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 with label RETURN6 from which the abend (SVC 13) is issued. This 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. xx 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 xx is set by IMODULE.
Reg15=0006 00xx	RETURN6	OSAM I/O pool allocation failed. Return code xx is set by DFSSTM00.
Reg15=0007 00xx	RETURN6	IMODULE DELETE failed for work area. Return code xx 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, the copy of DFSMRCL0 for the highest level should be used.

ABENDU0725

DFSISMN0

Explanation
ABENDU0725 is issued from DFSISMN0 after issuing a DFS0CSS GETMAIN call to obtain additional ZIBs and FAQEs.

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.

ABENDU0728

DFSDLN00

Explanation
While attempting to make the IMS batch region nonswappable, the SVC encountered a problem. The return code is set in register 15. For an explanation of the IMSAUTH NOSWAP return codes, see the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

Analysis
ABENDU0728 is a standard abend issued by module DFSDLN00.

ABENDU0729

DFSSPM40

Explanation
During a DFSPOOL GET request, module DFSSPM40 was called to allocate a block from an existing storage pool. Unrecoverable internal errors were detected during processing.

ABENDU0729 is issued from both DFSSPM40 and DFSSPM50. During a DFSPOOL GET request, module DFSSPM40 is called to allocate a buffer from an existing storage pool. During a DFSPOOL REL request, module DFSSPM50 is called to release a buffer. If an unrecoverable internal error is detected by either module, a U0729 abend is issued.

Analysis
ABENDU0729 is a general abend issued by module DFSSPM40 and DFSSPM50. Register 15 contains the abend subcode.

Code	*Meaning*
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.

ABENDU0730

DFSXSPM0

Explanation

During system initialization, module DFSXSPM0 was called to perform storage pool manager initialization. An error was encountered during processing.

Analysis

ABENDU0730 is a general abend issued by module DFSXSPM0. Register 8 contains the abend subcode. For an explanation of IMODULE return codes, refer to the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

Code ***Meaning***

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 bit map 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.

ABENDU0731

DBFINI20, DFSIINB0, DFSIING0, DFSIINQ0, DFSXESI0

Explanation

The storage pool manager was called for a DFSPOOL ALLOC function, to allocate one or more of the storage pools defined in DFSSPM10. During the storage pool allocation process, one or more errors were encountered. The storage manager returned a nonzero return code which caused the caller to terminate processing.

Analysis

Register 2 contains the DFSPOOL parameter list address which points to one or more pool allocation entries. Each entry is mapped by the POOLALOC DSECT which is generated by DFSPOOL 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 or not 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 DFSPOOL return code encountered during processing for all entries. For an explanation of DFSPOOL and IMODULE GETMAIN return codes, refer to the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

ABENDU0732

Multiple Modules

Explanation
The storage pool manager was called for a DFSPOOL GET function to obtain a buffer from a previously allocated storage pool. An error was encountered during the buffer allocation process. The storage manager returned a nonzero return code, which caused the caller to terminate processing.

Analysis
Register 15 contains the DFSPOOL GET return code. For an explanation of the DFSPOOL return codes, refer to the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*. 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.

ABENDU0735

DFSCMI00, DFSCMS00, DFSICI00, DFSASLT0

Explanation
An IMS internal error has occurred. IMS detects an error has occurred while the CLB itask is running. Some other process invalidly resets the running flag CLB3DECB.

Analysis
PSW should tell where the abend occurs in DFSICI00 or DFSASLT0. IMS dump and logs are needed for problem diagnosis.

ABENDU0736

DFSTMEI0

Explanation
During IMS initialization, the GETMAIN to obtain storage for the Stop Region Work Area failed, because of insufficient storage.

Analysis
ABENDU0736 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.

ABENDU0738

DFSASK00, DFSCPY00, DFSUICC0

Explanation
ABENDU0738 may be a pseudoabend or a standard abend issued when an ISWITCH fails because the region is no longer there. The ISWITCH may 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.

ABENDU0741

DFSIIMS0

Explanation
An error occurred during IMS Multiple Systems Coupling (MSC) initialization.

Analysis
ABENDU0741 is a standard abend issued by DFSIIMS0. The program status word (PSW) at entry-to-abend will point 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, prior to abend, 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 will contain the return code passed back by the IMS SVC (DFSMODU0).
Reg14=BAL	IMOS0005	The IMODULE locate for DFSCMC40 failed. Register 15 will contain 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 executed 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.

Possible Cause

1. Insufficient amount of CSA to build MSC control blocks.
2. Loading or deleting of the MSC post handlers (DFSCMC20-DFSCMM20) failed because of an invalid request to SVC4.
3. No local SIDs for this system (probable system definition bug).

APAR Processing

Abend dump, console sheet.

ABENDU0742

DFSDLA00, DFSRDBC0, DFSDHD00

DFSDLA00

Explanation

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.

Analysis

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.

DFSRDBC0

Explanation

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.

Analysis

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.

DFSDHD00

Explanation
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

Analysis
Field PSTTSKID contains the identification of this task.

ABENDU0743

DFSSABN0

Explanation
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
ABENDU0743 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.

ABENDU0745

DFSSBMP0

Explanation
The IMS scheduler (DFSSBMP0) was called with an invalid function.

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
SSOB passed by DFSASK00 is invalid.

ABENDU0746

DFSDLA00

Explanation
DFSDLA00 detected an error indication on return from DFSDLOC0 on a CLOSE call at 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 ABENDU0746 if PSTOCBAD is on and another abend has not been initiated.

ABENDU0747

DFSASK00

Explanation
A second CREATE-THREAD was issued for the same TCB without issuing a TERM-THREAD.

Analysis
This is a pseudoabend issued by module DFSASK00.

Possible Cause
This is probably an IMS system error.

ABENDU0748

DFSCST00

Explanation
The IMODULE Macro encountered an error performing an IMODULE LOAD request for DFSCST00.

Analysis
ABENDU0748 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 will contain the return code. Refer to the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1* for an explanation of the return codes. The high-order half of register 15 will contain the reason code loaded by DFSCST00.

Key	Label	Description
Reg15 = X'000400xx'	Term	IMODULE LOAD failed for DFSTRM00.
Reg15 = X'000800xx'	DBCMDS	IMODULE LOAD failed for DFSDBDR0.

Attention: 'xx' is equal to the return code passed from the IMODULE macro.

ABENDU0749

DFSPCC20, DFSASK00, DFSSDA20, DFSSABN0, DFSBCK00

Explanation
ABENDU0749 is issued when a user abend code of zero has been issued by an application program. IMS uses this abend as a completion code for IMS messages and log records. Additional information can be obtained from your system console log.

This abend will also result from the subsystem function 'END OF TASK' doing the TERMINATE-THREAD processing after a dependent region abends. The IMS (E)STAE either did not get control or terminated abnormally before completing TERMINATE-THREAD processing.

Analysis
ABENDU0749 is a standard abend issued by modules DFSASK00 and DFSPCC20 for the reason given in the Explanation section of this abend.

ABENDU0756

DFSCST00

Explanation
In an XRF environment, either the IMODULE LOAD failed for DFSQDOC0, or an OPEN failed for one or more of the local message queue data sets. IMS also issues ABENDU0756 if the size of the local message queue is too small. This occurs when the number of records in any of the three data sets (IMS.LGMSGL, IMS.SHMSGL, and IMS.QBLKSL) is less than the number of records specified in the SHUTDWN parameter of the MSGQUEUE macro during system definition.

Analysis
ABENDU0756 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 a description of these codes, see the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

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 in *IMS Version 9: Messages and Codes, Volume 2*.

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.

ABENDU0757

DFSQC010, DFSQC020, DFSQC030, DFSQC040, DFSQC060, DFSQC070,

DFSQC080, DFSQBFM0, DFSQDQ00, DFSQEQ00, DFSQFIX0, DFSQGU00,

DFSQIS00, DFSQLOG0, DFSQMGR0, DFSQNP00, DFSQRH00, DFSQRL00,

DFSQRSQ0, DFSQRST0, DFSQXF00

Explanation

An error condition which should not occur was *detected* by the Queue Manager routine in the IMS system. It can be caused by a logic error in any of the Queue Manager's callers, by an operations error relating to an emergency restart situation, or by a logic error within one of the Queue Manager routines.

Analysis

ABENDU0757 is a standard abend issued by the Queue Manager from any of the modules listed under the **ABENDU0757** header on this page. 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 ABENDU0757. 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.

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, QBFALSTD, 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
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
02C	DFSQEQ00
02D	DFSQGU00
02E	DFSQRL00
02F	DFSQRL00
030	DFSQC020
031	DFSQC020
032	DFSQC020
033	DFSQEQ00
034	DFSQEQ00
035	DFSQEQ00
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 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
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
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

DFSQC010

Analysis

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.

DFSQC020

Analysis

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.

SC	Key Data	Label	Description
04C	Reg15=X'4C' QSAPAB15=DFSBCB return code	QSMFRE25	An attempt to obtain an MSGP type message block failed.

DFSQC030

Analysis

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 will abend. 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 QC310080 QC310090 QC330005 QC330010 QC33RCT	An attempt to release the Queue Manager latch failed with QSAPAB15 containing the Latch Manager return code.
052	Reg15=X'52' Reg4=A(latch header)	QC300005 QC310005	An invalid latch header address was detected.
053	Reg15=X'53' Reg8=latch mode	QC300020 QC320005	An invalid latch mode was requested. Valid modes are SHR or EXCL.

DFSQC040

Analysis

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.

DFSQC060

Analysis

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 QXCDEQ	An attempt to release the DC system latch obtained by the Queue Manager failed.

DFSQC070

Analysis

The DFSQC070 module contains two entry points (DFSQC070, DFSQC071), one of which (DFSQC071) issues a reason code. DFSQC071 Queue Manager decrement QPOOL nopurge 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.

DFSQC080

Analysis

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.

DFSQBFM0

Analysis

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(DECB) DECBRBN=Req type code	QBMLCT QBMLCTA QBMRDSU QBMSNID2	The 'D' portion of the DRRN (byte 0 of DECBRBN) does not equal X'00', X'04', X'08', or the in-storage X'88'.
101	Reg15=X'101' Reg1=0 Reg1=DECBRBN & X'00FFFFFF' Reg9=A(DECB)	QBMLCTB QBMSNID3	After stripping the 'D' from the DRRN, the remaining 'RRN' portion was found to be zero.
102	Reg15=X'102' Reg3=(DECBTYPE-1) & X'0000001F' Reg3>X'14' Reg9=A(DECB)	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=DECBRBN Reg4=A(QBFFR)= A(input buffer) NOT(QBIDL≤DECBRBN≤QBIDH)	QBINT050 QBINT070	The input DRRN was not included in the specified buffer range.

DFSQDQ00

Analysis

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	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 a 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 a 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) MSGCFLG1 bit: MSGC1RAC on QDFLG2 bits: QDF2CNT on	DEQ13B	During a DEQUEUE call the RACF segment item was not in the prefix although the prefix flag indicates that it exists.
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.

SC	Key Data	Label	Description
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.

DFSQEQ00

Analysis

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' reenq FIFO

X'0F' reenq LIFO

X'2F' reenq 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.
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.

SC	Key Data	Label	Description
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 zeroes 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.
041	Reg15=X'41' Reg6=A(error detected Reg7=A(QDEST) Reg8=A(QTPPCB) Reg9=A(DECB) 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.

SC	Key Data	Label	Description
04C	Reg15=X'4C' Reg5=A(QSAPWKAD) Reg6=A(error detected Reg7=A(QDEST) Reg8=A(QTPPCB) Reg9=A(DECB) 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.

DFSQFIX0

Analysis

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.

SC	Key Data	Label	Description
801	Reg15=X'801' Reg10=A(message) QSAPAB15=CHKRCD return code	MCHNCK4	A non-recoverable message was detected while checking the record of a message.
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 a X'3C' log record, the block was not a 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 a 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 a 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.

SC	Key Data	Label	Description
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.

DFSQGU00

Analysis

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.
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	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.

SC	Key Data	Label	Description
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.
026	Reg15=X'26' Reg10=A(message) Reg8=A(QTPPCB)	GU82 GU83	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	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	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	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.

SC	Key Data	Label	Description
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	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	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.
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	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	Reg15=X'50' QSAPAB15=DFSBCB return code	QGU09_200 QGU09_400	An attempt to release the Queue Manager message buffer work area (QMBA) failed during a GU call.
058	Reg15=X'58' Reg8=A(QTPPCB) QTPDST=0	QGUCONT1 QMGGUT DLIGU010 BYP60500	The destination address was zero during a GU call.
060	Reg15=X'060' Reg3=A(parmlist) Reg5=A(QSAP) Reg6=A(log record) Reg10=A(message)	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.

SC	Key Data	Label	Description
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	The return code value returned from a called routine is invalid.

DFSQIS00

Analysis

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	ISMF10J	Is not the first segment of the message text.
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's LRECL.
006	Reg15=X'06' Reg3=Length of remaining message text Reg10=A(message)	ISMC30	During an INSERT-MOVE call, the previous caller exceeded the destination DCB's LRECL.
00A	Reg15=X'0A' Reg11=A(SCD) SCDSSTYP 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 MSGFLAGS bit: MSGFLAST on	ISICMP	Invalid continued message segment. The last DRRN of the message was encountered.

SC	Key Data	Label	Description
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.
03E	Reg15=X'3E' Reg11=A(SCD) SCDPRDEF bit: SCDPDMUL off	PREFO21	An UPDATE MSC FLAGS was requested during an INSERT-PREFIX call, but the MSC was not in the system.
046	Reg15=X'46'	PREFO21C	Invalid system segment for MSC during processing of the UPDATE MSC FLAGS request during an INSERT-PREFIX call.
046	Reg15=X'46'	PREFO22A	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	PREFO29	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	PREFO29B	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	PREFO29C	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.

DFSQLOG0

Analysis

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 w/o 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.

SC	Key Data	Label	Description
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 non-recoverable 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.
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=DFSPOOL 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's five queues.
316	Reg15=X'316' Reg4=A(message)	LMSGA	The expected system extension segment, which contains the UTC time stamp, was not found.

DFSQMGR0

Analysis

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.
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.
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	QMGRLATE	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.

DFSQNP00

Analysis

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.

DFSQRH00

Analysis

DFSQRH00 is the new parm list interface. The parm 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.

DFSQRL00

Analysis

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=DFSFNDST 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.

DFSQRSQ0

Analysis

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)	UPDMSGA0 UPDMSGA1	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 a 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 UPDPFXI2	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.

DFSQRST0

Analysis

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¬=X'01', X'03', or X'07' and NOT(X'30'≤MSGLCODE≤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¬=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) DECBRBN¬=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 prior to 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¬=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.
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 a X'33' log record without a work area.

SC	Key Data	Label	Description
215	Reg15=X'215' Reg5=A(message) MSGFLAGS bit: MSGFFRST on MSGCFLG2 bit: MSGC2APP on	PISRT32	When attempting to insert a 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	While processing a X'31' log record, the destination was not found.
216	Reg15=X'216' Reg5=A(QLENQRCD)	PENQ20 PENQ31B	While processing a X'35' log record, the destination was not found.
216	Reg15=X'216' Reg5=A(QLXFRRCD)	PXFER27	While processing a X'37' log record, the destination was not found.
216	Reg15=X'216' Reg5=A(QLRIRCD)	PRELI10	While processing a X'38' log record, the destination was not found.
216	Reg15=X'216' Reg5=A(QLFRERCD)	FENDC20	While processing a X'33' log record, the destination was not found.
216	Reg15=X'216' Reg5=A(QLMCRCD)	CLNC08 CLNDCUR	While processing a 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 a 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 a 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 a 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.
21D	Reg15=X'21D' Reg5=A(QLENQRCD) or A(QLDQSRCD)	PENQ60 PDEQ30A	While processing a 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 a 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	There was no APPC system segment header during BUILDQ processing although the flag indicated that it existed.

DFSQXF00

Analysis

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 zeroes 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=DFSFNDST 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.
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 Func=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 xxxx
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 cancelled
..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	Invoke AOI processing for the specified command
..01 0010	AOI SYSMSG	Invoke AOI processing for the specified message
..01 0011	Abort	Abort 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

Function	Name	Description
..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

ABENDU0758

DFSQC010

Explanation

ABENDU0758 is issued from DFSQC010 because a message queue data set overflowed before the system could be shut down with an internal checkpoint dump queue.

For IMS running with Shared Queues, the ABENDU0758 is issued from DFSQC010 because all of the available DRRNs have been exhausted.

Analysis

ABENDU0758 is a standard abend. The program status word (PSW) at entry to abend points to the failing module.

Possible Cause

This abend is usually 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'
  - Reg9+X'0C'= DECBAREA = Address of the text

For shared queues this abend is usually caused by setting QBUFMAX. If QBUFMAX is set 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 on the QBUFMAX parameter, see the *IMS Version 9: Installation Volume 2: System Definition and Tailoring*.

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'.

Key	Label	Description
REG2='QBLK' or 'SMSG' or 'LMSG'	ASSIGN65	Register 2 contains the identity of the data set in error.

If register 2 contains 'SMSG' 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 *IMS Version 9: Installation Volume 1: Installation Verification* and the information about message queue data set allocation in *IMS Version 9: Administration Guide: System*.

ABENDU0759

DFSQBFM0

Explanation

ABENDU0759 is issued from DFSQBFM0 because an unrecoverable error occurred in the queue manager while attempting to read or write for the message queue data set. The I/O operation is retried three times. The abend is issued if retry is not successful.

Analysis

ABENDU0759 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. Refer to *IMS Version 9: Messages and Codes, Volume 2* for a complete description of the I/O error encountered.
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=DECBIOBE There is no more room in CSA storage to allocate an IPAGE for additional IOSBs.

Key	Label	Description
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.

ABENDU0760

DFSQDOC0

Explanation

An error was detected by DFSAOS10 while opening one of the following system data sets:

IMS.QBLKS

IMS.SHMSG

IMS.LGMSG

IMS.SPA

Analysis

ABENDU0760 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. MSGDFS986A, issued just prior to ABENDU0760, 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.

ABENDU0762

DFSDSPX0

Explanation

A system error was detected during IMS cross-memory processing.

Analysis

ABENDU0762 is issued by module DFSDSPX0 when it detects severe internal errors when processing a cross-memory function. Register 14 in the abend registers will point to the address within DFSDSPX0 where the error was detected. Register 15 will contain one of the following reason codes:

Code ***Meaning***

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 node of an ITASK.

Probable Cause
All of these reason codes denote IMS internal errors.

ABENDU0763

DFSQBFM0, DFSQCP00, DFSQMGR0

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.

Analysis
ABENDU0763 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.

Key	Label	Description
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.

ABENDU0764

DFSAOS80

Explanation
The OSAM access method was unable to convert an RBN to a full-disk address in the form MBBCCHHR during WRTQUED processing.

Analysis
ABENDU0764 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 INTERR 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.

ABENDU0765

DFSAOS80

Explanation
OSAM encountered virtual to read address translation error or a page fix error while initiating an I/O operation. The DECB was subsequently posted with other than an X'7F' condition-code.

Analysis
ABENDU0765 is a standard abend issued by DFSAOS80. The error registers are in the abend SVRB. The program status word (PSW) at entry-to-abend will point 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.

ABENDU0766

DFSIDSP0

Explanation
A system error was detected during IMS ITASK dispatching.

Analysis
ABENDU0766 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:

Code	*Meaning*
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 redispatched has never IWAITed before.

ABENDU0767

DFSIDSP0

Explanation
A system error was detected while processing an ISERWAIT or IWAIT call.

Analysis
ABENDU0767 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:

Code	*Meaning*
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.
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.

ABENDU0768

DFSIDSP0, DFSREP00

Explanation

A system error was detected while processing an ISWITCH call

Analysis

ABENDU0768 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:

Code	Meaning
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 redispatched after an ISWITCH has an invalid post code. The code must be either " TO" or " RET".
X'07'	An ECB was being redispatched 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.

ABENDU0769

DFSIDSP0, DFSREP00

Explanation
A system error was detected while trying to initialize dispatcher control blocks.

Analysis
ABENDU0769 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:

Code	*Meaning*
X'01'	On an INITECB call, the ECB being initialized was either on a chain, or was already waiting. Register 8 points to the ECB prefix.
X'02'	On a 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 is 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.
X'0A'	An obsolete SCP post routine was called.

ABENDU0770

DFSIDSP0, DFSREP00

Explanation
A system error was detected while a dependent region was attempting to OPEN or CLOSE a dependent region.

Analysis
ABENDU0770 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:

Code	*Meaning*
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.

ABENDU0773

DFSDDLC0, DFSDDLE0

Explanation

The application received a read or open error from the buffer handler during an ISRT or DLET call that required a single-call backout to remove updates made during the call. The single-call backout failed for the following reasons.

- No disk log for dynamic backout was specified in batch.
- Positions in other PCBs in the PSB were adjusted.
- Backout failed.

Analysis

ABENDU0773 is a pseudoabend issued by DFSDDLCO or DFSDDLEO 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, ABENDU0773 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).

ABENDU0774

DFSPIEX0

Explanation

A PST that was waiting for a lock for a resource out of module DFSPIEX0 was posted out of the wait with a post code that was neither a X'60' nor a 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 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 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.

ABENDU0775

DFSFXC10

Explanation

This abend, issued out of DFSFXC10, is either a standard or a pseudoabend.

Case 1 A standard abend is issued at label XC10ABND. The registers at entry-to-abend or in the abend SVRB, taken from the ABENDU0775 dump, should be used to determine which of the CASE1 entries applies to the current failure.

Case 2 A pseudoabend is set up at label NOCORE. The information about message DFS2450I in the paragraph below is helpful in determining the cause of the ABENDU0775 and in obtaining a resolution.

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 desired console, then the routing codes can be changed by using the MCS parm of the IMSCTRL macro (see *IMS Version 9: Installation Volume 2: System Definition and Tailoring*).

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 may 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/DEQUEUE storage size in *IMS Version 9: Administration Guide: System*. The write-up contains a formula that should be used to determine the maximum amount of storage that ENQ/DEQ could possibly use at any given time. A separate calculation must be done for each MPP and BMP that will be running and the separate calculations then totaled. The PIMAX parameter should be larger than this total so that ABENDU0775 can be avoided.

Attention: Taking checkpoints more frequently in a BMP will decrease the total storage size required. If running under CICS and not using DBCTL, the value specified by the PIMAX and PIINCR parameters will be handled differently. A CICS initialization parameter overrides the second parameter of CORE, which specifies maximum storage. For a more detailed discussion of how the ENQUEUE pool is handled in CICS, see *IMS Version 9: Installation Volume 2: System Definition and Tailoring*.

If this abend still persists, program isolation (PI) trace (see /TRACE command in *IMS Version 9: Command Reference*) should be run and the results printed using DFSERA10 with the DFSERA40 exit routine. The results should be analyzed to determine the number of ENQs per PST that are still outstanding at the time this abend occurred. This number should then be used by PST in the formula mentioned to recompute the total size of storage required.

If message DFS2450I cannot be located, obtain a dump by using the technique under “Possible Cause”, following. This dump will help 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.
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 can not 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 can't 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.

Key	Label	Description
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.
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 isn't an enqueue request, and a matching QCB wasn't 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 ABENDU0775, get a normal ABEND dump at the time the error was detected. To get the dump, change the unconditional branch to ENQXTA to an unconditional branch to XC10ABND (a few instructions after label NOCORE).

ABENDU0776

DFSECP10, DFSECP20

Explanation

A failure occurred in dynamic backout while processing a ROLB call.

Analysis

ABENDU0776 is a pseudoabend issued by DFSECP10 or DFSECP20, depending on the region type. It is detected by DFSRBOI0 and set up by DFSFXC30. Message DFS981I accompanies this abend and should be used to determine the reason for the failure. Batch backout must be run for the databases named in the DFS981I message.

ABENDU0777

DBFIRC10, DBFLRH00, DFSESPR0, DFSLRH00, DFSFXC10, DFSCMD30, DFSCMD60

Explanation
The application program was abended to break a potential deadlock. A message processing program (MPP) is automatically rescheduled.

Analysis
ABENDU0777 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 CREATE THREAD return code 24 indicating a deadlock situation. The external subsystem can issue a message indicating the cause of the problem. The external subsystem should be consulted to determine why resources were not available. IMS takes the following actions:

- Invokes the external subsystem Resolve In-Doubt exit routine
- Places the input message back at the top of the SMB message queue
- Discards all uncommitted output

When issued from DFSCMD30 or DFSCMD60, the abend breaks a potential deadlock between a DBR command, a MODIFY COMMIT command, and an MPP region with an application program issuing an ICMD or CMD call.

ABENDU0778

DFSDLA00

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 active regions being terminated as a result of an internally generated ROLL call.

Analysis
ABENDU0778 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 reexecuted. 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. ABENDU0778 is normal for this situation.

ABENDU0779

DFSDLA00

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
ABENDU0779 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 enqueuing more resources than it needs and perhaps going into a loop. For information on the *Q command code, see *IMS Version 9: Application Programming: Database Manager*.

ABENDU0780

DFSCLM00, DFSCLM10, DFSCLM20, DFSDBDR0, DFSDTTA0, DFSASK00

Explanation
The IMS latching routine detected a critical system error.

Analysis
| ABENDU0780 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 `DFS655I`, 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 SAPACLLE. The CLLE definitions are listed below.

The following codes are for DFSCLM00 abends. DFSCLM00 gets the latch for the requester.

Code	*Meaning*
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.
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.

Code	*Meaning*
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.

Code	*Meaning*
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.

Code	*Meaning*
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).

Code ***Meaning***

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 specified.

X'206' Exclusive latch — recovery exit routine specified but none provided.

X'207' Exclusive latch — recovery exit routine specified and driven but failed and returned nonzero return code.

X'208' Shared latch not owned.

ABENDU0783

DFSDPDM0

Explanation

The DMB and PSB Pool Space Manager attempted to delete and release 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.

ABENDU0790

DFSUSE00, DFSUSE10, DFSUSE20

Explanation

The IMS USE manager routine detected a critical system error.

Analysis

ABENDU0790 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 inuse, lock, connect, merge or inquire functions for the requester.

Code ***Meaning***

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 inuse function prior to issuing a lock function.

X'004' The USE/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 inuse requester already has an inuse on the resource.

X'009' User request = INUSE/LOCK/CONNECT. The resource has been logically deleted.

X'010' User request = MERGE. The requestor does not have a LOCK on the 'FROM' resource.

USE01000, a subroutine in DFSUSE00, allocates CULE to latch requesters.

Code ***Meaning***

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' Unable to allocate an AWE block for the ITASK requester. Attempting to allocate an AWE to call DFSBCB00 to release CULE.

X'022' Unable to deallocate a CULE block for an ITASK requester.

X'023' Unable to deallocate an AWE block for an ITASK requester. The AWE is used to deallocate the CULE.

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 an 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.

DFSUSE10 does no-use, unlock, or disconnect functions.

Code ***Meaning***

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 parm 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.

Code ***Meaning***

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) inuse ownerships from the AWE queue.

Code ***Meaning***

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 or work.

Code ***Meaning***

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 or 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 or 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 or work.

DFSSLC00, DFSSLC10

Explanation

The IMS latching routine detected a critical system error.

Analysis
ABENDU0790 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:

Code	*Meaning*
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:

Code	*Meaning*
X'01'	Passed ECB mismatched with ECB attached to SAP.
X'02'	Invalid search type has been specified.

ABENDU0791

DFSAERS0, DFSAERN0, DFSAERC0

Explanation
The ODBA Syncpoint TCB initialization or processing detected a critical system error.

Analysis
ABENDU0791 is a standard abend that can be issued by modules DFSAERS0, DFSAERN0, or DFSAERC0.

DFSAERS0

Code	*Meaning*
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.

DFSAERN0

Code	*Meaning*
X'04'	DFSBCB GET call failed.
X'20'	ODS Function invalid.

DFSAERC0

Code	*Meaning*
X'04'	DFSBCB GET call failed.

ABENDU0793

DFSISTS0

Explanation
Module DFSISTS0 was unable to open either the LOGIN or LOGOUT DD statements.

Analysis
This is a standard abend issued by module DFSISTS0.

Possible Cause
z/OS error.

ABENDU0794

DFSISTS0

Explanation
Module DFSISTS0 attempted to link to the SORT program and encountered a nonzero return code.

Analysis
ABENDU0794 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.

ABENDU0795

DFSISTS0

Explanation
An invalid log record type was encountered in the output phase of the SORT program.

Analysis
ABENDU0795 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:

Code	*Meaning*
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 will point to the instruction within label BADSORT from which the abend (SVC 13) is issued. Register 7 in the abend SVRB will point 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.

APAR Processing
Console sheet, dump, copy of IMS log.

ABENDU0796

DFSDLD00

Explanation
While processing a REPL call, the replace module DFSDLDR0, within DFSDLD00, found that the root segment was not locked by Retrieve.

Analysis
ABENDU0796 is a pseudoabend issued from module DFSDLDR0.

ABENDU0796 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 below.

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).

ABENDU0797

DFSDLD00

Explanation

Delete/Replace received an unexpected segment from the buffer handler.

Analysis

ABENDU0797 is issued from module DFSDLD00 and can be a standard or a pseudoabend. ABENDU0797 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***
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.
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.

ABENDU0799

DFSDLD00, DBFCMP10

Explanation

Delete/Replace called the user's compression routine to expand or compress a segment. On return, Delete/Replace found that the new 'LL' for the segment was incorrect.

Analysis

ABENDU0799 is issued from module DFSDLD00 and can be a standard or a pseudoabend.

ABENDU0799 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 used below.

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

1. Compressed-data length was greater than 10 more than DMBSGMX.
2. Error in Edit/Compression routine, or the Edit/Compression routine was passed meaningless data.

DBFCMP10

Explanation

After returning from the user's segment compression exit routine, the key was altered or the length (LL) was incorrect.

Analysis

ABENDU0799 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

1. The LL is greater than or equal to the specified minimum length in 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)

Reg5 returned LL

Reg6 address of segment

Reg7 CI size (120 bytes)

Reg8 A (DMAC)

Reg10 A (EPST)

The maximum (SDBLMAX) and the minimum (SDBLMIN) can be found by locating A (SDBS) from MLTESDBS. The CI size is DMACBLKL.

ABENDU0800

DFSDLR00

Explanation

Retrieve called the user's segment edit/compression routine to expand a compressed variable-length segment. The segment returned had a length greater than the maximum allowable or was less than or equal to 2 bytes.

A PCB with PROCOPT=GO, GON, or GOT (GOx) detected an invalid length in a compressed segment's LL field while processing in the VLEXP routine in DFSDLR00. The length value in the LL field exceeded the segment's maximum length by more than 10 bytes or was less than or equal to 2 bytes. This is probably a result of an update transaction changing the segment data when it is about to be expanded for the PROCOPT=GOx PCB.

Analysis

ABENDU0800 is a pseudoabend issued from module 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	VLLEN2	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.

ABENDU0801

DFSDLR00

Explanation

There are three reasons for the abend:

1. The 'PSBIOAWK' work area is too small to hold a concatenated key or segments to be returned. This problem may result from a user error if the user defines an area too small in the PSB GEN (PSBGEN statement, IOASIZE parameter) or if a path call has been specified in such a way that the combined length of the concatenated segments to be returned to the application would be greater than 65,535 bytes. This problem may result from an IMS error if the 'IOASIZE' parameter is omitted in the PSB GEN, in which case the size of the 'PSBIOAWK' would be determined by the block builder.
2. Storage manager issued a nonzero return code in response to a GET BUFF call from DFSDLR00 to the 'DBWP' pool. This code usually indicates there is not enough room in the 'DBWP' pool to satisfy the request.
3. The length field in a variable length segment was either less than 2 or greater than the defined maximum length.

Analysis

ABENDU0801 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 retrieve trace is described in *IMS Version 9: Diagnosis Guide and Reference*. 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 a 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	RDLP200 (RDLPABND)	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	RDPR010 (RDPRABND)	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)	CCTOMEB (CCTOMEER)	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.
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 PSBGEN.

APAR Processing
Listings of DBDGEN and PSBGEN.

ABENDU0802

DFSDLD00

Explanation
A REPLACE of a variable-length segment indicates that 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
ABENDU0802 is issued from module DFSDLD00 and can be a standard or a pseudoabend.

ABENDU0802 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*
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.

ABENDU0803

DFSDLD00

Explanation
Delete/Replace was unable to find a logically related segment (usually a LP of a physically paired LC); or a logical parent counter is negative if decreased for deletion of a LC; or a logical child could not be found on a logical twin chain.

Analysis
ABENDU0803 is issued from module DFSDLD00 and can be a standard or a pseudoabend.

ABENDU0803 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*
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

1. After database reorganization, Prefix Update was not run prior to any other calls to the database.
2. During a delete of a physically-paired logical child, equal intersection data was not loaded for the logical children in a physical pair.
3. After the Prefix Resolution utility was run, the return code, if not 0, was not checked for implications before Prefix Update was run.
4. When using symbolic keys, the logical parent did not exist.
5. If a logical structure exists, pointers can have gotten lost when using nonunique keys.

ABENDU0804

DFSDLD00

Explanation

There was insufficient storage in the control region for Delete/Replace to obtain storage for building work areas, or the work-area chains were invalid.

Analysis

ABENDU0804 is issued from module DFSDLD00 and can be a standard or a pseudoabend.

ABENDU0804 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***
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

1. VS SORT did not release 80 bytes of allocated storage when SORT was invoked from COBOL V3.1.
2. The application was run in a batch region in overlay mode, but the application macro was not coded to reflect this.
3. For a DB/DC environment, the DBWP is too small.
4. For a batch environment, the region size is too small.

ABENDU0805

DFSDLA00

Explanation
During Reload processing for a partition of a High Availability Large Database (HALDB), one of the following was invalid:

- Prefix
- Prefix length

Analysis
ABENDU0805 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.

ABENDU0806

DFSDLD00

Explanation
Delete/Replace received an invalid return code from the Buffer Handler. The request to the Buffer Handler may have been either a Mark Buffer Altered, a Byte-Locate call, an Erase-Logical-Record call, an SETL, or a Block-Locate call.

Analysis
ABENDU0806 is issued from module DFSDLD00 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

ABENDU0806 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 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.

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.

ABENDU0807

DFSDLD00

Explanation

Delete/Replace was unable to find the path to the root for a segment to be deleted or replaced. The failure may have occurred while searching the logical blocks or while searching the physical path to the root.

Analysis

ABENDU0807 is issued from module DFSDLD00 and can be a standard or a pseudoabend.

ABENDU0807 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*

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.

ABENDU0808

DFSDLD00

Explanation

Delete/Replace compared the physical database blocks with the logical structure to validate the existence of logical-parent and logical-child relationships, and found an error in the blocks or could not calculate the address of the PSDB for the segment code just retrieved.

Analysis

ABENDU0808 is issued from module DFSDLD00 and can be a standard or a pseudoabend.

ABENDU0808 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.

Key	Description
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.
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.

ABENDU0810

DFSDLOC0

Explanation

The user program attempted to load a database and was not sensitive to all data set groups within the database.

Analysis

ABENDU0810 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 will contain the registers when the open/close module was called. Register 1 in this save set will point 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.

APAR Processing
PSBGEN, DBDGEN parameter, copy of the system log, and dependent region dump.

ABENDU0811

DFSDLD00

Explanation
The Delete module of Delete/Replace was attempting to change a pointer in the prefix section of some segment, and the old value of the pointer was incorrect.

Analysis
ABENDU0811 is issued from module DFSDLD00 and can be standard or a pseudoabend.

ABENDU0811 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.

ABENDU0812

DFSDLDC0, DFSDLR00

Explanation
A return code greater than 4 was returned by the randomizer routine.

Analysis
ABENDU0812 is a pseudoabend issued by modules DFSDLDC0 and DFSDLR00.

ABENDU0814

DBFDCAP0, DFSECP10, DFSECP20, DFSPCC30, DFSCPY40

Explanation

An unrecoverable error was detected by data capture during processing required because changed data user exit routines or changed data logging was defined for the database. The type of error depends upon the module generating 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, Volume 1 (ABEND-HSPSERV)* for an explanation of CSRCESRV return codes.

DFSECP10/20 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:

The first byte identifies the module detecting the error as follows:

04 DFSDCAP0

The second byte identifies the routine detecting the error as follows:

04 DO BYTE LOCATE

08 BYTE LOCATE

12 BUILD LOGICAL PARENT CONCATENATED KEY

16 FIND POINTER

20 GET WORK

24 GET STORAGE

28 DELETE PATH DATA

The third byte is a reason code for the error.

04 Subcode is DMB segment code.

08 Subcode is PST return code.

16 Subcode is capture function that failed.

The fourth byte is a subcode for the error.

nn PST return code if REASON code 04.

nn DMB segment code if REASON code 08.

04 UNABLE TO GET WORK STORAGE IF REASON CODE 16.

08 NO DATA SPACE STORAGE AVAILABLE IF REASON CODE 16.

ABENDU0816

DFSDLR00

Explanation

Index synchronization error. The contents of a secondary index do not correspond to the contents of the target database. This 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.

ABENDU0819

DFSFXC30, DFSDSC00

DFSFXC30

Explanation

The sync point processor (DFSFXC30) was called to process an undefined function.

If SPOOL/API processing was occurring, the system abended for the following reasons:

- A 'CLOSE' failure occurred during phase 1 of sync-point and the decision was made to abort sync-point processing.
- A participant voted 'NO' in phase 1 of sync-point.

Analysis

ABENDU0819 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:

Code ***Meaning***

X'01' MSG get unique code

X'42' Normal/abnormal termination

X'79' Sync 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.

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.

DFSDSC00

Explanation
A COMMIT request was made without a previous PREPARE request.

ABENDU0820

DFSDLA30, DFSFXC40

Explanation
One of the following errors has been detected in either module DFSDLA30 or module DFSFXC40.

1. 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.
2. 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 DFSICLR0. 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
ABENDU0820 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.

DFSDLA30

Analysis
ABENDU0820 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.
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.

Key	Label	Description
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.

DFSFXC40

Analysis

ABENDU0820 is a standard abend issued by DFSFXC40, the communications sync-point processor. DFSFXC40 is called by DFSFXC30 to handle the communications portion of a sync-point. 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.

ABENDU0821

DFSBIND0

Explanation

An error has occurred during initialization of the program and database control blocks while executing a DBB type IMS region.

Analysis

ABENDU0821 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).

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.
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 *not* 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 BLDFLAG 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.

ABENDU0822

DFSDLA30

Explanation

A single message queue segment will not fit into the PSB index work area.

Analysis

ABENDU0822 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 *not* fit into the XWA.

ABENDU0824

DFSDLA30

Explanation

IMS received an invalid return code from the user's program routing exit routine.

Analysis

ABENDU0824 is a standard abend issued from DFSDLA30, the communications DL/I call handler. The multiple system (MS) program routing exit module (DFSCMPR0) 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:

Code	*Meaning*
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.

Key	Label	Description
Reg14=BAL Reg15<zero or>X'08'	CHGNONC	An invalid return code was passed back to DFSDLA30 by DFSCMPR0.

ABENDU0825

DFSDDBH0, DFSDXMT0

Explanation

An unexpected return code from the buffer handler (DFSDDBH0) occurred or a catastrophic error occurred while executing a move character long (MVCL) instruction.

Analysis

ABENDU0825 is a pseudoabend detected by DFSDXMT0, the DL/I index maintenance module. The SNAP of the control blocks will be 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 will indicate the DSG in which the call was performed and the contents of PSTBYTNM passed to DFSDVBH0.

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 will be 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.

ABENDU0826

DFSDXMT0

Explanation

A situation was encountered by index maintenance that prohibits further processing of the database.

Analysis

ABENDU0826 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) the PST, which can be found on the SNAP of the controls blocks, and (2) message DFS840I, 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 DFSDVBH0) or OPEN/CLOSE (DFSDLOC0).

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 *open* 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.

ABENDU0827

DFSDXMT0

Explanation
An error occurred during the LATCH/UNLATCH or IMODULE LOAD processing related to use of the index exit routine.

Analysis
ABENDU0827 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 will contain the registers when the index maintenance module was called.

This abend will only occur 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 will point to the Index Maintenance parameter list (DMXMPRM). At label DMBXMXNM is the name of the user supplied exit routine, and register 15 will contain 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 will be provided as a guide through the logic and the KEY information will be 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 *not* 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.

I/O error occurred while reading the exit routine from the library.

APAR Processing
Assembly of DBDs, dependent region dump, and copy of the system log.

ABENDU0828

DFSDXMT0

Explanation
Index maintenance attempted to insert a new index entry, with index specified as unique, and encountered a duplicate index entry.

Analysis
ABENDU0828 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 will contain the registers when the index maintenance module was called.

The printout of the PST will be important and should be located. Fields PSTRTCDE and PSTFNCTN are the two important fields. PSTWRK4 will contain the SSA of the index entry segment. Message DFS840I will precede the abend and will be issued to the IMS master console. The status code will be 'NI'.

If duplicate secondary index entries occur, the index should be specified nonunique, 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.

APAR Processing
Copy of the system log.

ABENDU0829

DFSDXMT0

Explanation
When a VSAM erase call was issued, an invalid return code was returned from in the VSAM buffer handler.

Analysis
ABENDU0829 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 will contain 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.

Key	Label	Description
PSTABTRM=X'1000033D' PSTFNCTN=X'F1'	DDLTVSAM	An erase operation returned an invalid buffer handler return code.

APAR Processing

Copy of the system log, dependent region dump.

ABENDU0830

DFSDLA30

Explanation

This pseudoabend is issued when an application program issues a message GU call and the TM call handler (DFSDLA30) detects that the message has invalid fields (such as MSC SYSIDS, IMSID, or SOURCE name). This error can result from one of the following situations:

- The resources from which these fields are set are changed while the message is on the queue in a shared queues (SQ) IMSplex.
- The resources from which these fields are set are changed while the message was inflight in a remote IMS in a MSC network.
- An IMS internal error occurred.

Analysis

ABENDU0830 is a standard pseudoabend that is issued by DFSDLA30. The following events occur:

- Message DFS554A is issued to the master terminal.
- A pseudoabend ABENDU0830 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, with the following control cards:

```
OPTION   PRINT O=5,V=6701,L=2,C=M,T=X,E=DFSERA30
OPTION   PRINT O=9,V=LA3E,L=4,C=E,T=C,E=DFSERA30
OPTION   PRINT O=5,V=67D0,L=2,C=E,T=X,E=DFSERA30
```

The 6701-LA3E record contains 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)

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
```

Note that 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 ABENDU0830 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
```

- Reg15 is the U830 unique reason code. These codes range from X'00000001' through X'00000019'. The remaining registers, saved when the error was detected, are from module DFSDLA30.
- 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 is the time 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 is a message 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 (should be 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

Note that DFSQCLD0 is OCO. You might need to contact IBM Software Support to analyze this information.

The ABENDU0830 Reg15 reason code descriptions are provided in the following table.

Key	Label	Description
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 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 is an IMS internal error.
Reg15=00000006		The source SYSID at PSTSIDS, taken from the message at MSGMSIID (one byte SYSID) or MSGSRSID (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, this is a procedure error. 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 MSGSRSID (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, this is a procedure error. That is, a remote IMS with remote SYSIDs was brought up alone without bringing up the front end IMS in the SQG. Bring up the front end IMS and release the message to process again. Otherwise, this error is probably an IMS internal error.
Reg15=000000008		The source SYSID at PSTSIDS, taken from the message at MSGMSIID (one byte SYSID) or MSGSRSID (two byte SYSID) is invalid. Determine if this SYSID has changed since the message was created. If it has changed, this is a procedure error. Otherwise, this error is probably an IMS internal error.
Reg15=00000009		The source SYSID at PSTSIDS, taken from the message at MSGMSIID (one byte SYSID) or MSGSRSID (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, this is a procedure error. Otherwise, this error is probably an IMS internal error.

Key	Label	Description
Reg15=0000000A		The message source SYSID at MSGMSIID (one byte SYSID) or MSGSRSID (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 a MSNAME type, but the source name at PSTIDSTN/MSGIDSTN is an APPC/OTMA token name. This 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 should only match for local messages. This 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 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 this is a valid MSC MSNAME for this IMS. If not, this MSNAME might have been removed while the message was on the SQ to be processed. Otherwise, this 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 a LTERM or TRANCODE. If this is not the problem, this is probably an IMS internal error.
Reg15=000000010		A FINDDEST call to locate the source MSNAME at PSTIDSTN/MSGIDSTN was not successful. This is a non shared queues IMS. Verify that the MSNAME was not removed from the system definition after the message was created, and IMS was started with warm start and the message was left on the queue. Changing MSNAMES requires at least a CTLBLKS sysgen and a cold start to remove messages from the queue. Otherwise, this is probably an IMS internal error.

Key	Label	Description
Reg15=000000011		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 a LTERM or TRANCODE. If not, this 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 is an MSC message that 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 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 is an MSC message that 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 the name is still a valid LTERM and not a TRANCODE or MSNAME. If not, this is probably an IMS internal error.
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 a LTERM or TRANCODE. If not, this 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 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 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 is an IMS internal error.

Key	Label	Description
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 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 a LTERM or TRANCODE. If not, this is probably an IMS internal error.

Related Reading: For more information about MSC and IMSplex shared queues, and particularly about avoiding pseudoabend ABENDU0830, see *IMS Version 9: Administration Guide: Transaction Manager*.

ABENDU0832

DFSRCHB0, DFSFRSP0

Explanation

An incorrect free space element (FSE) was found or created in the record while requesting or freeing space in a PHDAM, PHIDAM, HDAM, or HIDAM database. 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

ABENDU0832 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' will be 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 will contain the register 14 return address of the caller of space management. The caller will be backout, delete or replace, or load or insert.

Field PSTDATA will point to the buffer containing the block or CI currently being used.

DFSRCHB0

Analysis

DFSRCHB0 is entered to search through the FSE chain and locate a block of free space which will make 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) space enough for the segment plus a minimum length segment, and 3) space for two segments. For DFSRCHB0, register 2 will point to the start of the block.

Key	Label	Description
Reg2=A(BLOCK) Reg3>Reg11	CTNUSRCH	If free space pointer will shows a position which 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 will exceed 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.

DFSFRSP0

Analysis

DFSFRSP0 will be called due to a request to free a segment. The segment will be 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¬=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¬=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.

ABENDU0833

DFSDLA30

Explanation
An invalid System-ID (SYSID) was encountered.

Analysis
ABENDU0833 is a standard abend issued by DFSDLA30. The program status word (PSW) at entry-to-abend should be used to isolate the failing module.

Possible Cause
Internal logic error or data areas have been destroyed.

DFSDLA30

Analysis
ABENDU0833 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 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.

ABENDU0834

DFSGGSP0

Explanation
While processing calls from the HD space management module, the buffer handler encountered I/O errors reading the data set.

Analysis
ABENDU0834 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.

ABENDU0835

DFSDXMT0

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
ABENDU0835 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.

ABENDU0840

DFSDLR00, DFSDXMT0, DFSDDLE0, DFSDLDC0, DFSDCAP0, DFSPRIMS, DBFCMP10

Explanation
A user-written Segment Edit/Compression Routine has 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 using the file select and formatting print program, DFSERA10. The OPTION PRINT statement must specify EXITR=DFSERA30.

PSTLOGWA PSTLOGWA contains diagnostic information related to the processing error.

Offset	*Item*
X'00'	The label name or error reason 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:

word 1
Does not contain valid information.

word 2
Address of source segment. This is an input field.

word 3
Address of destination segment. This is an output field.

word 4
PSDB address, or zero if Fast Path.

word 5
Address of DMBCPAC or DBFCMPC if Fast Path.

word 6
Entry Code.

X'20' A copy of the DMBCPAC or DBFCMPC if Fast Path.

DFSDLR00

Explanation
ABENDU0840 is a pseudoabend issued from module DFSDLR00 at label COMPCALL. For analysis, see **SYSTEM LOG** and **PSTLOGWA** above.

DFSDXMT0

Explanation
ABENDU0840 is a pseudoabend issued from module DFSDXMT0 at label NOKEY. For analysis, see **SYSTEM LOG** and **PSTLOGWA** above.

DFSDDLE0

Explanation
ABENDU0840 is a pseudoabend issued from module DFSDDLE0 at label NDXREP3, NOASRT2, or CKKEYLP. For analysis, see **SYSTEM LOG** and **PSTLOGWA** above.

DFSDLDC0

Explanation
ABENDU0840 is a pseudoabend issued from module DFSDLDC0 in the EXPAND routine. For analysis, see **SYSTEM LOG** and **PSTLOGWA** above.

DFSDCAP0

Explanation
ABENDU0840 is a pseudoabend issued from module DFSDCAP0 at routine CALL_PSEUDO_ABEND. For analysis, see **SYSTEM LOG** and **PSTLOGWA** above.

DFSPRIMS

Explanation
ABENDU0840 is a pseudoabend issued from module DFSPRIMS at routine IMS0B. Register 14 contains the address of **PSTLOGWA**.

DBFCMP10

Explanation
ABENDU0840 is a pseudoabend issued from module DBFCMP10 at routine CALLEXIT. At abend, register 15 contains the label name or error reason returned by the Segment Edit/Compression routine. Register 14 contains the abend code returned from the routine.

ABENDU0842

DFSDBH40, DFSDVBH0

Explanation
An error occurred while attempting to do a pseudo data set extend to an OSAM database in a data sharing environment.

DFSDBH40

Analysis
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 DSNAME of the data set with the error. Refer to message DFS842I in *IMS Version 9: Messages and Codes, Volume 2* for the definitions of the reason codes.

DFSDVBH0

Analysis
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 ABENDU0842 and changes it to a standard abend.

ABENDU0843

DFSDLR00

Explanation
The block number passed to DL/I from the user's randomizing routine, when multiplied by the block size, yields a value that exceeds the maximum addressable data address. The data address must be addressable by 32 bits in VSAM or 31 bits in OSAM.

Analysis
This is a pseudoabend issued by module DFSDLR00.

ABENDU0844

DBFIRC10, DFSDVBH0, DFSDVSM0, DFSGGSP0

Explanation
No space was available in the data set or an error occurred while extending the data set in a data sharing environment.

The following defines the modules that issues this abend:

DBFIRC10	No available space in the DEDB area.
DFSDVSM0, DFSGGSP0	Insufficient space in the database to insert new blocks.
DFSDVBH0	A non-Fast Path failure that has an options statement with DUMP=YES statement.

DBFIRC10

Analysis
This is a pseudoabend 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 EPSTSPRC field.

DFSDVSM0 or DFSGGSP0

Analysis
This is a pseudoabend issued when a data set is full. Message DFS844I is issued to the master terminal operator and z/OS console indicating the data set and database names of the data set having the error.

DFSDVBH0

Analysis
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 ABENDU0844 and changes it to a standard abend.

ABENDU0845

DFSDVSM0, DFSFXC50, DFSNOTB0

Explanation
An unexpected condition occurred in DFSNOTB0, DFSDVSM0, or DFSFXC50. This is an IMS system error.

DFSDVSM0

Analysis
ABENDU0845 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'	AB0845A	A write error BUFC was found, but no Write Error Queue Element exists.
Reg1=X'034D'	AB0845B	A nonzero return code was received from NOTIFY during insert processing.
Reg1=X'034D'	AB0845C	A nonzero return code was received from VSI update during insert ESDS processing.
Reg1=X'034D'	AB0845D	A nonzero return code was received from NOTIFY during insert ESDS processing.
Reg1=X'034D'	AB0845E	A nonzero return code was received from NOTIFY during ESDS PUT new LR.
Reg1=X'034D' Reg8=RPL Block	AB0845F	The RPL passed to the JRNAD exit routine is not an IMS RPL.
Reg1=X'034D'	AB0845G	A nonzero return code was received from NOTIFY during write error processing.
Reg1=X'034D'	AB0845H	No Write Error Queue Element was returned. Excessive write errors, which use all the WEQEs in the pool, could cause this problem.
Reg1=X'034D'	AB0845I	A write error was detected in every buffer in the subpool.
Reg1=X'034D' Reg2=buffer address Reg3=CI size Reg4=RBA of CI	AB0845J	Either an invalid RDF, CIDF, or both, was found in the buffer that was to be written. Message DFS0442A is issued before this abend.
Reg1=X'034D' Reg8=RPL Block	AB0845K	The RPL passed to the UPAD exit routine is not an IMS RPL.
Reg1=X'034D'	AB0845L	An RPL was already active in VSAM on entry to module DFSDVSM0.
Reg1=X'034D' Reg2=buffer address Reg4=RBA of CI	AB0845M	Either an invalid RDF, CIDF, or both, was found in the buffer in the VSAM buffer pool during shared-read processing.
Reg1=X'034D'	AB0845N	Unable to acquire space for a READ JFCB macro.
Reg1=X'034D'	AB0845O	A nonzero return code was received from a READ JFCB macro.
Reg1=X'034D'	AB0845P	A nonzero return code was received from QUIT READ.

DFSFXC50

Explanation

At sync point time, after the buffer pools have been purged, a buffer was found that was not written back to the database.

Analysis

ABENDU0845 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 will resolve the problem.

Possible Cause

The buffer was not written or was written before the BQEL was created.

DFSNOTB0

Analysis

ABENDU0845 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.

ABENDU0846

DFSDBH00

Explanation
Module DFSDBH00 was posted with an unexpected post code. This 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.

ABENDU0847

DFSDBH00

Explanation
An unexpected condition occurred in the OSAM buffer handler routine. This is an IMS system error.

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.

ABENDU0847 could also occur if an invalid local vector index was detected while testing for buffer validity using coupling facility services.

ABENDU0848

DFSDBH20

Explanation
All the buffer prefixes in the OSAM buffer subpool have been locked because of permanent WRITE I/O errors.

Analysis
ABENDU0848 is a pseudoabend issued by DFSDBH20. Message DFS0762I is printed for each buffer marked in error.

ABENDU0849

DFSDLR00

Explanation
Retrieve attempted to issue a pseudoabend for a DBPCB that had PROCOPT=GO. The abend code was changed to U0849.

Analysis
ABENDU0849 is a pseudoabend issued from module DFSDLR00. Prior to changing the abend code to U0849, registers 14 through 12 from the original abend were saved starting at X'C' in the last save area in the PST. The save area starts at label PSTSAVL. Register 14 is indicated by a X'AA' in the 1st byte. The hexadecimal value of the original abend is in the two low order bytes of REG0, which was saved at PSTSAVL+X'14'. Once the original abend code is found, the FAST diagram for that abend code should be referenced.

ABENDU0850

DFSDLR00

Explanation
Retrieves request to the buffer handler for a key type call or a byte locate call failed.

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.

Analysis
ABENDU0850 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.

ABENDU0851

DFSDLR00

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.

Analysis
ABENDU0851 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.

ABENDU0852

DFSDLR00

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.

Analysis

ABENDU0852 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:

- 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'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'.)

Key	Label	Description
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.

ABENDU0853

DFSDLR00

Explanation

For an HDAM, HIDAM, PHDAM, or PHIDAM database, the segment returned on a buffer handler byte locate call for a root segment either did not have a segment code of 1 or (for HIDAM or PHIDAM) had a key value that did not match the key value in the index pointer segment.

Analysis

ABENDU0853 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.

ABENDU0854

DFSDLR00

Explanation
Retrieve's HSAM service routine was looking for the next root in the current block and encountered an invalid segment code.

Analysis
ABENDU0854 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' DSGNOSAM=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.

ABENDU0855

DFSLRH00, DBFLRH00, DFSDLA00

Explanation
Pseudoabends DFSLRH00 and DBFLRH00 are issued when the lock request handler detects an error while processing a lock request.

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'.

DFSLRH00, DBFLRH00

Analysis
ABENDU0855 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.
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.

Key	Label	Description
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.

DFSDLA00

Analysis

Prior to restarting IMS and rerunning the transaction, ensure that the exit is link-edited 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'.

ABENDU0857

DFSDLR00

Explanation

Verification of position for insertion of an HDAM, HIDAM, PHDAM, or PHIDAM segment failed to locate a valid chain.

Analysis

ABENDU0857 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.

ABENDU0858

DFSDLR00

Explanation
The pair of a physically paired segment could not be located.

Analysis
ABENDU0858 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.

ABENDU0859

DFSDLR00

Explanation
IMS found a logical child on a GET call and the logical parent could not be found.

Analysis
This 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.

ABENDU0860

DFSDDLE0

Explanation

A return error from a call to the database handler occurred.

ABENDU0860 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 ABENDU0860, you will need the pseudoabend SNAP dump of control blocks. Use the SCD to get the entry points for DFSDDLE0 (SCDDLI07), space management (SCDDHDS0), the buffer handler (SCDDBH0), and DFSDLD00 (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.

Name	***Description***
PSTFNCTN	requested function
PSTBYTNM	requested RRN or RBN
PSTRTCDE	return code
PSTDSGA	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 DFSDLD00. 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.

Key	Label	Description
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 DMBVLDFG 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.
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.

Key	Label	Description
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.
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'	SEGTONEW	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.
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.

Key	Label	Description
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.

ABENDU0861

DFSDDLE0

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 in determining the length of a segment to the right of the insert point.

Analysis

ABENDU0861 is issued from DFSDDLE0, at label ABEND861. A SNAP of the control blocks issued to the log is needed to diagnose the cause of ABENDU0861.

Search down through the save areas until you find the last save area. Register 12 equals the entry address of DFSDDLE0 (SCDDLI07). Use the registers in this register save area. Register 14 indicates the point in DFSDDLE0 from which this subroutine was entered. Register 10 points to the insert point or the first segment to be shifted. 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 'COMPSHFT'. 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 'COMPSHFT'. 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.

Key	Label	Description
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.

Possible Cause

- DBD was redone changing the length of segments without reloading the database.
- A bad insert position in the SDBs was passed from Retrieve (DFSDLR00) or Replace (DFSDLD00).

ABENDU0862

DFSDDLE0

Explanation

ABENDU0862 is issued from DFSDDLE0 on insert of a logical child/logical parent concatenated segment when logical parent insert rule equals virtual. The format of the user I/O area was correct, but when attempting to replace a logical parent, the key in the logical parent did not match the corresponding portion of the concatenated key of that logical parent in the logical child.

Analysis

ABENDU0862 is issued from DFSDDLE0, at label ABEND862. To diagnose this abend, you will 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.

ABENDU0863

DFSDDLE0

Explanation

ABENDU0863 is issued from DFSDDLE0 because an error in the user's edit/compression routine resulted in one of the following:

- The segment length is greater than the defined maximum.
- For fixed-length segments, the length is greater than the defined length.
- The key was changed without key compression being specified.

Analysis
ABENDU0863 is detected by DFSDDLE0, at label VLDERR. To diagnose ABENDU0863 the pseudoabend SNAP of control blocks put out to the system log will be 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.

ABENDU0864

DFSDDLE0

Explanation
An unexpected return was received from the program isolation (PI) ENQ/ DEQ routine, or from an invalid post to IWAIT.

Analysis
ABENDU0864 is a pseudoabend which is not issued but the code that sets it is in DFSDDLE0 at label ABEND864. The code remains in DFSDDLE0 for possible diagnostic efforts.

ABENDU0865

DFSDDLE0

Explanation
When inserting a segment into a HISAM database, a work area was required to hold one LRECL. Space was not available using the ICREATE macro.

Analysis
ABENDU0865 is a pseudoabend. To diagnose an ABENDU0865, you will need the pseudoabend SNAP of control blocks that was written to the log data set. You will need the entry point of DFSDDLE0. (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 DFSDDLE0. In the *next lower* save area, register 12 should contain the entry point of DFSDDLE0 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' will point 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 ABENDU0865.

Key	Label	Description
Reg14=BAL	GETPOOL	On return from DFSICRET (ICREATE) when ICREATE was unable to get space, ABENDU0865 is issued. Register 2 will contain the 4-byte name of the POOL. Register 3 will contain the POOL size requested in bytes.

ABENDU0867

DFSDDLE0

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
This is a pseudoabend issued by module DFSDDLE0.

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.

ABENDU0868

DFSDDLE0

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 keyed in the index record.

Analysis
ABENDU0868 is issued from DFSDDLE0 at label ABEND0868. To diagnose this abend occurring in an online region, you will 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:

Code	*Meaning*
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. ABENDU0868 denotes that X'3C' was zeros.

ABENDU0869

DFSDVBH0

Explanation
The DL/I buffer-handler router was asked to perform an undefined function. This is an IMS system error.

Analysis
This is a standard abend issued by module DFSDVBH0. The pertinent PST field is PSTFNCTN.

ABENDU0870

DFSDLR00

Explanation
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
ABENDU0870 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. The ABENDU0870 is issued when VSAM returns the same next higher key a second time.

Possible Cause
- Broken index (that is, missing index entries)
- HIDAM roots previously inserted out of sequence

ABENDU0880

DFSDLR00, DFSDDLE0, DFSDXMT0, DFSDLDC0

Explanation
Internal logic errors encountered during DL/I processing of a HALDB result in pseudoabend ABENDU0880.

Analysis
ABENDU0880 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.

RC	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'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 invoked.
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.

| **X'C077'** A HALDB OLR cursor refresh failed.

| For more information about HALDB Online Reorganization, see *IMS Version 9: Administration Guide:*
| *Database Manager*.

ABENDU0885

DFSDLA00

Explanation
Pseudoabend issued by DFSDLA00 when a required user exit (DFSDBUX1) could not be loaded or DATXEXIT=YES was specified in the DBDGEN, but the user exit set SRCHFLAG to x'FF'.

Analysis
The programmer must ensure that the exit is link-edited 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'.

ABENDU0888

DFSIINF0, DFSIING0

Explanation
ABENDU0888 is issued when the active format library does not have any members and the IMS system has MFS-supported terminals.

Analysis
ABENDU0888 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.

ABENDU0889

DFSIINF0, DFSIING0

Explanation
A DEVTYPE macro was issued with an invalid area address.

Analysis
ABENDU0889 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 will point 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 abending.

Register 11 in the abend SVRB registers is the base register for this module. Register 15 will contain 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' Reg14=BALR Reg15=return code X'08'	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 will cause a branch to label ABEND1 for ABENDU0889.

ABENDU0890

DFSIINF0, DFSIING0

Explanation
A DEVTYPE macro was issued for ddname FORMATA or FORMATB, which was in the active format library and a not-found condition occurred.

Analysis
The active format library ddname is identified in the message text of DFS3410I.

ABENDU0890 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 will point 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 abending.

Register 11 in the abend SVRB registers is the base register for this module. Register 15 will contain 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 ABENDU0890.

Possible Cause
A DD statement defining the active format data set with ddname FORMATA or FORMATB was omitted from the job stream.

ABENDU0891

DFSIINF0, DFSTMEI0

Explanation
During IMS initialization, the active format data set could not be opened.

Analysis
ABENDU0891 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 will isolate 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'.

DFSTMEI0

ABENDU0891 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.

DFSIINF0

ABENDU0891 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 will point to the DCB for which open failed, and message DFS891A will be 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.

ABENDU0892

DFSIINF0, DFSIING0

Explanation

Insufficient storage was available in the message format block pool (MFBP) to complete the initialization of the pool.

Analysis

ABENDU0892 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 will point 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 prior to 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 will have been overlaid by the processing abend.

Key	Label	Description
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 will have been overlaid by the processing abend.
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. Note that at the time of abend, the contents of register 2 and register 3 will have been overlaid by the processing abend.

Possible Cause
Check the FBP parameter on the EXEC statement in the IMS procedure. It may be necessary to define a larger pool (the pool size for FBP is defined in 1KB blocks). Check the FBPR parameter on the EXEC statement in the IMS procedure.

ABENDU0893

DFSIINF0, DFSIING0

Explanation
An I/O error occurred issuing a POINT macro for the active format data set during initialization of the message format block pool (MFBP).

Analysis
ABENDU0893 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 will point 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 DFS983 to the IMS master console prior to abending.

Register 11 in the abend SVRB registers is the base register for this module. Register 3 will contain the abend completion code, X'8000037D'.

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, 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.
Reg8=BAL for READ routine Reg14=0 R1ERRBIT=X'20'	HIT	While reading the $$IMSDIR directory block, register 14 should contain the address of the resident directory. Register 14 is tested and found to be zero, indicating an I/O error occurred during the reading of the $$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.

Key	Label	Description
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 Reg14=0 R1ERRBIT=X'20'	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.

ABENDU0894

DFSIINF0, DFSIING0

Explanation
An I/O error occurred during issuing of 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
ABENDU0894 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 will point 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 will contain 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.

Key	Label	Description
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.

ABENDU0895

DFSIING0

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 IMS format library.

Analysis
Rerun the job when more space is available in the IMS extended private area.

ABENDU0896

DFSIINF0, DFSIING0

Explanation
The DD DUMMY parameter is not supported for the active FORMAT data set. The ddname is FORMATA or FORMATB. Refer to message DFS3410I, which identifies which of the ddnames is the active format data set.

Analysis
ABENDU0896 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 will point 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 ABENDU0896 will not be 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).

ABENDU0897

DFSIING0

Explanation
Insufficient storage was available in the IMS private area to allocate the staging buffers used to read MFS control blocks from the active IMS format library.

Analysis
Rerun the job when more space is available in the IMS private area.

ABENDU0898

DFSIING0

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 IMS format library.

Analysis
Rerun the job when more space is available in the IMS extended private area.

ABENDU0899

DFSIING0

Explanation
Insufficient storage was available in the IMS private area to allocate the MFS buffer pool control area.

Analysis
Rerun the job when more space is available in the IMS private area.

ABENDU0901

DFSIMP00

Explanation
An error exists in the Security Maintenance utility program parameter field.

Analysis
ABENDU0901 is a standard abend issued from module DFSIMP00. Entry registers to DFSIMP00 are stored (register 14 through register 12) at register 13 plus X'C'.

The error is detected in the initialization routine, SMPINIT, where parameter options are checked. The abend is issued because of an error in the parameter field. The parameter information is passed in register 1. Use register 1 from the save area set for problem diagnosis.

Register 12 in the abend SVRB is the base register from DFSIMP00.

The abend is issued from a common routine at label SMPABND. When an error is detected, a branch is taken to label SMPI1, which loads register 1 with the abend code and branches to SMPABND to issue the abend (SVC 13).

Key	Label	Description
Reg1 + 4¬=C'LIST'	SMPINIT	An error was encountered while checking parameter options. The LIST option was not specified, so the abend is issued.
Reg1 + 2=C'UPDATE' Reg1 + 0¬=X'8'	SMPI2	It has been determined that default is set for the update option, but register 1 + X'0' does not equal X'8', so the abend was issued.
Reg1 + 2=C'UPDATE' Reg1 + 8¬=C','	SMPI2	When checking parameter options, if a comma is missing at register 1 + '8', the parm information is invalid and the abend is issued.
Reg1 + 2=C'LIST'	SMPI3	Option setting is complete, but register 1 + X'0' does not equal X'6', so the abend is issued.
Reg1 + 2=C'LIST' Reg1 + 0=X'6' Reg1 + 6¬=C','	SMPI3	A comma is missing, so the parameter information is incorrect and the abend is issued.

ABENDU0902

DFSIMP00

Explanation
JOBLIB does not contain the DFSISDBx or DFSISDCx module specified by the parameter field.

Analysis
ABENDU0902 is a standard abend detected in module DFSIMP00. The program status word (PSW) at entry-to-abend and the registers in the abend SVRB can be used in problem isolation.

ABENDU0902 is issued from the initialization routine, SMPINIT, where parameter options are checked. The abend is issued because of a BLDL failure for the SDB. Register 15 contains the return code passed by BLDL:

Code ***Meaning***

X'00' Successful completion

X'04' One or more entries in the list could not be filled; the list supplied may be invalid. If a search attempted by the entry is not found, the R field (byte 11) for that entry is set to zero.

X'08' A permanent I/O error was detected when the system attempted to search the directory.

X'0C' The number of command entries exceeded the maximum allowed in the Command Verb Symbolic list (CVB).

Register 12 is the base register for DFSIMP00. Register 0 contains the BLDL list address (SMPLIST).

The abend is issued from a common routine at label SMPABND. SMP14 is the BLDL routine. When an error is detected, it branches to SMPABND.

Key	Label	Description
Reg0=SMPLIST Reg15¬=0	SMPINIT SMPI4	No parameters were passed to the PARM initialization routine, so the default values are used. This routine is doing a BLDL to see if the desired SDB (DFSISDBx or DFSISDCx) is in the library. A nonzero return code after the SVC 18 indicates that a BLDL failure occurred and the abend is issued.
Reg0=SMPLIST Reg15¬=0	SMPI2 SMPI4	The default value is set for update. More validity checking is done and the BLDL is issued to see if the desired SDB (DFSISDBx or DFSISDCx) is in the library. A nonzero return code after the SVC 18 indicates that a BLDL failure occurred and the abend is issued.

Possible Cause
The BLDL list supplied may be invalid.

ABENDU0903

DFSIMP00, DFSIMP20

Explanation

DD statements required by the Security Maintenance utility program are missing.

DFSIMP00

Analysis

ABENDU0903 is a standard abend issued from module DFSIMP00 or DFSIMP20. The program status word (PSW) at entry-to-abend identifies which module issued the abend.

In DFSIMP00, the abend is issued because of a failure to successfully open the DCB for one of the following four data sets: SYSPRINT, SYSPUNCH, SYSIN, or SYSLIN.

The register 14 BAL to routine CRDCB can be used to isolate a particular label. Register 12 is the base register for the module.

The register 14 BAL points back to the routine which requested that a DCB be created and opened. Each routine saves its caller's registers (register 14 through register 12) at register 13 + X'C'. Beginning with register 13 in the abend SVRB and tracing through these save area sets, the flow of control that led up to the error condition can be traced.

The abend is issued from a common routine at label SMPABND. CRDCB is the routine that creates and opens the DCBs. When an OPEN error is detected, it branches to SMPABND.

Key	Label	Description
Reg3=DCB addr DCBOFLGS¬=X'10' Reg14=BAL	SMPRT (CRDCB)	This routine was BALed to from SMPSNFG to print the current system configuration report. The DCBOFLGS field of the SYSPRINT DCB is tested for a successful OPEN. If an OPEN error occurs, the abend is issued.
Reg3=DCB addr DCBOFLGS¬=X'10' Reg14=BAL	SMSIN1 (CRDCB)	This routine was BALed to from P1MORE to read the SYSIN input. The DCBOFLGS field of the SYSIN DCB is tested for a successful OPEN. If an OPEN error occurs, the abend is issued.
Reg3=DCB addr DCBOFLGS¬=X'10' Reg14=BAL	P1END (CRDCB)	After the SYSIN control statements have been read in and edited, a branch is taken to this routine to punch out the SYSIN input. The DCBOFLGS field of the SYSPUNCH DCB is tested for a successful OPEN. If an OPEN error occurs, the abend is issued.
Reg3=DCB addr DCBOFLGS¬=X'10' Reg14=BAL	ENDPASS2 (CRDCB)	The SYSLIN DCB is to be opened prior to generating the Link Edit control statements. The DCBOFLGS field is tested for a successful OPEN. If an OPEN error occurs, the abend is issued.

DFSIMP20

Analysis

The abend is issued from this module because of a failure to successfully open either the SYSUT1 or SYSUT2 data set.

DFSIMP00 calls DFSIMP20 to open the SYSUT data sets during various stages of its processing. The register 14 BAL can be used to determine which routine called DFSIMP20 and to determine which data set it was trying to open.

The entry registers (register 14 through register 2) are stored at register 13 + X'C'.

The abend is issued from a common routine at label OSYS5. Register 12 is the base register.

Key	Label	Description
Reg14=BAL SYSUT1+OFLGS¬=X'10'	OSYS (OSYS3)	DFSIMP00 (routine SMPPH1) has called DFSIMP20 to open the SYSUT1 work file. SYSUT1+OFLGS is tested for a successful OPEN. If an OPEN error occurs, the abend is issued.
Reg14=BAL SYSUT2+OFLGS_X'10' or SYSUT1+OFLGS¬=X'10'	OSYS (OSYS3)	DFSIMP00 (routine PASS2) has called DFSIMP20 to open the SYSUT data sets. SYSUT2 + OFLGS is tested for a successful OPEN. If an OPEN error occurs, the abend is issued. If SYSUT2 has been opened successfully, then SYSUT1 + OFLGS is tested for a successful OPEN. If an OPEN error occurs, the abend is issued.
Reg14=BAL SYSUT2+OFLGS¬=X'10'	OSYS (OSYS2)	DFSIMP00 (routine PASS3) has called DFSIMP20 to open the SYSUT2 work file. SYSUT2 + OFLGS is tested for a successful OPEN. If an OPEN error occurs, the abend is issued.

ABENDU0904

DFSICV50, DFSIINF0, DFSFDIR0

Explanation

While processing the PDS directory entries, an entry for a format block larger than 32767 bytes was detected.

Analysis

ABENDU0904 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.

ABENDU0905

DFSDLBL0

Explanation

The DL/I batch block builder was unable to obtain sufficient storage to build the required control blocks.

Analysis

ABENDU0905 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS905I will accompany 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.

Code	***Meaning***
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 DFS905I 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 BLDLIMT2	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.

Key	Label	Description
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.
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.

ABENDU0906

DFSDLBL0

Explanation
A SENSEG statement had an invalid processing option specified.

Analysis
ABENDU0906 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS906I will accompany 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 DFS906I 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.

ABENDU0907

DFSDLBL0

Explanation
A PSB had a PCB which referenced a logical DBD and had a PROCOPT of L or LS.

Analysis
ABENDU0907 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS907I will accompany 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 DFS907I 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.

ABENDU0909

DFSDLBL0, DFSDPSB0

Explanation
The DBD name is not a valid DBD.

Analysis
ABENDU0909 is a standard abend issued from multiple CSECT modules DFSDLBL0 and DFSDPSB0.

Message DFS909I will accompany this abend.

DFSDLBL0

Explanation
The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 6 in the abend SVRB point to the DBD.

Analysis
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 DFS909I 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.

DFSDPSB0

Explanation
This module is called by DFSDLBL0 if the region parameter is UDR or ULU. A parameter list address is passed in register 1.

Analysis
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 ABENDU0909 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.

ABENDU0910

DFSDLBL0

Explanation
An internal programming error has occurred while processing the PSB.

Analysis
ABENDU0910 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS910I will 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 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).

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.

ABENDU0911

DFSDLBL0

Explanation
The processing option intent list length was calculated incorrectly for the named PSB.

Analysis
ABENDU0911 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS911I will accompany 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.
Reg3=current limit table entry address Reg14=BAL	UPISRTB	Update intent will be set for the current PSDB and related PSDBs for 'I' 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 'I' 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.

ABENDU0912

DFSDLBL0

Explanation

The named PSB referenced the named SEGM in the named DBD. The named SEGM does not exist in the named DBD.

Analysis

ABENDU0912 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS912I will accompany 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.

ABENDU0913

DFSDLBL0

Explanation
The named DBD contains an invalid or unknown access method.

Analysis
ABENDU0913 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS913I will 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 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 DFS913I 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.

ABENDU0916

DFSDLBL0

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 nonunique.

Analysis
ABENDU0916 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS916I will 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.

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 DFS916I 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.
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.

ABENDU0917

DFSDLBL0

Explanation

The first DBD referenced the named SEGM in the second DBD. The SEGM does not exist in the second DBD.

Analysis

ABENDU0917 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS917I will accompany 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 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 DFS917I 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.

ABENDU0919

DFSDLBL0

Explanation

The named PCB in the named PSB contains a KEYLEN parameter that is too small to hold the longest fully concatenated key. The correct length is indicated.

Analysis

ABENDU0919 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS919I will accompany 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 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 DFS919I 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.

ABENDU0920

DFSDLBL0

Explanation

The named PSB contains 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 may be direct, such as in a PCB statement, or the reference may be indirect, such as in a DBD that references another DBD using logical or index relationships.

Analysis

ABENDU0920 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS920I will accompany 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 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 DFS920I 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.

ABENDU0921

DFSDLBL0

Explanation
The named PSB was loaded. Upon examination, it was discovered that the PSB was not a valid PSB.

Analysis
ABENDU0921 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS921I will 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.

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 DFS921I 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.

ABENDU0922

DFSDLBL0

Explanation
The named DBD was loaded. On examination, it was discovered that the DBD was not valid. On HD unload and reload, the abend was issued when no DBD name was given. In a database recovery region, no DBD name was given.

Analysis
ABENDU0922 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS922I will 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.

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 DFS922I 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.

ABENDU0923

DFSDLBL0

Explanation

The indicated logical child segment in the indicated DBD had a BYTES specification shorter than its logical parent's fully concatenated key. The minimum acceptable length is indicated.

Analysis

ABENDU0923 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS923I will accompany this abend.

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 DFS923I 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 DMBSECSC=segment 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.

ABENDU0924

DFSDLBL0

Explanation

The root segment in the named INDEX DBD had a data length that was too small to hold the required index data. The minimum acceptable length is indicated.

Analysis
ABENDU0924 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS924I will accompany this abend.

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 DFS924I 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.

ABENDU0925

DFSDLBL0

Explanation
A named 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.

Analysis
ABENDU0925 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS925I will accompany 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 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 DFS925I 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.

ABENDU0926

DFSDLBL0

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.

Analysis
ABENDU0926 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS926I will accompany 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 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 DFS926I 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.

ABENDU0927

DFSDLBL0

Explanation
The named INDEX DBD specified an indexed field in the INDEXED DBD which did not exist.

Analysis
ABENDU0927 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS927I will accompany this abend.

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 DFS927I 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.

ABENDU0928

DFSDLBL0

Explanation
The named INDEX DBD indexes a valid field, but the field is not a sequence field.

Analysis
ABENDU0928 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS928I will 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 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 ERROR928. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR928. A branch is then taken to RETURN, where message DFS928I 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.

ABENDU0929

DFSDLBL0

Explanation
A BLDL was issued for the named member. The member was not found in the DBD or PSB library.

Analysis
ABENDU0929 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS929I will 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 member name.

ABENDU0929 is issued because of a BLDL failure for the PSB or DBD. Register 15 contains the return code passed by BLDL:

Code ***Meaning***

X'00' Successful completion

X'04' One or more entries in the list could not be filled; the list supplied may be invalid. If a search is attempted but the entry is not found, the R field (byte 11) for that entry is set to zero.

X'08' A permanent I/O error was detected when the system attempted to search the directory.

ABENDU0929 is also issued if the linkage editor has not created the member, that is, the named member does not exist in load module form in the PDS. In this instance, the BLDL return code will be X'00'.

The contents of register 15 should be used to determine the reason this abend was issued for the named PSB or DBD.

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 DFS929I 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 linkage editor 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 linkage editor has not created the DBD, so the abend is issued.

ABENDU0930

DFSDLBL0

Explanation

The named 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.

Analysis

ABENDU0930 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS930I will accompany this abend.

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 DFS930I 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.

ABENDU0931

DFSDLBL0

Explanation

The named 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.

Analysis

ABENDU0931 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS931I will accompany this abend.

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 DFS931I 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.

ABENDU0932

DFSDLBL0

Explanation
The named INDEX DBD does not have a sequence field defined for the index segment.

Analysis
ABENDU0932 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS932I will 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 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 DFS932I 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.

ABENDU0933

DFSDLBL0

Explanation
The indicated PSB contained an INDICES operand which was invalid. The indicated value of the INDICES operand was not a valid index name for the associated SENSEG statement.

Analysis
ABENDU0933 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS933I will accompany 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 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.

ABENDU0934

DFSDLBL0

Explanation

The named PSB referenced the name logical child SEG, in the named DBD. The logical child requires the logical parent's concatenated key to be stored physically, but VIRTUAL was specified.

Analysis

ABENDU0934 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS934I will accompany this abend.

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 DFS934I 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.

ABENDU0935

DFSDLBL0

Explanation

The named PSB referenced the named DBD through a SENSEG statement. However, a logical structure or relationship within this segment definition is invalid.

Analysis

ABENDU0935 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS935I will accompany this abend.

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 DFS935I 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.
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.

ABENDU0936

DFSDLBL0

Explanation

The name 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.

Analysis

ABENDU0936 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 (SDBFB3) is for a LOAD. A virtual segment can not be loaded, so the abend is issued.

ABENDU0937

DFSDLBL0

Explanation
The named DBD specified the named as a virtual segment. The SOURCE operand in the SEGM statement contained more than one SOURCE operand.

Analysis
ABENDU0937 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS937I will accompany 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.

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 DFS937I 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.

ABENDU0938

DFSDLBL0

Explanation
The named database PCB within the named PSB has no SENSEG statements defined at PSBGEN time. The PSBGEN was invalid.

Analysis
ABENDU0938 is a standard abend issued from multiple CSECT module DFSDLBL0.

Messages DFS945I and DFS0938I will accompany 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 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.

ABENDU0939

DFSDLBL0

Explanation

The indicated INDEX DBD contained an INDEX=operand which specified a field name with a /CK as the first 3 characters.

Analysis

ABENDU0939 is a standard abend issued from multiple CSECT module DFSDLBL0.

Messages DFS946I and DFS0939I will 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 DFS946I 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.

ABENDU0941

DFSDLBL0

Explanation

The indicated segment in the indicated 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

ABENDU0941 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS948I will accompany this abend.

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 DMBVLDFG=X'C' Reg15=physical pair's PSDB address DMBVLDFG¬=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 (DMBVLDFG=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. (DMBVLDFG=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 DMBVLDFG¬=X'04'	CLRDDIR6	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 DMBVLDFG¬=X'04'	CLRDDIRA	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.

ABENDU0942

DFSDLBL0

Explanation

The indicated INDEX DBD contained an LCHILD statement which contained the same database name for the indexed database as the name for this index database.

Analysis

ABENDU0942 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS949I will 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 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 DFS949I 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.

ABENDU0943

DFSDLBL0

Explanation

An invalid logical relationship exists. One of the following conditions is 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

ABENDU0943 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS950I will accompany 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 DFS950I 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¬=egment 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.

ABENDU0944

DFSDMSG0, DFSUMSG0

Explanation
Message ID not found in the message text module DFSUMGT0.

Analysis
ABENDU0944 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.

ABENDU0945

DFSDMSG0, DFSUMSG0

Explanation
The message formatter expected inserts to be passed; however, none were specified by the UERR macro.

Analysis
ABENDU0945 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 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.

ABENDU0947

DFSUAMB0

Explanation

An invalid secondary list code has been found in the DMB.

Analysis

ABENDU0947 is a standard abend issued from module DFSUAMB0.

Messages DFS0570I and DFS0951I will accompany this abend.

The registers in the abend SVRB should be used 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.

ABENDU0948

DFSUACB0

Explanation
The SYSPRINT data set had a permanent I/O error.

Analysis
ABENDU0948 is a standard abend issued from module DFSUACB0.

This abend 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.

Note: 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.

ABENDU0949

DFSUACB0

Explanation
The SYSPRINT data set could not be opened.

Analysis
ABENDU0949 is a standard abend issued from module DFSUACB0.

The registers in the abend SVRB should be used for problem isolation.

The program status word (PSW) at entry-to-abend will point 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.

ABENDU0950

DFSDSEH0

Explanation
The work data set generator 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
ABENDU0950 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 (CDSDCB). CDSDCB 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'04'C “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.

ABENDU0951

DFSDSEH0

Explanation
The work data set generator was performing a dynamic limit-check and the limit check failed.

Before the database reorganizes, it will have provided a diagnostic message during its execution that indicates segments for which limit-check failure occurred.

Analysis
ABENDU0951 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.

ABENDU0952

DFSDSEH0

Explanation
The work data set generator was attempting to open a work data set. It was not opened successfully. The work data set must be specified by a DD statement named DFSURWF1.

Analysis
ABENDU0952 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.

ABENDU0953

DFSDSEH0

Explanation
The work data set generator 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
ABENDU0953 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.

ABENDU0955

DFSURGP0, DFSURGS0, DFSURG10, DFSURPR0

Explanation
An abend control statement was supplied in the SYSIN input stream.

Analysis
This is a standard abend issued by module DFSURGP0, DFSURGS0, DFSURG10, or DFSURPR0.

ABENDU0956

DFSDSEH0

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
ABENDU0956 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 Reg4=CENT Reg8=PSDB Reg9=PCB Reg10=PST	LPLC #@LB62	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.

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).

ABENDU0957

DFSDLBL0

Explanation
The referenced segment is physically paired. Either the two intersection data lengths are not equal (for fixed-length segments), or the maximum intersection data lengths for the two paired segments are not equal.

Analysis
ABENDU0957 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS951I will accompany 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 DFS951I 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 Reg3=physical pair's PSDB Reg4=segment's data length Reg15=segment's DMB	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.

ABENDU0958

DFSDLBL0

Explanation
A PSB may not reference a primary index DBD for update.

Analysis
ABENDU0958 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS952I will accompany this abend.

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.

ABENDU0959

DFSDLBL0

Explanation

An error exists in the logical child SEGM statement in the referenced database. 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

ABENDU0959 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS953I will accompany 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 ERROR959, which saves register 0 through register 15 at label ERRORSVE. A branch is then taken to RETURN, where message DFS953I 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' DMBLORG=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.

ABENDU0960

DFSDLBL0

Explanation

An error exists in the LCHILD statement in the referenced data base. Either PTR=SNGL or PTR=DBLE was specified. Both specifications are incorrect. It is not permissible to specify a direct pointer to a database with HISAM organization.

Analysis

ABENDU0960 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS954I will accompany 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.

ABENDU0961

DFSDLBL0

Explanation

An error exists in the XDFLD statement in a DBD that points to the Shared Index. More than one XDFLD constant of the same value has been specified. Each XDFLD statement for a Shared Index must have a unique CONST=specification.

Analysis

ABENDU0961 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS955I will accompany 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 ERRORSAVE. A branch is then taken to RETURN, where message DFS955I 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.

ABENDU0962

DFSDLBL0

Explanation

The SENSEG statements of the named PSB were not specified in hierarchic sequence (top to bottom-left to right).

Analysis

ABENDU0962 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS956I will accompany 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 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 DFS956I 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.

ABENDU0963

DFSBACK0

Explanation

Backout was unable to open the checkpoint control data set (SYSIN), or the DEVTYPE macro failed.

Analysis

ABENDU0963 is a standard abend issued from module DFSBACK0.

The registers in the abend SVRB should be used for problem isolation. Register 12 is the base register.

The program status word (PSW) at entry-to-abend will point 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 will contain the return code from DEVTYPE.

Code ***Meaning***

X'04' ddname 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 CHKPDCB. If the OPEN was unsuccessful, DCBOFLGSSVRB should be used for problem diagnosisX'10', the abend is issued.

ABENDU0965

DFSURPR0

Explanation
There is an inconsistency in the DL/I control blocks involving a segment in a database that is being reorganized or initially loaded. Control blocks for a logically-related database are incomplete or in error. This may be an IMS system error, or an improperly defined logical relationship.

Analysis
This is a standard abend issued by module DFSURPR0.

ABENDU0966

DFSDLBL0

Explanation
The PCB contains a PROCOPT of H but the DBD is not a DEDB. The PROCOPT H is only valid for a DEDB.

Analysis
ABENDU0966 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS0964I will accompany 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 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 DFS0964I 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 HCHK1	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.

ABENDU0967

DFSDLBL0

Explanation
The PSB contains more than one explicit reference to a PHIDAM DBD with a PROCOPT of L or LS.

Analysis
ABENDU0967 is a standard abend issued from multiple CSECT module DFSDLBL0.

Message DFS1008I will accompany 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 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.

ABENDU0969

DFSURIO0

Explanation
On the 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
This abend is issued by module DFSURIO0 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 will point 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 DFSURI00	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.

ABENDU0970

DFSRBCP0

Explanation
One of several possible error conditions occurred while restart processing was attempting to determine the checkpoint to be used in restart.

Analysis
ABENDU0970 is a standard abend issued from module DFSRBCP0.

The program status word (PSW) at entry-to-abend can be used to determine which instruction issued the abend (SVC 13). The registers in the abend SVRB should be used for problem isolation. Register 10 contains the subcode.

This abend only occurs during restart.

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. **OPNx** 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. **RLVx** 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. **GIBx** An IOSB, used in the I/O for the restart data set, could not be obtained. This is a system error. **BCPx** An entry in the control block table could not be found for the checkpoint table, BCPT. This is a system error. **BCBx** A quick save area could not be obtained. This is a system error. **LOGx** 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. **CBTx** 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. **ALTx** 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. **QCSx** 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. **IDMX** 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.

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.

APAR Processing

Storage dump.

ABENDU0971

DFSRLP00, DFSRST00, DFSRBCP0

Explanation

An internal error occurred during restart processing.

Analysis

Register 15 at entry to abend contains one of the following codes:

Code ***Meaning***

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.

ABENDU0979

DFSRDBL0

Explanation

The database change logger, DFSRDBL0, encountered an error condition or a process failure and cannot complete its request processing.

Analysis

ABENDU0979 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.
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.

ABENDU0982

DFSRBOI0

Explanation
Emergency restart was unable to read the backout queue records.

Analysis
ABENDU0982 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 (DFSAOS30) and a return code is passed back in register 15.

The program status word (PSW) at entry-to-abend will point to the instruction within the routine labeled X100 from which the abend (SVC 13) is issued. Register 10 will contain the address of the database log record. Register 5 will contain the address of the current PST. Register 11 will contain the address of RSRDSECT. Register 12 will contain 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.

ABENDU0985

DFSRBLB0

Explanation
Emergency restart was unable to perform a backout.

Analysis
ABENDU0985 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 will contain a reason code with the following meanings:

Code	*Meaning*
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.

APAR Processing
Storage dump and log used for emergency restart.

ABENDU0986

DFSDLBL0

Explanation
A segment in a logical database has specified as its source a segment in another logical database.

Analysis
This is a standard abend issued by module DFSDLBL0.

Message DFS2425I will accompany 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.

ABENDU0987

DFSDLBL0

Explanation
An LCHILD statement in an index DBD has specified an incorrect segment name for the indexed segment.

Analysis
This is a standard abend issued by module DFSDLBL0.

Message DFS2426I will accompany 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 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.

ABENDU0988

DFSDLBL0

Explanation
An LCHILD statement in an indexed DBD refers to a nonexistent segment in the index DBD.

Analysis
This is a standard abend issued by module DFSDLBL0.

Message DFS2427I will accompany 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 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.

ABENDU0989

DFSDLBL0

Explanation

In the process of resolving index relationships it was found that one index DBD was referenced by multiple LCHILD statements from one or more indexed segments. Message DFS2428I contains the DBD names.

Analysis

This is a standard abend issued by module DFSDLBL0.

Message DFS2428I will accompany 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 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.

ABENDU0990

DFSDLBL0

Explanation
An error was detected 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.

Analysis
This is a standard abend issued from multiple-CSECT module DFSDLBL0.

Message DFS2429I will accompany this abend.

The registers saved at label ERRORSVE in CSECT DFSDLB80 within DFSDLBL0 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.

ABENDU0991

DFSDLB70

Explanation
A concatenation of logical child to destination parent was found to be invalid.

Analysis
This is a standard abend issued from module DFSDLB70.

Message DFS2430I will accompany this abend.

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 DMBFLG¬=DMBLPEX 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.

ABENDU0992

DFSDLBL0

Explanation

An indexed DBD was referenced invalidly, or a non-indexed DBD was referenced as indexed.

Analysis

This is a standard abend issued from multiple-CSECT module DFSDLBL0.

Message DFS2431I will accompany this abend.

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 below 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.

ABENDU0993

DFSDLBL0

Explanation

The field name specified in a SENFLD statement could not be found in the segment definition.

Analysis

ABENDU0993 is a standard abend issued from the composite module DFSDLBL0.

Message DFS2432I will accompany this abend.

The registers saved at label ERRORSVE in DFSDLB80 within DFSDLBL0 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.

ABENDU0994

DFSDLBL0

Explanation
Field mapping specified in PSBGEN causes destructive overlap.

Analysis
ABENDU0994 is a standard abend issued from the composite module DFSDLBL0.

Message DFS2433I will accompany this abend.

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.

ABENDU0995

DFSDLBL0

Explanation
Insert sensitivity was specified for the segment, but sensitivity to the key field was not specified.

Analysis
ABENDU0995 is a standard abend from the composite module DFSDLBL0.

Message DFS2434I will accompany this abend.

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.

ABENDU0996

DFSDLBL0

Explanation

Field level sensitivity was specified for a logical child segment and the segment has a PROCOPT of I or L.

Analysis

ABENDU0996 is a standard abend issued from the composite module DFSDLBL0.

Message DFS2435I will accompany this abend.

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.

ABENDU0997

DFSDLBL0, DFSDLB80

Explanation

Internal processing limit exceeded for PSB and PCB.

Analysis
ABENDU0997 is issued from the composite module DFSDLBL0. Message DFS2436I will accompany this abend. The registers saved at label ERRORSVE in 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 below are in DFSDLB70.

Key	Label	Description
Reg5=PCB Reg14=BAL	SUMNOIX1 SUMALTSD SUMNOIX3 SUMGENP2 SUMNDSG SUMLDND	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.

ABENDU0998

DFSDLBL0, DFSDLB50

Explanation
A field defined for a virtual logical child did not fit in the virtual logical child.

Analysis
ABENDU0998 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.

Key	Label	Description
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.

Chapter 4. IMS Failure Analysis Structure Tables (FAST) 1001 - 4095

The following topics provide additional information about abends 1,001 through 4,095.

ABENDU1001

DBFUDLB0, DFSBIND0

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 a DEDB.

Analysis
ABENDU1001 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.

ABENDU1003

DFSRRA00

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 specified incorrectly.

Analysis
ABENDU1003 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.

ABENDU1004

DFSRRA00

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 specified incorrectly.

Analysis
ABENDU1004 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.

ABENDU1005

DFSSBMP0

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. However, 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
ABENDU1005 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
- 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 U1005 will be issued.

ABENDU1006

DFSISI00

Explanation
The number of buffers requested for page fixing exceeds the total number of buffers currently available.

Analysis
ABENDU1006 is a pseudoabend issued during dependent region scheduling when a Fast Path buffer page fix request cannot be satisfied. ABENDU1006 is detected by DFSSMSC0 or DFSSBMP0, and issued by DFSISI00.

The number of page-fixed buffers required is determined as follows:

```
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.

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.

ABENDU1007

DBFCPY00

Explanation
During a Fast Path (IFP) region initialization, program DBFCPY00 detected the scheduled Fast Path PSB contains one or more GSAM PCBs.

Analysis
This 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.

ABENDU1008

DBFATRM0

Explanation
During a Fast Path region termination, module DBFATRM0 detected an error. The error is one of the following:

- An IFP message-driven application program returned normally but without a QC status code posted on I/O PCB or without releasing Fast Path database buffers.
- An MPP, BMP, or IFP nonmessage-driven application program returned normally without releasing Fast Path database buffers.

Analysis
ABENDU1008 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.

Possible Cause
Application program error.

ABENDU1009

Several Modules

Explanation
Either 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.

Analysis
ABENDU1009 is a standard abend issued from the following modules: DBFLRLS0, DBFPUXR0, DBFIBUF0, DBFINI20, DBFINTE0, DBFXPIX0, DFSASK00, DFSDBLM0, DFSDBLP0, DFSDLBL0, DFSPCC20, DFSSABN0, DFSSMSC0.

DBFHIEL0, DBFHQMI0

Explanation
An FP module received a nonzero return code while attempting to release an EMHB using the DFSPOOL REL=EMHB function.

Analysis
ABENDU1009 is a standard abend issued from modules DBFHIEL0 and DBFHQMI0.

Key	Label	Description
Reg15=X'21'		

DBFIBUF0

Explanation
A critical condition that should not occur was detected in DBFIBUF0 while performing a page-fix or page-free call.

Analysis
ABENDU1009 is a standard abend issued from DBFIBUF0. The content of register 15 indicates the reason for the failure.

Key	Label	Description
Reg15=X'05'		An unexpected error occurred in an authorized page-fix or page-free call.

Possible Cause
Internal program logic error.

APAR Processing
z/OS console sheet, IMS control region abend dump, and log data set printout.

DBFINI20

Explanation
A critical condition that should not occur was detected in DBFINI20 while performing the load of the user hash module.

Analysis
ABENDU1009 is a standard abend issued form DBFINI20. The content of register 15 indicates the reason for the failure.

Key	Label	Description
Reg15=X'OD'		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.

DBFINTE0

Explanation
A critical condition that should not occur was detected in DBFINTE0 during an IFP region initialization and termination.

Analysis
ABENDU1009 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.

Possible Cause
Internal program logic error.

APAR Processing
z/OS console sheet, IMS control region abend dump, and log data set printout.

DBFXPIX0

Explanation
A DEDB resource is to be freed, but cannot be found on the DMAC chain.

Analysis
This 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

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 UXRB in the UXRB DMAC chain when all associated XCRBs for the UXRB have been freed.
Reg15=X'12'		Internal error: DBFXPIX0 was called to free a UXRB with a UOW lock not owned by OTHREAD.

Possible Cause
Internal program logic error.

APAR Processing
IMS control region dump.

DBFLRLS0

Explanation
A DEDB resource that is to be freed cannot be found on the DMAC chain.

Analysis
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.

DBFPUXR0

Explanation
A DEDB resource that is to be freed cannot be found on the DMAC or EPST chain.

Analysis
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.

DFSDBLM0, DFSDBLP0, DFSDLBL0, DFSSABN0, DFSSMSC0, DFSCST00

Explanation
A critical condition that should not occur was detected while an exit routine to the Fast Path routine was attempted.

Analysis
ABENDU1009 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.

APAR Processing
IMS control/dependent region abend dump and IMS system definition STAGE 1 output.

DFSPCC20

Explanation
IFP was specified on the EXEC statement in the PARM field, but Fast Path was not generated in this IMS system.

Analysis
This is a pseudoabend issued by module DFSPCC20.

Possible Cause
IFP specified when Fast Path was not generated in this IMS system.

APAR Processing
IMS control region dump.

ABENDU1010

DBFDBIL0

Explanation
DBFDBIL0 found the segment pointer in an ECNT entry had already been used for another segment.

Analysis
Only one terminal-related MSDB segment or nonrelated 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.

ABENDU1011

DBFICIR0, DBFICI10, DBFINI20

Explanation
An IMS control region initialization failed because an error was encountered during Fast Path initialization.

Analysis
ABENDU1011 is a standard abend issued from DBFICIR0, DBFICI10, and DBFINI20.

DBFICIR0

Analysis
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 will point to the instruction label ABEND, where the abend was issued. Register 14 in the abend SVRB will be used as the KEY. Register 12 is the base register, and register 15 will contain 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.

DBFICI10

Analysis
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 slave command ITASKS.
- Create PST-EPST for common service.

The nonzero return code is passed back to DBFICI10 from one of the lower-level modules. The PSW at entry-to-abend will point to the instruction label ABEND, where the abend was issued. Register 14 in the abend SVRB will be used as the KEY. Register 12 is the base register, and register 15 will contain the return code.

Key	Label	Description
Reg1=completion code 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 slave 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.

DBFINI20

Analysis

DBFINI20 initializes Fast Path control blocks. Prior to the abend, one of the following IMS messages can be issued to indicate the reason for the failure: DFS2702A, DFS2703A, DFS2704A, DFS2705A, DFS2711A, or DFS2730A.

Key	Label	Description
Reg1=completion code Reg3=X'01' Reg15=IMODULE GETMAIN return code	ABEND1	Storage could not be obtained for building DMCB/DMAC/BHDR. DFS2702A message is issued.
Reg1=completion code Reg3=X'02' BSIZ is zero and ESCDBUFL is zero	ABEND2	Module DBFINI20 could not find the information about the buffer length. Message DFS2704A is issued.
Reg1=completion code Reg3=X'03' DBBF is zero and ESCDNBUF is zero	ABEND3	The number of buffers defined is zero but MSDBs and DEDBs databases are defined. Message DFS2705A is issued.
Reg1=completion code Reg3=X'04' Reg15=DFSQCSS FUNC=STORAGE return code	ABEND4	Storage could not be obtained for building Fast Path control blocks (DBFCONT0). DFS2703A message is issued.
Reg1=completion code Reg3=X'05' Reg15=DFSQCSS FUNC=LOAD return code	ABEND5	Unable to load the EMH User Input exit routine 0). Nonzero return code was passed from the lower modules of DFSQCSS FUNC=LOAD. DFS2730A message is issued.
Reg1=completion code Reg3=X'06' Reg5=requested buffer size	ABEND6	BSIZ was incorrectly specified in the PARM field on the EXEC job control statement of the IMS startup procedure, in the DFSPRRG0 or in the system definition parameter Fast PathCTRL. DFS2711A message is issued.
Reg1=completion code Reg3=X'07' Reg15=IMSAUTH FUNC=LOCATE return code		Unable to locate the DEDB composite I/O module DBFMMIO0.
Reg1=completion code Reg3=X'08' Reg15=IMSAUTH FUNC=PGFIX return code		Unable to page fix the DEDB composite I/O module DBFMMIO0.
Reg1=completion code Reg3=X'09' Reg15=return code from DBFBCNT0		ECNT load failed. More CNTs were scanned than are defined to the system.
Reg3=X'0A' Reg15=return code		GETMAIN failure for AREALIST storage.

Possible Causes
REGION parameter or BSIZ/DBBF operand in the PARM field on EXEC job control statement of IMS control region startup procedure.

Out of storage condition may have been encountered.

Explanation: At Fast Path initialization, module DBFINI20 calculates the amount of contiguous ECSA storage that is needed in order to load DBFCONT0, which contains the buffers, buffer headers, MSDBs, and other related control blocks. If DBFINI20 cannot obtain a large enough contiguous block of storage, abend U1011 is issued.

When this occurs, you can try IPLing the system, or you can stop other jobs and perhaps free up whatever was preventing DBFINI20 from obtaining the necessary storage.

You can look in register 8, which contains the amount of storage DBFINI20 was trying to obtain. This amount is the accumulated total sizes of the blocks needed by Fast Path. If you receive abend U1011 again, you can quickly perform the following calculation:

```
buffers x buffer size + MSDB_size
```

If the amount you calculate is close to the value in register 8, you can be fairly sure that IMS performed the calculations correctly; this means that the problem is with storage fragmentation.

ABENDU1012

DBFXFP10, DBFDBDL0

Explanation
IMS control region initialization failed because an error was encountered during Fast Path MSDB load determination process.

Analysis
The nonzero return code is passed back to DBFXFP10 from DBFINI10 or DBFDBDL0 issues the abend if DBFINI10 has set the ESCDMSDA bit.

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

Possible Cause
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.

ABENDU1013

DBFCPY00

Explanation
DBFCPY uses the count of segments from the DMCB (DMCBSGNR) 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 DEDB'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.

ABENDU1014

DBFXSL30

Explanation
Program DBFXSL30 attempted to deactivate the dependent region using the SUSPEND macro.

Analysis
ABENDU1014 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.

ABENDU1015

DBFCSTS0, DBFDBDL0, DBFMFLG0

Explanation
The FPS TCB service slave task, DBFCSTS0, received a bad return code from the page-fix/free module, DFSV4200, which was invoked by DBFCSTS0 using IMSAUTH macro.

Analysis
ABENDU1015 is a standard abend issued by DBFCSTS0. Register 2 contains 'FIX'. Register 15 contains the return code. For an explanation of the IMSAUTH return codes, see the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

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.

DBFDBDL0

Explanation
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.

Analysis
ABENDU1015 is a standard abend issued by DBFDBDL0. Register 2 contains 'FIX' or 'FREE'. Register 15 contains the return code. For an explanation of the IMSAUTH return codes, see the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

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.

DBFMFLG0

Explanation
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.

Analysis
ABENDU1015 is a standard abend issued by DFMFLG0. Register 2 contains 'FIX'. Register 15 contains the return code. See the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

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.

ABENDU1016

DBFDBDL0

Explanation
The MSDB image copy load module, DBFDBDL0, has found a mismatch between the checkpoint logical terminal name and the system logical terminal name. (ECNTNAME for checkpoint and system not equal.)

Analysis
ABENDU1016 is a standard abend issued by DBFDBDL0. 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.

ABENDU1017

DBFDBDL0

Explanation
The MSDB image copy load module, DBFDBDL0, 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
ABENDU1017 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.

ABENDU1018

DBFDSRP0

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 ABENDU1018.

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

Key	Label	Description
Reg9		Address of current SDEP buffer chain from DMACXNYW

ABENDU1019

DBFHQMI0

Explanation
An invalid call to Fast Path Queue Manager was made.

Analysis
Register 3 contains the address of the branch table for ABEND routines. ABENDU1019 occurs only when register 3 content is between 20 and 60 or greater than 68.

Possible Cause
Internal program logic or interface error.

ABENDU1020

DBFFATI0

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
ABENDU1020 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.

ABENDU1021

DBFMGAP0

Explanation
Invalid data was returned from a DEDB randomizing module. The possible causes are:

- 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.)
- Register 0 is not a valid anchor point offset within a DEDB area.

Analysis
ABENDU1021 is a standard abend issued by DBFMGAP0. The randomizing module is usually a user-supplied module. If the user has not supplied such a module, one is available from the system. The system sample is named DBFHDC40.

Key	Label	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.

Locate the save area chain at abend, then locate the save area two higher than the current save area. This save area will have c'rand' in the RET (R14) slot. The remaining registers, with the exception of R5, are those returned from the randomizing routine. You can use these registers, along with the information about the data entry database randomizing routine in the *IMS Version 9: Customization Guide*, to determine the cause of the problem in the randomizing routine.

Possible Cause
Randomizing routine error.

ABENDU1022

Multiple Modules

Explanation
A LOCESCD macro was issued to locate the Fast Path SCD extension identified by ESCD. 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 these Fast Path and non-Fast Path modules: The SCD address should be verified and the SCDESCDQ field in the SCD should be verified.

DBFCDAR0	DBFCDDB0	DBFCDPS0	DBFCDRC0
DBFCHKP0	DBFCPID0	DBFCPRC0	DBFCSTS0
DBFCST00	DBFBAU0	DBFEACL0	DBFENIS0
DBFENOT0	DBFEPSB0	DBFERST0	DBFERS20
DBFATC0	DBFFORI0	DBFHIEL0	DBFQMI0
DBFHRTR0	DBFHTMG0	DBFICIR0	DBFICI10
DBFICLI0	DBFICL20	DBFICL40	DBFINI10
DBFINI20	DBFINI30	DBFINI40	DBFINTE0
DBFIRC10	DBFLHCK0	DBFLIRL0	DBFLRH00
DBFNALC0	DBFNCBS0	DBFNOTM0	DBFNOTX0
DBFNRST0	DBFSBLK0	DBFTERM0	DBFTOFN0
DBFTOPU0	DBFTORS0	DBFXFP10	DFSIINM0
DFSMINI0	DFSPSTB0	DFSXNCL0	

Key	Label	Description
Reg2=CL4'ESCD'		Fast Path SCD Extension SCD

Possible Cause
An error in the link or generation of the IMS system with Fast Path.

ABENDU1023

DBFHIEL0

Explanation
The Fast Path message input edit routine detected that the user exit routine was 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.

Analysis
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 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.

ABENDU1024

DBFHGU10, DFSECP10, DFSISI00

Explanation
The Fast Path application program in an IFP region issued a GU call to the I/O PCB after receiving a QC status code on the previous GU call. The program is thus terminated with a pseudoabend by IMS.

Analysis
The application program should terminate on receiving the QC status code. Correct the program and rerun.

Possible Cause
Application program error.

ABENDU1025

DBFHSYN0, DBFIRC10

DBFHSYN0

Explanation
Module DBFHSYN0 detected an invalid return code after the DL/I call was processed by the Fast Path call analyzer.

Analysis
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 U1025 abend routine is a common routine. Register 8 indicates the calling routine.

Key	Label	Description
Reg5=RC Reg8=address of the instruction that detected the error		Routine calling abend. Return code from Fast Path Call Analyzer.

DBFIRC10

Explanation

Program DBFIRC10 detected an invalid return code after the DL/I call was processed by the Fast Path call analyzer.

Analysis

Only a return code of 0, 4, 8, 12, or 16 is valid on return from the Fast Path Call Analyzer. The ABENDU1025 abend routine is a common routine. Register 8 indicates the calling routine.

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.

ABENDU1026

Several Modules

Explanation

An IMS Fast Path module has detected a condition that should not occur and has issued the DBFDEBUG macro or a hard abend to describe the condition.

Analysis

ABENDU1026 is a standard abend issued by these modules:

DBFARDC0	DBFATRM0	DBFBENQ0	DBFBGET0
DBFCHKP0	DBFCSTS0	DBFDRSC0	DBFDT190
DBFHCAS0	DBFMBED0	DBFMCCV9	DBFMCSS9
DBFMCRP9	DBFMCTL0	DBFMDLT0	DBFMDPT9
DBFMDRB0	DBFMDSG9	DBFMFLG0	DBFMFSE0
DBFMGLA9	DBFMGNR0	DBFMGNX0	DBFMGPD0
DBFMGPF0	DBFMGRF0	DBFMIRC9	DBFMIRT0
DBFMLOP0	DBFMOCI0	DBFMPER9	DBFMPIO9
DBFMPSG9	DBFMPSI9	DBFMPUG0	DBFMRCU0
DBFMRPX0	DBFMRQC0	DBFMSEG0	DBFMSFI9
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	DBFVXOI0
DBFXCGL0			

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 will be 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. This 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
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 wish to pursue this problem through the IBM Support Center, 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 forth. For example, the first invalid subcode would be X'033', the second X'133', the third X'233', and so forth, as long as subcode X'33' is needed. The subcodes have the following meanings:

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

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

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

On the following pages 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

DBFARDC0, DBFCSTS0, DBFDT190, DBHCAS0, DBFSYN10, DBFSYN20, DBFMSRT0, DBFUHDSO, DBFVSOP0, DBFVXOI0

Analysis

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

DBFATRM0

Analysis

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).

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 - EPSTR1SL is set, Reg10 + X'B5' = X'80'

DBFBENQ0

Analysis

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

DBFBGET0

Analysis

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

DBFCHKP0

Analysis

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

DBFDRSC0

Analysis

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

DBFMCCV9

Analysis

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

DBFMCRP9

Analysis

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. DBFMPGO0 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

DBFMCSS9

Analysis

Message Text: **UNEXPECTED SEGMENT CODE**

Key	Label	Description
Subcode=X'53'	DEBUG	The segment code for the retrieved segment does not agree with that defined for the segment type.

Normal register usage is as follows:

Register	*Contents*
0	Offset in buffer to segment
1	Subcode
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)

DBFMCTL0

Analysis

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.

Key	Label	Description
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. DBFMPGO0 was called, and the abend was issued from macro DBFMRCCS. 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

DBFMDLT0

Analysis

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

DBFMDPT9

Analysis

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

DBFMDRB0

Analysis

Message Text: **NO TEXT SUPPLIED**

Key	Label	Description
Subcode=X'53'		The value in DSEGCODE (R7 + X'00') should be 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 at this point.

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

8	DMAC address
9	DMHR address
10	EPST address
11	ESCD address
15	DMCB address

DBFMDSG9

Analysis

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. DMHRBUFP (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.

DBFMFLG0, DBFSDEQ0, DBFSMP10

Analysis

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

DBFMFSE0

Analysis

Message Text: **LENGTH PARM (R7) INVALID**

Key	Label	Description
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**

Key	Label	Description
Subcode=X'12' Reg2=MLTE address Reg3=EPCB address Reg5=DFSE address Reg7=length request in low-order 2 bytes 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	FSEBAD	The length of the FSE is greater than allowed by the physical record length.

Message Text: **FREE IOVF CI HAS WRONG UOW#**

Key	Label	Description
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#**

Key	Label	Description
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 BALinstruction) 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 that of the current FSE. The chain is in error.

Key	Label	Description
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.
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	***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

DBFMGLA9

Analysis

Message Text: **NO TEXT SUPPLIED**

Key	Label	Description
Subcode=X'20'	DEBUG	PTRPMLTE (Reg2) is equal to zero (Reg14). This is invalid.
Subcode=X'120'	DEBUG	The pointer in register 7 (MLTECRBA - Reg4 + X'14') is zero. A zero pointer is invalid.

Key	Label	Description
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. DMHRBUFP (Reg9 + X'10') is the address of the buffer.
Subcode=X'33'	LOGERRO2	An invalid return code was received from DBFMCCS9; 0, 4, 8, and 12 are valid return codes.

Normal register usage is as follows:

Register	*Contents*
1	Subcode
3	EPCB address
4	MLTE address
10	EPST address
11	ESCD address

DBFMGNR0

Analysis

Message Text: **WRONG SEGMENT CODE**

Key	Label	Description
Subcode=X'53'		The return code from DBFMPGO0 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

DBFMGNX0

Analysis

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
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

DBFMGPD0

Analysis

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

DBFMGPF0

Analysis

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 may exist.
Subcode=X'32'		IMSAUTH pagefix error - A real storage shortage may exist.

Register	***Contents***
1	Subcode
10	EPST address
11	ESCD address
15	Return code

DBFMGRF0

Analysis

Message Text: **DEDB PROBABLY HAS ERROR IN IT**

Key	Label	Description
Subcode=X'150' Reg6=address of segment, DSEGCODE is first byte Reg7=last segment pointer RBA Reg14=return address from subroutine Reg15=DMCB address	NEXTITEM	DSEGCODE (Reg2 + X'00'), the first byte of segment, is X'00'
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.

Key	Label	Description
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 DBFMPGO0. The return code from DBFMPGO0 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 should be 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 should be 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.

Register	***Contents***
1	Subcode
2	MLTE address
3	EPCB address
8	DMAC address
9	DMHR address
10	EPST address

11	ESCD address

DBFMIRC9

Analysis

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

DBFMIRT0

Analysis

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

DBFMLOP0

Analysis

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

DBFMOCI0

Analysis

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

DBFMPER9

Analysis

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

DBFMPIO9

Analysis

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

DBFMPSG9

Analysis

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

DBFMPSI9

Analysis

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

DBFMPUG0

Analysis

Message Text: **NO TEXT SUPPLIED**

Key	Label	Description
Subcode=X'20' Reg8=DMAC address Reg9=0	DEBUG	Previous MLTE pointer is equal to zero. This is invalid.
Subcode=X'120' Reg8=DMAC address Reg9=0	DEBUG	The pointer in MLTEPARP (Reg4 + X'3C') is zero. This 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.
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

DBFMRCU0

Analysis

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 DBFMRCCS. 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

DBFMRPX0

Analysis

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

DBFMRQC0

Analysis

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	ESCDMSGA (ESCD + X'3E8')

DBFMSEG0

Analysis

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

DBFMSFI9

Analysis

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.

Register	***Contents***
1	Subcode
3	EPCB address
4	MLTE address
10	EPST address
11	ESCD address

DBFMSFO9

Analysis

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'1E') is not equal to the DSEGCODE of the segment pointed to by register 7. The abend is issued from macro DBFMRCPS. DMHRDMAC (Reg9 + X'20') points to the DMAC. DMHRBUFP (Reg9 + X'10') is the address of the buffer.
Subcode=X'153' Reg6=offset in buffer to segment Reg7=address of previous segment, DSEGCODE's first byte	DEBUG	MLTESGCD of the segment (Reg4 + X'1E') is not equal to the DSEGCODE of the segment pointed to by register 8. 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
2	Address of previous MLTE
3	EPCB address
4	MLTE address
10	EPST address
11	ESCD address

DBFMSSA9

Analysis

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 DBRMRCCS. 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

DBFMSSD9

Analysis

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 indicates data was moved at this level.

Register	***Contents***
1	Subcode
3	EPCB address
4	MLTE address
10	EPST address
11	ESCD address

DBFMSSR9

Analysis

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

DBFMUHE0

Analysis

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
9	DMHR address from caller. Address of DMAC for Area is in DMHRDMAC field.
10	EPST address
11	ESCD address

DBFMVSN9

Analysis

Message Text: **LOGICAL ERROR IN ACTION MODULES**

Key	Label	Description
Subcode=X'20'		EPCBPRGP (Reg0, loaded from Reg3 + X'60') contains zero, but should contain 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

DBFPFDS0

Analysis

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

DBFPFPB0

Analysis

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

DBFPGAP0

Analysis

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

DBFSLG20

Analysis

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

DBFSLOG0

Analysis

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

DBFSYP20

Analysis

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)

DBFTOPU0

Analysis

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

DBFUHCF7

Analysis

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 CIDF).

Normal register usage is as follows:

Register	*Contents*
1	Subcode
2	FSE address in buffer where segment will be moved.
9	UHSW address

DBFUHGS7

Analysis

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

DBFUMAF0

Analysis

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

DBFUMAI0

Analysis

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

DBFUMAN0

Analysis

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

DBFUMCB9

Analysis

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

DBFUMCF9

Analysis

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, and should equal the precalculated offset to the CI end overhead at UTDWDEN0 (Reg11 + X'8A8'). The two values are not equal.

Register	***Contents***
1	Subcode
3	CITB address
4	Buffer address

9	DMAC address
10	SDBF address
11	UTDW address

DBFUMCW9

Analysis

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

DBFUMFR9

Analysis

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 should remain, 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 should remain, but a different value was found.

Register	*Contents*
1	Subcode
2	Index
3	EPCB address
4	MLTE address
10	EPST address
11	UTDW address

DBFUMGS9

Analysis

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

DBFUMIM9

Analysis

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.

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.

DBFUMPI0

Analysis

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 UTDWPDCB (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

DBFUMSC0

Analysis

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

DBFUMTQ9

Analysis

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

DBFXCGL0

Analysis

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 should not occur, but 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')

ABENDU1027

Several Modules

DBFBENQ0, DBFCBHL0, DBFSLOG0

Explanation

A logical error occurred while attempting to enqueue or dequeue a Fast Path resource.

Analysis
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'.
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.

DBFSLOG0

Explanation
An unacceptable return code was received from the IMS ENQ/DEQ routine DFSLRH00.

Analysis
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.

DBFMGXC0

Explanation
An unacceptable return code was received from the IMS ENQ/DEQ routine DFSLRH00.

Analysis
There are two routines within DBFMGXC0 that branch to the abend routine, as indicated in the table below.

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'	ENQORTN	Internal logical error; these codes should not occur.

Possible Cause
Internal program logic error.

DBFPUXC0

Explanation
During UOW contention detection, DBFPUXC0 found a UXRB in exclusive mode on the UXRB DMAC chain.

Analysis

Key	Label	Description
Reg15=X'11'		A UXRB in exclusive mode was found on the UXRB DMAC chain during UOW contention detection. The new UXRB 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'		DBFPUXC0 tried to chain a new UXRB to the UXRB EPST chain, but a UXRB for the same area and RBA was already on this chain.

DBFSYN20

Explanation
During SYNC, COMMIT is checked to be certain it is valid before known locks are freed.

Analysis

Key	Label	Description
Reg15=X'14'		DBFSYN20 frees locks through DFSLR but the lock table is either not available or not accessible. -An ABENDU113 can also occur if DL/I, DC, or Fast Path I/O was processing at the time of the ABENDU0127.

DBFXPIX0, DBFSDEQ0, DBFLRLS0, DBFPUXR0, DBFSLG20

Explanation
A nonzero return code was received from the IMS ENQ/DEQ routine (DFSLRH00).

Analysis
The abending module 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.

ABENDU1028

DBFUHRE0

Explanation
Module DBFUHAC7 found a non-ascending key for a segment during area reorganization, and subcode = X'C' was issued.

Analysis
At the time of the U1028 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.

ABENDU1029

DBFXWU30

Explanation
Program DBFXWU30 attempted to reactivate a dependent region that was not deactivated.

Analysis
ABENDU1029 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.

ABENDU1030

DBFXTCU0

Explanation
Program DBFXTCU0 attempted to reactivate a dependent region that was not deactivated.

Analysis
ABENDU1030 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.

ABENDU1031

DBFDBDP0, DBFERS20, DBFERS21, DBFTOPU0

Explanation
An IMS Fast Path module received a nonzero return code from the page fix/free module, DFSV4200, which was invoked by the IMSAUTH macro.

Analysis
ABENDU1031 is a standard abend issued by any one of the three modules. Register 2 contains 'FIX' or 'FREE'. Register 15 contains the return code. For an explanation of the IMSAUTH return codes, see the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

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.

ABENDU1032

DBFDBDP0

Explanation
The MSDB Checkpoint/Dump module, DBFDBDP0, 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
ABENDU1032 is a standard abend issued by DBFDBDP0. 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
- Hardware error.
- Internal program logic or interface error.

ABENDU1033

DFSECP10, DFSECP20, DFSISI00

Explanation
The message-driven application program or MPP required more Fast Path database buffers than were specified by the NBA and OBA parameters in the dependent region JCL parameters.

Analysis
ABENDU1033 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.

Possible Cause
Application program error, or either the NBA, OBA, or both parameters were specified too small.

ABENDU1034

DBFATRM0, DBFCSTS0, DBFDT900, DBFIRC10, DBFMER00, DBFMFLG0, DBFMLCL0, DBFMLOP0, DBFPICS0, DBFMPOP0, DBFMSDP0, DBFSGAB0, DBFXVUN0

DBFATRM0, DBFIRC10

Explanation
An ISWITCH could not be executed because a dependent region was terminating.

Analysis
ABENDU1034 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.

DBFCSTS0, DBFMER00,DBFMFLG0, DBFXVUN0

Explanation
Either an ISWITCH to the CTL TCB or an ISWITCH back to the XFP TCB cannot be executed.

Analysis
An abendU1034 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.

DBFDT900

Explanation
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.

Analysis
ABENDU1034 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

Key	Label	Description
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)

DBFMLCL0

Explanation

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.

Analysis

ABENDU1034 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 Reg14=BALR Reg15=ISWITCH TO=FPE return code Reg15=ISWITCH TO=RET return code		A nonzero return code was passed from the DFSIDSP0 (ISWITCH entry).

DBFMLOP0

Explanation

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.

Analysis

ABENDU1034 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 Reg14=BALR Reg15=ISWITCH TO=FPE return code Reg15=ISWITCH TO=RET return code		A nonzero return code was passed from the DFSIDSP0 (ISWITCH entry).

DBFMPOP0

Explanation

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.

Analysis

ABENDU1034 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 Reg14=BALR Reg15=ISWITCH TO= FPE return code Reg15=ISWITCH TO=RET return code		A nonzero return code was passed from the DFSIDSP0 (ISWITCH entry).

DBFMSDP0

Explanation
Module DBFMSDP0 is a Fast Path module that sets up the area SDEP buffer.

Analysis
ABENDU1034 is a standard abend issued by DBFMSDP0. Module DFSIDSP0 (the ISWITCH entry) passed a nonzero return code.

DBFPICS0

Explanation
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.

Analysis
ABENDU1034 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).

DBFSGAB0

Explanation
Module DBFSGAB0 is a Fast Path module that gets a buffer from a shared pool.

Analysis
AbendU1034 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.

ABENDU1035

DBFMSRH0

Explanation
A search of a DEDB area name list (DBFAREA) for a specified DDNAME resulted in a 'NOT FOUND' condition.

Analysis
ABENDU1035 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.

ABENDU1036

DBFMER00

Explanation
The error message number is not included in the error message table used by module DBFMER00.

Analysis
ABENDU1036 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.

ABENDU1038

DBFERST0

Explanation
During an emergency restart, a VSAM GENCB macro was executed and a nonzero code was returned from VSAM.

Analysis
ABENDU1038 is a standard abend issued by DBFERST0. This is a standard abend issued by module 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 are documented in: DFSMS Macro Instructions for Data Sets.

Possible Cause
The operands in the GENCB macro were specified incorrectly.

ABENDU1039

DBFUMRT0

Explanation
The option **ERRORACTION=ABEND** provides diagnostic information when FP DEDB on-line utility ends with an error message.

ABENDU1041

DFSEIPB0

Explanation
The high level programming interface (HLPI) found a condition that was caused by a programming error, or DL/I returned a status code to HLPI, which indicates a programming error.

Analysis
ABENDU1041 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.

ABENDU1046

DBFUMDL0, DBFUMDL0, DBFUMRT0

Explanation
The SDEP utility program has detected an error condition. Messages DFS2712I are 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 timestamp, the start position RBA, the stop position RBA and their options, the working buffer set, and the registers at time of abend.

Analysis
ABENDU1046 is a standard abend issued from the following modules: DBFUMDL0, DBFUMSC0, and DBFUMRT0.

Register 14 contains the address of the instruction before the abendU1046 is issued.

The subcodes have the following meanings:

Key	Label	Description
Subcode=X'91'		Timestamp conversion not successful.
Subcode=X'92'		Conversion to store clock 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.

ABENDU1050

DFSICVE0

Explanation
During online change for security, a DFSCBTS scan resulted in a nonzero return code.

Analysis
ABENDU1050 is issued by module DFSICVE0. Register 15 contains the return code from DFSCBTS, register 14 points to the routine where the failure was detected.

Possible Cause
The Security Maintenance utility matrices are inconsistent with the current number of resources in the IMS system.

APAR Processing
ABENDU1050 dump and Security Maintenance utility output.

ABENDU1060

DFSSBI00/DBFPHI00

Explanation
The DFSCTL data set contains an error. Messages DFS0491A and DFS0492I are issued along with this abend. The dependent region is terminated.

Analysis
Correct the error based on the information on messages DFS0491A and DFS0492I in *IMS Version 9: Messages and Codes, Volume 2*.

DBFCPY00

Explanation
IMS found a problem with the PSB scheduled in the dependent region, and issued one of the following messages: DFS0501A, DFS0504A, or DFS0525A.

Analysis
Correct the error; see the message information in *IMS Version 9: Messages and Codes, Volume 2*.

ABENDU1061

DBFPCAA0, DBFPHI00, DBFCPY00

Explanation
The GETMAIN routine failed to obtain segment workareas (EPSTSEG1 and EPSTSEG2) in sp231 ECSA.

Analysis
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.

ABENDU1062

DBFPICS0, DBFSIC10, DFSECP10

Explanation
HSSP image copy failed.

Analysis
ABENDU1062 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.

ABENDU1063

Several Modules

DBFIBUF0

Explanation
While attempting to pagefix/pagefree database buffers, a private buffer was found on the available queue. The IMS control region is terminated.

Analysis
This is an internal IMS system error. Contact the IBM Support Center for help in determining the specific problem.

DBFPFAB0

Explanation
A normal DB buffer was returned to the free private buffer routine. The IMS control region is terminated.

Analysis
This is an internal IMS system error. Contact the IBM Support Center for help in determining the specific problem.

DBFMDBQ0

Explanation
DBFMDBQ0 found a dummy XCRB pointing to a DMHR while trying to queue buffer headers (DMHRs) and XCRBs to their respective queues.

DBFPGAB0

Analysis
Message Text: **NOT ENOUGH PRIVATE BUFFERS OBTAINED**

Register	*Contents*
2	Address of DMAC
11	Address of ESCD

DBFSGAB0

Explanation
Csect DBFSGAB1 of module DBFSGAB0 gets a buffer from a shared pool for restart.

Analysis
AbendU1063 is a standard abend issued by DBFSGAB1 when restart cannot get a buffer and the pool is not being extended. This is an internal IMS system error. Contact the IBM Support Center for help in determining the specific problem.

ABENDU1064

DBFPFPB0

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. The IMS control region is terminated.

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.

ABENDU1065

DBFIRC10

Explanation
An application made a call with a PCB defined with PROCOPT H and PROCOPT GO. This is an invalid combination of processing options. The dependent region is terminated.

Analysis
Correct the invalid combination of processing options, or use another PCB on the call.

ABENDU1500

DFSMDA00

Explanation
During an attempt to recover a database, an incompatibility occurred between the task I/O and the scheduler work areas. A ddname was allocated so that the database 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.

ABENDU1501

DFSMDA00

Explanation
Database recovery was attempted for an unsupported DASD.

Analysis
This is a standard abend issued by module DFSMDA00.

Possible Cause
User error.

ABENDU2017

DXRRL1D0, DXRRL170

Explanation
IRLM established a functional recovery routine (FRR) to intercept abends that occur during its execution under the IMS execution unit (TCB or SRB) during the processing of an RLMREQ request.

The RLMREQ request or IMS exit routine processing was abnormally terminated. An IRLM FRR intercepted the abend and requested a z/OS SYS1.LOGREC entry and an SDUMP of the failure. The FRR

issued a CALLRTM to terminate the IRLM with abend code 2017. Console message DXR123E, issued when the IRLM terminates, contains the z/OS error ID recorded in the SYS1.LOGREC record and in the SDUMP.

Analysis
ABENDU2017 is a standard abend issued by DXRRLID0 and DXRRL170. User abend 2017 terminated the IRLM. The original problem was intercepted by the FRR to avoid abnormal termination of the IMS system.

Analyze and, if necessary, report the original failure, not ABENDU2017.

Analyze the SDUMP to determine the original problem. The problem usually is a program check within IRLM code.

1. Locate the program status word (PSW) and register contents at entry to abend, either from the software LOGREC entry, or from the RTM2WA summary in the formatted section of the SDUMP.

   If the program status word (PSW) is not within an IRLM module, determine the system component where the abend occurred, and use the diagnostic procedure for that component to resolve the problem. IRLM modules are prefixed with DXR.
2. Use the software LOGREC entry or the RTM2WA summary entry for the original error in the related SRB for problem diagnosis.
3. Register 9 normally contains the address of the RLMCB if the error occurred during IRLM processing.
4. Register 12 normally contains the base register contents for the module in control at the time of the error.

ABENDU2018

DXRRL010

Explanation
The IRLM initialization failed. Console messages DXR116E is issued at the time of the failure. A z/OS SYS1.LOGREC entry and an SDUMP were requested.

Analysis
ABENDU2018 is a standard abend issued by DXRRL010.

An analysis of the SDUMP or z/OS dump is necessary to determine the reasons for specific failures.

Register 15 at the time the abend command was issued contained one of the following reason codes:

Register	***Contents***
X'01'	An invalid EXEC parameter was issued.
X'02'	A parameter or option was invalid, or a required option was missing. was missing or invalid.
X'03'	An invalid SCOPE parameter value was issued.
X'04'	IRLM was already active.
X'05'	IRLM was not defined as a z/OS subsystem.
X'06'	An error building IRLM RLMCB occurred.
X'07'	A failure occurred in an IRLM initialization routine.
X'09'	The IRLM is already active.

X'0B'	A COMPARE-and-SWAP instruction failed to activate the IRLM SSVT. The IRLM is already active.
X'0C'	The SETDIE service failed to establish a first deadlock time interval.
X'0D'	Invalid PC parameter specified.
X'0E'	PC=yes specified at z/OS and does not support close memory.
X'0F'	Invalid ITRACE keyword.
X'10'	Too many APPL parameters specified.
X'11'	APPL parameter not specified in pairs.
X'12'	Invalid RULES parameter specified.

ABENDU2019

DXRRL400

Explanation

The IRLM storage manager detected invalid or inconsistent control information within its storage pool structure and did not attempt to complete the request that detected this error. Global sharing with this IRLM terminated. z/OS system console message DXR122E was issued at the time of the failure. A z/OS SYS1.LOGREC entry and an SDUMP were requested.

Analysis

ABENDU2019 is a standard abend issued by DXRRL400. Analyze the SDUMP to determine the problem.

Register 9 contains the address of the RLMCB.

Register 12 contains the base register contents for the module in control at the time of the error.

Register 14 contains the return address of the caller of storage management.

Register 15 at the time the abend was issued contained one of the following reason codes:

Register	***Contents***
X'01'	A GET serialized request detected a destroyed storage pool.
X'02'	A GET unserialized request detected a destroyed storage pool.
X'03'	A FREE serialized request attempted to free an element that was already freed.
X'04'	A FREE serialized request detected a destroyed storage pool.
X'05'	A FREE unserialized request attempted to free an element that was already freed.
X'06'	A FREE unserialized request detected a destroyed storage pool.

ABENDU2020

DXRRL020

Explanation

The IRLM was terminated abnormally by the `F irlmproc,ABEND` command and global sharing with the IRLM was consequently terminated. Console message DXR124E was issued at the time of the failure. A z/OS SYS1.LOGREC record and an SDUMP were requested. If the NODUMP parameter was omitted, an SDUMP was requested.

Analysis
ABENDU2020 is a standard abend issued by DXRRL020. The IRLM control block structure is formatted when the SDUMP is printed. This structure shows the IRLM status at the time the SDUMP was taken.

Register 9 contains the address of the RLMCB.

ABENDU2022

DXRRL010, DXRRL020

Explanation
An IRLM subtask abnormally terminated. The end-of-task (ETXR) routine specified for the subtask by an ATTACH parameter was entered. The end-of-task routine issued an ABENDU2022 to force IRLM to terminate. Global sharing with the IRLM terminated. z/OS system console message DXR122E was issued at the time of the failure. A z/OS SYS1.LOGREC record and an SDUMP were requested.

Analysis
ABENDU2022 is a standard abend issued by DXRRL010 and DXRRL020. When the PTB TASK subtask abnormally terminates, its ESTAE requests the recording of the failure in SYS1.LOGREC and an SDUMP. ABENDU2022 does not take a second SDUMP.

When other IRLM subtasks abnormally terminate, no ESTAE exists, so z/OS takes an abend dump (the default action). In these cases, the SYSABEND or SYSUDUMP dump can provide additional information. In addition, an SDUMP is taken.

Analyze and, if necessary, report the original abend on the subtask (PTB TASK, DXRRL080 or DXRRL0B0), not ABENDU2022.

Analyze the SDUMP to determine the reason for the original abend.

1. Locate the program status word (PSW) and register contents at entry to abend either from the LOGREC entry or from the RTM2WA summary in the formatted section of the SDUMP.

   If the program status word (PSW) is not within an IRLM module, determine the system component where the abend occurred. Use the diagnostic procedure for that component to resolve the problem. IRLM modules are prefixed with DXR.
2. Use the LOGREC entry or the RTM2WA summary entry for the original error in the subtask for problem diagnosis.
3. Register 9 normally contains the address of the RLMCB if the error occurred during IRLM processing.
4. Register 12 normally contains the base register contents for the module in control at the time of the error.

ABENDU2023

Several Modules

Explanation
The IRLM encountered an internal logic error or received an error from a z/OS service that must be performed for IRLM to continue execution. An example of an IRLM internal error is the detection of an invalid function code in an internal queue element. An example of a z/OS service failure is a nonzero return code from SETDIE. The IRLM deadlock and Intersystem Communication (ISC) functions cannot be performed without SETDIE services. Do not force termination of the IRLM. Global sharing with this IRLM terminated. z/OS system console message DXR122E was issued at the time of the failure. A z/OS SYS.LOGREC record and an SDUMP were requested.

Analysis

ABENDU2023 is a standard abends issued by modules: DXRRL210, DXRRL220, DXRRL240, DXRRL250, DXRRL300, DXRRL310, DXRRL330, DXRRL370, DXRRL1F0, DXRRL2A0, DXRRL2E0, DXRRL2G0, DXRRL2J0, and DXRRL2K0.

Analyze the SDUMP to determine the reasons for the specific failures.

1. Locate the program status word (PSW) and register contents at entry to abend either from the LOGREC entry or from the RTM2WA summary in the formatted section of the SDUMP.

   If the program status word (PSW) is not within an IRLM module, determine the system component where the abend occurred. Use the diagnostic procedure for that component to resolve the problem. IRLM modules are prefixed with DXR.
2. Register 9 normally contains the address of the RLMCB if the error occurred during IRLM processing.
3. Register 12 normally contains the base register contents for the module in control at the time of the error.

Register 15 contains one of the following reason codes:

Register	***Contents***
X'01'	Module DXRRL210 GETMAIN for the ISL-SEND RLMQE element failed.
X'02'	Module DXRRL210 ISL-MERGE detected the same IMS subsystem identified to both IRLMs.
X'03'	Module DXRRL210 GETMAIN for merged ISL storage failed.
X'04'	Module DXRRL300 SETDIE return code for initial PTB process-delay-time interval failed.
X'05'	Module DXRRL310 SETDIE return code for PTB wait-time interval failed.
X'06'	Module DXRRL310 SETDIE return code for PTB process-delay-time interval failed.
X'07'	Module DXRRL330 detected an unexpected RLMQE element while processing request queues.
X'08'	Module DXRRL330 attempted to free an RH-owned RLMQE element. A logic error in the PTB-RH protocols was found.
X'09'	Module DXRRL330 request for a storage pool for the RLMPPTBQ element failed.
X'0A'	Module DXRRL370 issued a request to obtain storage for an RLMPPTBQ element that failed.
X'0B'	Module DXRRL370 issued a request to obtain storage for an RLMPQE60 element that failed.
X'0C'	Module DXRRL370 issued a request to obtain storage for an RLMPNCB element that failed.
X'0D'	Module DXRRL370 GETMAIN for an RLMQE element failed.
X'0E'	Module DXRRL240 GETMAIN for the deadlock workspace failed.
X'0F'	Module DXRRL250 GETMAIN for the deadlock parameter list (DPL) storage failed.
X'10'	Module DXRRL250 SETDIE for the deadlock-time interval failed.
X'11'	Module DXRRL2A0 or DXRRL2E0 issued a request to obtain storage for an RMLPCOMQ element that failed.
X'13'	Module DXRRL2A0 issued a request to obtain storage for an RLMPQE28 element that failed.
X'14'	Module DXRRL1F0 issued a request to obtain storage for an MLB that failed.

X'15'	Module DXRRL2K0 issued a request to obtain storage for an 11-25 RLMQE that failed.
X'16'	Module DXRRL2K0 issued a request to obtain storage for an MLB that failed.
X'17'	Module DXRRL2K0 issued a request to obtain storage for an 11-20 RLMQE that failed.
X'I8'	Module DXRRL2K0 issued a request to obtain storage for a dummy WHB that failed.
X'19'	Module DXRRL220 issued a request to obtain storage for a dummy WHB that failed.
X'1A'	Module DXRRL2E0 issued a request to obtain storage for a dummy WHB that failed.
X'1B'	Module DXRRL2C0 issued a request to obtain storage for an RLSPL that failed.
X'1C'	Module DXRRL2C0 issued a request to obtain storage for an SRB that failed.
X'1D'	Module DXRRL2G0 issued a request to obtain storage for an 07-05 RLMQE that failed.
X'1E'	Module DXRRL350 issued a double DXRRFSAV macro.
X'1F'	Module DXRRL350 failed to obtain PQE28 storage.
X'20'	Module DXRRL200 encountered a zero secondary latch use count (RLMUCNT/RLMRHSTA).
X'21'	Module DXRRL200 failed to get storage for an SRB or RHWKA.

ABENDU2024

DXRRL100

Explanation

A functional recovery routine (FRR) was established by the IRLM to intercept abends that occur while executing under the IMS execution unit (TCB or SRB), during the processing of some RLMREQ request.

The IRLM encountered a program check while attempting to access an IMS-owned storage area, either an RLPL (RLMREQ interface control block), or some other storage address passed to the IRLM by the RLPL (hat is, a VERIFY list). The error, as well as symptom string data, was recorded in SYS1.LOGREC, and the FRR retried to DXRRLI00 to clean up the IRLM-owned structures.

Analysis

The user abend did not terminate the IRLM, but abended the IMS execution unit.

Examine the SYS1.LOGREC entry recorded by the IRLM FRR to extract the program status word (PSW) and registers at the time of the abend.

At the time the abend was issued, register 10 contained one of the following codes:

Register	***Contents***
X'05'	An invalid IMS owned RLPL storage address was detected after completion of the IRLM request. The request was processed in cross-memory mode (PC=YES).
X'07'	An invalid IMS owned RLPL storage address was detected before completion of the IRLM request. The request was processed in cross-memory mode (PC=YES).
X'09'	An invalid IMS owned storage address (other than the RLPL) was detected, while processing the request in cross-memory mode (PC=YES).

In all cases, the invalid IMS-owned storage address is available from the variable recording area (VRA) of the SDWA, recorded within the SYS1.LOGREC entry.

ABENDU2025

Several Modules

Explanation
ABENDU2025 is issued by the following modules: DXRCHNGP, DXRRL120, DXRRL2E0, DXRRL2R0, DXRRL2R1, DXRRL2R2, DXRRL2R3, DXRRL2R4, DXRRL2T0, DXRRL2T1, DXRRL2U0, DXRRL2V0, DXRRL7C1, DXRRL7C2, DXRRL711, DXRRL712, DXRRL730, DXRRL732, DXRRL750, DXRRL752, DXRRL753, DXRRL754, DXRRL760, DXRRL770, and DXRRL780. The IRLM request to the cross-system extended services (XES) or the cross-system coupling facility (XCF) failed. The job step ESTAE issued console message DXR139E with the return and reason codes. Message DXR122E was also issued. An SDUMP was requested.

Analysis
See messages DXR122E and DXR139E for more information. Obtain the SDUMP of the requesting IMS subsystem. The dump title includes a module name and an offset into the module where the abend was issued. Contact the IBM Support Center for help in determining the problem.

ABENDU2027

Several Modules

Explanation
This abend code is issued by the following modules: DXRGLBL, DXRRL2R1, DXRRL2R4, DXRRL2S0, DXRRL2T0, DXRRL2T2, DXRRL2V0, DXRRL200, DXRRL220, DXRRL248, DXRRL7B1, DXRRL700, DXRRL711, DXRRL712, and DXRRL770.

IRLM detected a logical inconsistency in either its own processing or the local lock structure. An SDUMP of the IRLM address space was requested.

Analysis
Obtain the SDUMP. The dump title includes a module name and an offset into the module where the abend was issued. Contact the IBM Support Center for help in determining the problem.

ABENDU2031

DXRRL2F0

Explanation
An abend occurred while processing an SRB dispatched to an cross-system coupling facility (XCF) exit or to an System Lock Manager (SLM) exit in the IRLM address space. The exit functional recovery routine (FRR) issued a CALLRTM with a completion code of 2031 to terminate the IRLM address space.

A SYS1.LOGREC record was written. The z/OS error ID was placed in the IRLM RLMCB control block. The job step ESTAE issued message DXR122E to the console and requested an SDUMP.

Analysis
See message DXR122E for more information. Obtain the SDUMP. The dump title includes a module name and an offset into the module where the abend was issued. Contact the IBM Support Center for help in determining the problem.

ABENDU2478

DFSFXC10

Explanation
The application program has been chosen to abend because it must wait for a resource and DFSFXC10 cannot handle any more systems in a WAIT status. A message processing program (MPP) is automatically rescheduled.

Analysis
This is a pseudoabend issued by DFSFXC10.

ABENDU2479

DFSDBH10, DFSDVSM0

Explanation
Insufficient storage is available for a buffer queue elements, (BQEL). A message processing program (MPP) is automatically rescheduled.

Analysis
ABENDU2479 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 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.

ABENDU2480

Various

Explanation
An internal error in Database Recovery Control (DBRC) was detected. Message DSP0300I identifies the DBRC module that detected the error and indicates the nature of the error. Refer to message DSP0300I in *IMS Version 9: Messages and Codes, Volume 1*.

Analysis
Contact your IBM Support Center for assistance.

ABENDU2481

DFSULGI0, DBFLOGC0, DFSUCMN0, DFSUICP0

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

ABENDU2482

DSPCRTR0

Explanation
DBRC was called while servicing a previous request. DBRC can process only one request at a time.

Analysis
DBRC detected a programming error on the part of the caller. 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.

ABENDU2483

DSPLOADR

Explanation
A module within DBRC has attempted to call either a nonexistent entry point within a known DBRC load module, or an entry point within an unknown load module.

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. ABENDU2483 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 re-compile or properly link-edit a module affected by a recent change in DBRC's "call intercept" mechanism.

ABENDU2484

DBFDIDT0, DBFDT300, DBFICLI0, DBFMEQE0, DBFMFLG0, DBFMLBI0, DBFMLBR0, DBFMLOP0, DBFNRST0, DBFUMER0, DBFARD10, DBFARD30, DBFARD40, DBFCST00, DBFVXCS0, DBFVXCS0, DBFICL40

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 (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 AWE cannot be obtained.
Reg2=C'ADSC'		The ADSC cannot be obtained.
Reg2=C'FNCB'		The FNCB cannot be obtained.
Reg2=C'FPCP'		The FPCP cannot be obtained.
Reg2=C’FNCB’		The FNCB cannot be obtained.
Reg2=C'L56X'		The L56X block cannot be obtained.

ABENDU2485

DBFMLOP0, DBFMLCL0

Explanation
A DMAC exclusive latch hold that uses the DBFLATCH macro failed because it already existed.

Analysis
ABENDU2485 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.

ABENDU2488

DBFERST0, DBFEAIS0, DBFE2CI0

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

ABENDU2488 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.

ABENDU2489

DBFMOCI0, DBFEACL0

Explanation

An asynchronous work element (AWE) enqueue command issued to the Fast Path Enhancement common service ITASK routine, to stop the DEDB area, failed because of an internal error.

Analysis

ABENDU2489 is a standard abend issued from the Fast Path Enhancement DEDB second CI update processor (DBFMOCI0), or the Fast Path XRF area cleanup module (DBFEACL0). The program status word (PSW) at entry-to-abend points to the instruction within DBFMOCI0 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
```

ABENDU2496

DBFARD10, DBFARD20, DBFARD50, DBFEAIS0

Explanation

A critical request for a DEDB area lock failed. A bad return code was received from IRLM.

Analysis

ABENDU2496 is a standard abend issued by DBFARD10, DBFARD20, DBFARD50, 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. The abend is issued as follows:

DBFARD10 and DBFARD20:

- RC X'14' is returned from IRLM and LMRJECT 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).

DBFARD50:

- 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.

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). IRLM request return and reason codes can be found in *IMS Version 9: Messages and Codes, Volume 1*.

ABENDU2763

DBFNRST0

Explanation

During an IMS restart (warm start or emergency restart), the Fast Path normal restart processor, DBFNRST0, 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 DBFNRST0 and ABENDU2763 is issued. This could occur because the previous IMS cold start was performed without MSDBs specified.

ABENDU2800

DFSIIOC0, DFSCSL40

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:

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 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 IMSs in the IMSplex. All of the IMSs in the IMSplex must define the same OLCSTAT data set name.
X'10'	Global online change is in progress. IMS is not permitted to initialize while global online change is in progress.

ABENDU2801

DFSOLCS0

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'	Global online change occurred since this IMS initialized.

ABENDU2990

DFSCMPX0

Explanation
The IMS Segment Edit/Compression routine DFSCMPX0 has detected an error while attempting to perform 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 routine's work area.

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.

ABENDU2991

DFSKMPX0

Explanation
The IMS Segment Edit/Compression routine DFSKMPX0 has detected an error while attempting to perform 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.

Key	Label	Description
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.

ABENDU3000

DFSICV90

Explanation
An error occurred on an XRF alternate system while processing a type 4001 or type 70 log record.

Analysis
ABENDU3000 is a standard abend issued from module DFSICV90. A DFS33xx or DFS34xx message accompanies this abend.

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 half word containing the keyed message number at offset X'2E'.

ABENDU3006

DFSUTL40

Explanation
A directory block was read from the IMS.FORMAT library that was greater than 256 bytes in length.

Analysis
ABENDU3006 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').

ABENDU3007

DFSUTL40

Explanation

The MFS Language Utility Phase 2 processor has received an unexpected completion code from the host system STOW function.

Analysis

ABENDU3007 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 and this return code is saved in DFSUTL40 within a halfword at label STOWCODE.

Code ***Meaning***

X'00' Update of the directory was completed successfully

X'04' Duplicate block—directory already contains specified name

X'08' Undefined for STOW addition—specified name could not be found

X'0C' No space left in the directory

X'10' Permanent I/O error when attempting to update the directory

X'14' The specified DCB is not open or is opened incorrectly

X'18' 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.

ABENDU3008

DFSUTL40

Explanation

The MFS Language Utility Phase 2 processor has been unsuccessful in opening one or more of the data sets represented by the following DD statements: UTPRINT, FORMAT, or SEQBLKS.

Analysis

ABENDU3008 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:

Code ***Meaning***

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.

Key	Label	Description
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.
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).

ABENDU3009

DFSUTL40

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

ABENDU3009 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 STOWBLEN=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.

ABENDU3010

DFSUTL80

Explanation

An error has been detected during construction of the ITB tree.

Analysis

ABENDU3010 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.

ABENDU3011

DFSUTLA0

Explanation

The contents of the MSG ITB in the REFERAL library are not consistent with the input message indication in the IMS.REFERAL directory entry.

Analysis

ABENDU3011 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.

ABENDU3012

DFSUTLB0

Explanation

The contents of the MSG ITB in the REFERAL library are not consistent with the input message indication in the IMS.REFERAL directory entry.

Analysis

ABENDU3012 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 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.
Reg4= Module Control Table (MCT) in use MCTETYPE=MCTETMSI MSGFLAG1 ¬= MSG1INP		

ABENDU3013

DFSUTLC0, DFSUT150

Explanation

The LPAGE, DPAGE, or literal order relocation work stack has overflowed during the processing of an output message LPAGE statement or the device input DPAGE statement.

Analysis

ABENDU3013 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.

DFSUTLC0

Analysis

ABENDU3013 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.
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.

DFSUT150

Analysis

ABENDU3013 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.

Key	Label	Description
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.

ABENDU3014

DFSUTLD0

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
ABENDU3014 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.
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.

ABENDU3015

DFSUTLE0, DFSUTLG0

Explanation
The literal work stack has exceeded its maximum size during the processing of an output message MFLD literal statement.

Analysis
ABENDU3015 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.

DFSUTLE0

Analysis

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.

DFSUTLG0

Analysis

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.

ABENDU3016

DFSUTLT0

Explanation

An internal logic error has occurred during the sort of a symbol table (PPTSYM or PPTSYM1) into alphabetic order.

Analysis

ABENDU3016 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.

ABENDU3017

DFSUTLT0

Explanation
An internal logic error has occurred during the construction of a binary structured tree for the symbol tables (PPTSYM or PPTSYM1).

Analysis
ABENDU3017 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 TREERROR. 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
User modifications to the MFS Language utility.

ABENDU3018

DFSUT0I0, DFSUT020, DFSUT120, DFSUT130

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
ABENDU3018 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.

DFSUT0I0

Analysis

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.

DFSUT020

Analysis

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.

DFSUT120

Analysis
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.

DFSUT130

Analysis
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.

ABENDU3019

DFSUT050

Explanation
The literal work stack (PPT04STK) has exceeded its maximum size during processing of a DFLD statement.

Analysis
ABENDU3019 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.

ABENDU3020

DFSUT090, DFSUT160, DFSUT180, DFSUT190, DFSUT200, DFSUT260, DFSUT280, DFSUT290, DFSUT300

Explanation
The literal work stack has exceeded its maximum size during the processing of a DFLD statement.

Analysis
ABENDU3020 is a standard abend issued from module DFSUT090, DFSUT160, DFSUT180, DFSUT190, DFSUT200, DFSUT260, DFSUT280, DFSUT290, or DFSUT300.

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.

DFSUT090

Analysis
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.

DFSUT160

Analysis
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.

DFSUT180

Analysis

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.

DFSUT190, DFSUT200, DFSUT300

Analysis

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.

DFSUT260

Analysis

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.
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.

DFSUT280

Analysis
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.

DFSUT290

Analysis
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.

ABENDU3021

DFSUT090

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
ABENDU3021 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.

ABENDU3022

(Multiple modules)

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
ABENDU3022 is a standard abend issued from the following modules: 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, and DFSUT290.

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.

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.

In all cases, the abend is issued from the abend routine at label SIZABEND. There may or may not 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.

ABENDU3025

DFSUT0I0

Explanation
An internal logic error has occurred while processing the literal work stack (PPT055STK) for TABLE description.

Analysis
ABENDU3025 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 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.

ABENDU3026

DFSUT060, DFSUT170

Explanation
An internal logic error has occurred while processing a TABLE symbol table for DFLD tablename.

Analysis
ABENDU3026 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.

DFSUT060

Analysis
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.

DFSUT170

Analysis
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.

ABENDU3027

DFSUT0I0

Explanation
An internal logic error has occurred while processing an IF label reference for a TABLE description.

Analysis
ABENDU3027 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.

ABENDU3030

DFSUPB70

Explanation
The MFS Utility Phase 1 processor has detected an invalid member in the IMS.REFERAL library. This is an internal logic error.

Analysis
ABENDU3030 is a standard abend issued from module DFSUPB70.

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 wasn't, so the abend is issued.

Possible Cause
User modification to the IMS.REFERAL Library or the MFS Language utility.

ABENDU3040

DFSRRA30, DFSRRA70, DFSRRA80

Explanation
The initialization of an IMS/VS control or batch region failed for one of the following reasons:

DFSRRA30 Unable to acquire subpool 231 storage to construct a system queue header control block or an external subsystem work area.

DFSRRA70 Unable to acquire subpool 0 storage to construct a system queue header control block for a batch region.

DFSRRA70 Unable to acquire subpool 0 storage to load the HLPI bootstrap routine for a batch region.

DFSRRA80 Unable to acquire subpool 0 storage to load the bootstrap routine for a batch utility region.

Analysis
ABENDU3040 is a standard abend issued by DFSRRA30 (control region), DFSRRA70 (batch region), or DFSRRA80.

Register 14, at the time of abend, contains the address of the location where the error was detected. This identifies which one of the possible conditions caused this abend. Register 15 contains an IMODULE GETMAIN return code indicating the exact cause of the failure. For an explanation of these return codes, see the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

Register	Label	Module
Reg14=BAL Reg15=IMODULE return code	RACTL4	DFSRRA30
Reg14=BAL Reg15=IMODULE return code	RADL13	DFSRRA70
Reg14=BAL Reg15=IMODULE return code	RADSVCI	DFSRRA70
Reg14=BAL Reg15=IMODULE return code	UTLHLP	DFSRRA80

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, then increase the region size on the job/execute statement.

ABENDU3041

DFSESI30

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
ABENDU3041 is a standard abend issued by DFSESI30.

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

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 in *IMS Version 9: Messages and Codes, Volume 1* 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'08').

Register 14 contains the address of the location where the error was detected. See contents of register 15 as described above.

Label ABEND—an unsupported return code received from an external subsystem exit routine.

Possible Cause
Refer to message DFS3603I, DFS3606I, or DFS3607I for additional information. Consult the external subsystem documentation to determine the reason for the return code as indicated in register 15.

ABENDU3042

DFSESD70, DFSESPR0, DFSVES00

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 as described below 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 in *IMS Version 9: Messages and Codes, Volume 1* or to the DFSESFC 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.

ABENDU3043

DFSIESI0

Explanation
This abend is issued when the external subsystem mother task's (TCB) end-of-task-exit routine (ETXR) is scheduled and passes an invalid daughter task (TCB) address.

Analysis
ABENDU3043 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.

ABENDU3044

DFSESPR0, DFSFESP0

Explanation
A “should not occur” situation has been discovered in communicating with an external subsystem. Message DFS3624I is sent to the master terminal indicating the module that detected the problem as well as the return code describing the problem.

Analysis
ABENDU3044 is a pseudoabend issued by DFSESPR0 or DFSFESP0.

Refer to 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 in *IMS Version 9: Messages and Codes, Volume 1* 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'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.

ABENDU3045

DFSESPR0, DFSFESP0

Explanation
This abend is 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 exit routines indicating a failure condition.

- Signon 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
ABENDU3045 is a pseudoabend issued by DFSESPR0 or DFSFESP0.

Refer to message DFS3624I in *IMS Version 9: Messages and Codes, Volume 2* for the function (FC) identifying the module that detected the error and return code (RC) identifying the error condition. Refer to the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1* 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.

ABENDU3046

DFSESPR0, DFSFESP0

Explanation
This abend is 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 exit routines indicating a failure condition.

- Signon 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
ABENDU3046 is a pseudoabend issued by DFSESPR0 and DFSFESP0.

Refer to message DFS3624I, sent to the master terminal, which specifies the function code (FC) identifying the module that detected the error and return code (RC) identifying the error condition. Refer to the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1* 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 failure condition.

Register 15 = function code (FC) and return code (RC).

Possible cause
Consult the external subsystem documentation for possible cause.

ABENDU3047

DFSESPR0

Explanation
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.

Analysis

ABENDU3047 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 in *IMS Version 9: Messages and Codes, Volume 1* 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').

See contents of register 15 as described above.

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.

ABENDU3048

DFSESPR0

Explanation

This abend is issued because the external subsystem indicated a temporary failure or a temporary resource constraint has occurred (that is, database X could not be accessed).

Analysis

ABENDU3048 is a pseudoabend issued by DFSESPRO to allow 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 DFSESPR0 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.

ABENDU3049

DFSRRA00, DFSFESP0, DFSESPRO

Explanation
One of the following conditions has been detected by IMS:

- An invalid return code was detected upon return from an external subsystem exit routine. Either the return code was not within the acceptable range valid for the exit routine (that is, greater than X'20') or, not having exceeded the range check, the return code was not supported for this exit routine.
- Another possibility is that the required data (that is, parameter list, exit addresses) did not pass validity checking.

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 DFSESGL0. IMS system service was invalid. The caller of DFSESGL0 has a logic error that probably invalidated the function.

R15 = FD The count in the parameter list passed to DFSESGL0 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).

ABENDU3050

DFSKDP00, DFSKDS00

Explanation
This abend is issued when the module that initializes IMS dispatching blocks detects an error that cannot be corrected.

Analysis
ABENDU3050 is issued for all abnormal conditions. Register 15 contains the following reason codes for both modules DFSKDP00 and DFSKDS00.

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'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.

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 the IMODULE return codes, see the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

DFSKDS00

Code ***Meaning***

X'16' The calling dispatcher was not set up for DYN SAPS. Register 2 contains the requesting dispatcher type.

X'17' The requested number of dynamic SAPs was zero. Register 2 contains the requesting dispatcher type.

X'18' DFSBCB GET for SAPS failed. Register 2 contains the requesting dispatcher type and register 3 contains the DFSBCB return code.

X'19' DFSBCB GET for the save areas failed. Register 2 contains the requesting dispatcher type and register 3 contains the DFSBCB return code.

X'1A' DFSBCB GET for SAP work areas failed. Register 2 contains the requesting dispatcher type and register 3 contains the DFSBCB return code.

X'1B' 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 of the DFSBCB return codes, see *IMS Version 9: Messages and Codes, Volume 2*.

ABENDU3051

DFSESPR0

Explanation

This abend is issued when the external subsystem not operational (SNO) exit routine passes back an X'10' return code.

IMS has allocated this abend for the external subsystem's use. It is returned by the subsystem exit routine when an IMS abend is required for debugging purposes. It is usually accompanied by an X'55' log record, written to the IMS log, and contains additional debugging information.

Analysis

ABENDU3051 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.

ABENDU3052

DFSFESP0

Explanation

This IMS abend is issued when an internal required function completed unsuccessfully.

One of the following conditions occurred:

- An invalid function code was passed to the external subsystem synchronization point manager (DFSFESP0) during an application synchronization point.
- The one byte code is passed by the IMS synchronization point manager DFSFXC30 in PST field PSTSYNFC.

Analysis

ABENDU3052 is a pseudoabend issued by DFSFESP0, code resident in the dependent region SSOB save area. This code is only executed in the event that one or both of the aforementioned 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.

DFSXLRM0

Explanation
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.

Analysis
ABENDU3052 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 information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*. 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.

ABENDU3053

DFSESPR0, DFSFESP0

Explanation
This abend is issued when either the external subsystem signon or commit prepare exit routine passes back the 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.

Analysis
ABENDU3053 is a pseudoabend issued by DFSESPR0 and DFSFESP0.

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 in *IMS Version 9: Messages and Codes, Volume 1* or DSECT DFSESFC.

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 ABENDU3053 becomes a recurring problem. ABENDU3053 should be anticipated in the event IMS is cold started frequently.

See contents of register 15 as described above for the key.

Label RCSOFA in module DFSESSO0 (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.

ABENDU3054

DFSESI40, DFSESPL0

Explanation
This abend is issued when an error is detected while attempting to release resources (that is, storage work areas).

Analysis
ABENDU3054 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 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. For a description of the IMODULE DELETE return codes, see the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1*.

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, refer to either the table in the information on IMS system services return codes in *IMS Version 9: Messages and Codes, Volume 1* 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

ABENDU3055

DFSFESP0

Explanation

This abend is the result of a failure of a phase 1 commit request to DB2 (Commit Prepare). DB2 passed back a return code of 4 to IMS. This return code (RC04) from DB2 indicates that a DB2 module has voted no to the commit request.

Analysis

ABENDU3055 is a pseudoabend issued by module DFSFESP0. The return code is passed back from DB2 to DFSESP10, which in turn passes the return code back to DFSFESP0. DFSFESP0 sets up and issues the abend.

Possible Cause

This abend is the result of a DB2 problem. Check the external subsystem either for abends, messages indicating problems, or both.

ABENDU3056

DFSD2AF0

Explanation

This abend is issued because the DB2 RRSAF interface module, DFSD2AF0, encountered a severe error in the function identified by register 15.

Analysis

ABENDU3056 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 RRSAF function.

Code	***Meaning***
X'00'	Initialization
X'01'	RRSAF Identify
X'02'	RRSAF Signon
X'03'	RRSAF Create Thread
X'04'	RRSAF Terminate Thread

ABENDU3057

DFSLI000

Explanation

This abend results when an application program issues an external subsystem request (that is, DB2 SQL call) and one or both of the following conditions occurred:

- The SSM execute parameter defining the external subsystems to IMS was not specified for the control region.
- The SSM execute parameter was specified for the dependent region and the PROCLIB member identified by the SSM parameter was null (that is, contained no entries).

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.

Analysis

ABENDU3057 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.

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.

ABENDU3058

DFSBCB00, DFSBCB30, DFSBCB60, DFSBCB90

Explanation

Modules DFSBCB00, DFSBCB30, and DFSBCB90 support get/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 invokes 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 exactly the same way that they are chained off of 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 may 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 (SCDBCBQ). Module DFSXCB00, at initialization time, obtained BXQE storage. If an element is not available, 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 ABENDU3058 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 SCDBCBQ, 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

ABENDU3058 is a standard abend issued by DFSBCB00, DFSBCB30, DFSBCB60, DFSXBC60, and DFSBCB90.

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 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 may have been used.

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 block(s). 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 invoke 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 BXQE's 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'.
	DFSBCB00 FMTBXERR	Additional storage is needed to invoke 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.

Key	Label	Description
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.

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 U3058 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's 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's 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.

Key	Description
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.

ABENDU3059

DFSSTM00

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 ABENDUU3058 for further explanation and analysis.

Analysis

ABENDU3059 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.

ABENDU3100

DFSUPAM0

Explanation

An internal error was detected in the structure of the parse table.

Analysis

ABENDU3100 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.

ABENDU3101

DFSUPAM0

Explanation
An internal logic error has occurred while trying to provide error recovery on a source statement.

Analysis
ABENDU3101 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'C0'¬=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.

ABENDU3102

DFSUPAX0

Explanation
An internal logic error has occurred while attempting to locate storage for an ITB prior to being written to the IMS.REFERAL data set.

Analysis
ABENDU3102 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.

Possible Cause
Invalid parameters were passed by the calling module.

ABENDU3103

DFSUPAD0

Explanation
An internal logic error has occurred while trying to classify the character currently being processed.

Analysis
ABENDU3103 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
Error in source statement.

VALUETBL in MFSGBL contains an incorrect translate table for source record translation. The ALPHA statement processor (DFSUPAN0) may have incorrectly modified VALUETBL.

ABENDU3104

DFSUPAD0

Explanation
An internal logic error has occurred while trying to append the source character delimiter to a stack of characters representing the item just scanned. The item stack has overflowed.

Analysis
ABENDU3104 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 DELMOVER. 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.

ABENDU3105

DFSUPAE0

Explanation
An internal logic error in the branch vector table has been detected, after finding a continuation character.

Analysis
ABENDU3105 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.

ABENDU3107

DFSUPAX0

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
ABENDU3107 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' Reg3¬=address at label STGLOC	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.

ABENDU3108

DFSUPAX0

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
ABENDU3108 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.

ABENDU3109

DFSUPAX0

Explanation
An internal logic error occurred while attempting to print an ITB.

Analysis
ABENDU3109 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.

ABENDU3110

DFSUPB60

Explanation
An internal logic error occurred while processing a WRITE call of an ITB to the IMS.REFERAL data set.

Analysis
ABENDU3110 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.

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 ITBTYPE¬=¬=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.

ABENDU3111

DFSUPAZ0, DFSUPA10, DFSUPA30

Explanation
An internal logic error has occurred while the MFS preprocessor was analyzing the device type in the DEVITB (DEVOCCR4). The device code did not match any valid device type.

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.

ABENDU3115

DFSUPAF0, DFSUPBH0, DFSUPBJ0, DFSUPBK0

Explanation
An internal logic error occurred during the stacking or unstacking of records for the MFS preprocessor source output.

Analysis
ABENDU3115 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.

DFSUPAF0

Analysis
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.

DFSUPBH0, DFSUPBK0

Analysis
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 UABENDD1.
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 UABENDD2.

DFSUPBJ0

Analysis

Use the registers in the abend SVRB 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'	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 UABENDD1.
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 UABENDD2.
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 UABENDD3.

ABENDU3116

DFSUPAZ0

Explanation

An internal logic error occurred in a main storage exchange sort routine used by the MFS Preprocessor.

Analysis

ABENDU3116 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.

ABENDU3120

DFSRLP00

Explanation

During IMS restart, module DFSCMR00 is needed to reinstate the MSC control blocks but it is not available.

ABENDU3141

DFSRLP00

Explanation

An error was detected while reading the restart data set during a warm or 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 Causes

- An I/O error occurred on the restart data set.
- Extended recovery facility (XRF) specifications cannot be changed in this restart. (See message DFS3851I.)
- The alternate system is not the active system. (See message DFS3852I.)
- The IMSID name does not match the checkpoint RSE name. (See message DFS3868I.)
- The HSBID for the XRF active and alternate systems cannot be equal. (See message DFS3894I.)

ABENDU3265

DFSULG20

Explanation

DFSFDLW0 tries to find the log blocks following the last log block in the OLDS input, but the WADS(s) had an I/O error or more than one I/O error.

Analysis

ABENDU3265 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:

Code	***Meaning***
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:

Code	***Meaning***
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 - DECB address	LTG210	DFSFDLW0 issues a BSAM READ for the primary DCB while trying to locate the last used WADS. DECB 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'.
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' SEGTABA - 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'.

Key	Label	Description
	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 - DECB address	LTJ220 LTJ230	DFSFDLW0 issues a BSAM READ for the WADS DCB while rebuilding an OLDS block. DECB 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 - DECB 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'.

ABENDU3271

DFSULG10

Explanation

An end-of-file was encountered in the input OLDS, but the closing point had not yet been found.

Analysis

ABENDU3271 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.

ABENDU3272

DFSULG30

Explanation
DFSULG30 has detected an empty PSTTBL entry.

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#		

ABENDU3274

DFSUARC0, DFSULG10, DFSULG20

Explanation
The Log Archive or Recovery utility received an error return code from a DBRC exit routine.

Analysis
ABENDU3274 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'12'	ARCP10 (DFSUARC0)	An invalid volume stop time specified.
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'12'	DINI010 (DFSULG10)	DSPINIT macro failure or DSPRCLOS macro failure.
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.

Key	Label	Description
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

Key	Label	Description
Reg15= X'48'	DINI010 (DFSULG10)	A required parameter was not specified on the exit routine invocation.

ABENDU3275

DBFDBAU0, DFSDBAU0, DFSDBDR0, DFSDLOC0, DFSPCCC0, DFSDDUI0, DBFMLCL0, DBFMLOP0, DBFPICT0

Explanation

A logic error occurred in either IMS code or DBRC code. The following codes in register 15 give the reason for the abend.

Code ***Meaning***

X'03' A database authorization call was made to DBRC. A return code of X'0C', which indicated that the subsystem was not signed on to DBRC, was received upon return. This condition should not have occurred unless there was an error in the IMS subsystem initialization or in the DBRC. (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. (DBFPICT0)

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. (DBFPICT0)

When ABENDU3275 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 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).

(DBFARD30, DBFARD40, DBFSTAP0, DBFHDEP0, DBFMLCL0) When DBRC returns non-zero return code for UNAUTH request, DFS0019I and DSP0230I are written. When the return code>X'10', ABENDU3275 is issued. The return code of DBRC is written in *IMS Version 9: Messages and Codes, Volume 2*.

Analysis

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 *IMS Version 9: Messages and Codes, Volume 2* 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	DFSDDUI0

Possible Cause

A breakdown occurred in communication between IMS and DBRC. The status information may have been lost on the RECON data set.

ABENDU3276

DBFFORI0, DBFVXOE0, DBFVXOW0, DBFXCNX0

Explanation
While performing IO processing, Fast Path detected the inflight IO count (DMACWTCT) to be invalid. Its value is already zero or negative prior to be decrement.

Analysis
ABENDU3276 is a standard abend issued by:

- DBFFORI0: 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.

ABENDU3287

DFSDLA00

Explanation
The application program was a pseudoabend 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.

ABENDU3290

DFSFXC50

Explanation
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
ABENDU3290 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. Refer to the description of the UNLOCK request in the information about IRLM request return and reason codes in *IMS Version 9: Messages and Codes, Volume 2* to determine the cause of the failure.

ABENDU3300

DFSLRH00, DBFLRH00, DBFNOTM0

Explanation

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.

Refer to the description of the LOCK request in the information about IRLM request return and reason codes in *IMS Version 9: Messages and Codes, Volume 2* to determine the cause of the failure.

Analysis

ABENDU3300 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

Return Code	Reason Code	Description
X'08'	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.

ABENDU3302

DFSLRH00, DBFLRH00, DBFNOTM0

Explanation

An invalid condition was detected by the IRLM on a LOCK, UNLOCK, or NOTIFY request.

Analysis

ABENDU3302 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'.

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
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.

ABENDU3303

DFSLRH00, DBFLRH00, DFSDLA00, DBFDBAC0, DBFHGU10, DBFIRC10, DFSDVSM0, DFSDBH20, DFSDBLM0, DFXES0

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 subsystem. Message DFS3304I, issued in conjunction with this abend, describes the rejected lock request.
2. The IRLM has entered IRLM FAILED or COMM FAILED state. In order to ensure database integrity before allowing the IRLM to enter the INITIAL state, it was necessary for IMS to terminate the application.
3. An online IMS control region terminated all applications using data bases registered to DBRC at SHRLEVEL 2 or 3 when the IRLM terminated abnormally.
4. An application program tried to use a database that was not available either for access, update, or both. Message DFS3303I, issued in conjunction with this condition, gives the reason for the abend.
5. An application program received incorrect input data (for example, when an application program issues a ROLS call in error).
6. A coupling facility connection or structure failure occurred during a read-and-register operation to the coupling facility.
7. A recoverable in-doubt structure (RIS) was established in a CCTL-IMS connection. Message DFS0693I displays the PSB and the recovery token.
8. A batch job was running when a structured rebuild occurred. The batch job is then terminated.

DBFDBAC0, DBFHGU10, DBFIRC10

Explanation

1. ABENDU3303 was a normal occurrence following a /START, /STOP. or /DBRECOVERY database command for a full-function database.
2. DBFDBAC0 stopped all Fast Path IFP dependent regions whose PSB had at least one PCB that references the full-function (DL/I) database. These dependent regions were queued on a load balancing group (LBG).
3. When the /START command was entered, these IFP dependent regions were terminated with ABENDU3303, then rescheduled. The IFP was then able to pick up the change in status of the database.
4. When the /STOP or /DBRECOVERY command was entered, the IFP might not have been terminated and then rescheduled, depending on whether an INIT was issued for the application.
5. DBFDBAC0 set the flag, DBFHGU10 performed the setup, and DBFIRC10 issued the abend.
6. Message DFS3303I was not issued, and there were no type 67FF log records.

To determine the cause for failure 1, 2 or 3 listed above, refer to the description of the LOCK request in the information about IRLM request return and reason codes in *IMS Version 9: Messages and Codes, Volume 2* to determine the cause of the failure.

Analysis

This is a pseudoabend issued by DFSLRH00 and DBFLRH00.

If the application is 'Q' driven, it is abnormally terminated and the transaction is suspended from processing until the failed IMS subsystem is recovered.

Possible Cause

An IMS sharing subsystem has failed or the IRLM has failed.

ABENDU3305

DFSSTAX0

Explanation

This abend can occur when:

- 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.

Analysis
ABENDU3305 is a standard abend issued out of DFSSTAX0.

ABENDU3306

DFSLMGR0

Explanation
ABENDU3306 is issued to prevent attempts by non-APF authorized programs from gaining a PSW supervisor state, thus violating z/OS integrity. The IRPM pointer was found to be invalid. The key 8 IRPM does not point to its paired key 7 IRPM. The paired key 7/key 8 IRPMs are formatted to point to each other during SVC initialization. Outside of attempts to use IMS's PC number, an abend can occur here because of storage overlays by faulty programs. Either the PSTIRLMA pointer in the PST that points to IRLM RLPL storage within the IRPM block is overlaid, or the key 8 IRPM storage has been walked on.

ABENDU3312

DFSUDMP0

Explanation
An error occurred during DBRC processing. 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.

ABENDU3314

DFSPCC40

Explanation

An unrecoverable error was detected by module DFSPCC40 during processing required because changed data user exit routines or changed data logging was defined for the database, or the changed data user exit routine returned an ABEND return code. The reason code for the abend, on the z/OS message and in register 15, contains:

The first byte identifies the module detecting the error as follows:

04 DFSPCC40

The second byte identifies the routine detecting the error as follows:

00 SETUP

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

The third byte is a return code for the error.

04 The user exit routine defined for the segment was not found. Message DFS3311I contains the user exit name.

08 Storage Obtain or Release error.

12 Exit routine LOAD error.

16 Exit routine DELETE error.

20 Exit routine return code requests ABEND.

24 Invalid return code from exit routine (neither a multiple of 4 nor greater than 20).

The fourth byte contains a reason code for the error based on the return codes in the third byte as follows:

Return Code	*Reason Code*
04	00 - no additional qualification.
08	The low byte of the return code passed back by either a z/OS GETMAIN or FREEMAIN (see GETMAIN and FREEMAIN return codes).
12	00 - no additional qualification.
16	The return code passed back by z/OS on a DELETE of module request (see z/OS DELETE return codes).
20	Any value set up by a user exit routine. The value is user-defined.
24	Any value that is neither a multiple of 4 nor greater than 20.

ABENDU3315

DFSZLDU0

Explanation

The Hardware Data Compression Dictionary utility (DFSZLDU0) has detected an error while attempting to build or validate the dictionary.

The program status word (PSW) points to the instruction within the module from which 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, or HDCDOUT.
Reg15=6		A logic error has been detected in the statistics/data integrity module during data compression or expansion.

ABENDU3325

DFSZLDX0

Explanation

The Hardware Data Compression exit routine has 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.

Key	Label	Description
Reg14=D3C4E702 Reg6=length plus offset of sequence field	D3C4E702	The segment's 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.
Reg14=D3C4E706 Reg2=dictionary address	D3C4E706	The HDC Dictionary is not on a page boundary. The compression/expansion exit routine did not specify 'page DFSZHDCD' on its linkage step.
Reg14=D3C4E707	D3C4E707	Invalid compression/expansion routine. Eyecatcher was not DFSZHDCD. Examine the compression/expansion link step.
Reg14=D3C4E708 Reg3=CVT address	D3C4E708	Hardware Data Compression services are not available.
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.

ABENDU3399

DBFLBEU0

Explanation

Each time an RSR-covered, shareable DEDB area or FF database is allocated by DBRC, the counter SCDRSR_CVCNT increases by 1. Each time an RSR_covered, shareable DEDB area or FF database is deallocated by DBRC, the counter decreases by 1.

If the SCDRSR_CVCNT counter equals zero before an RSR-covered shareable DEDB is deallocated by DBRC, the IMS control region returns user abend 3399. The count must be positive before a deallocation. The count can only reach zero after the last remaining RSR-covered, shareable DEDB area or FF database is deallocated by DBRC.

Analysis

At the time of the abend, register 7 contains the SCDRSR_CVCNT value of zero. This error is indicative of an IMS internal logic error.

ABENDU3400

DFS0AER

Explanation
In general, this abend is requested by a sample, or user-written, application program for one of the following reasons.

- The operator requested an abend in response to message DFS3125A.
- The (sample) application program requested an abend by calling DFS0AER as a result of checking status codes.
- DFS0AER was invoked more than 20 times during the execution of the application program.

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.

ABENDU3411

DFSIIOC0

Explanation
IMS encountered an error trying to open the data set with the ddname of MODSTAT.

Analysis
ABENDU3410 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.

ABENDU3412

DFSIIOC0

Explanation
An I/O error was detected while attempting to read the MODSTAT data set.

Analysis
ABENDU3412 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.

ABENDU3413

DFSIIOC0

Explanation
The MODSTAT data set contains an invalid ddname.

Analysis
ABENDU3413 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.

ABENDU3414

DFSIIOC0

Explanation
IMS was unable to allocate storage.

Analysis
ABENDU3414 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.

ABENDU3415

DFSIIOC0

Explanation
IMS was unable to ENQ on the data set associated with the data set described in the ddname of message DFS3415X.

Analysis
ABENDU3415 is a standard abend issued by DFSIIOC0. To overcome this problem, wait until the other task has released the library and execute IMS.

ABENDU3416

DFSIINM0

Explanation
IMS encountered an error trying to open the data set referred to in 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
ABENDU3416 is a standard abend issued by DFSIINM0. To overcome this problem, either provide the missing DD statement, or create the data set referenced by the specified DD statement; then execute IMS.

ABENDU3417

DFSIINM0, DFSTMII0

Explanation
IMS was unable to compute the length of the DL/I control block's modules.

Analysis
ABENDU3417 is a standard abend issued by DFSIINM0 and DFSTMII0.

Possible Cause
The IMODULE LOCATE function failed for the control block's module named in message DFS3417.

ABENDU3418

DFSIINM0, DFSTMII0

Explanation
IMS was unable to delete the loaded DL/I control block's modules at initialization time.

Analysis
ABENDU3418 is a standard abend issued by DFSIINM0 and DFSTMII0.

Possible Cause
The IMODULE DELETE function failed for the control block's module named in message DFS3418.

ABENDU3419

DFSIINM0, DFSTMII0

Explanation
IMS was unable to load the DL/I control block modules. Message DFS3419 names the module.

Analysis
ABENDU3419 is a standard abend issued by DFSIINM0 and DFSTMII0.

Possible Cause
There was not enough storage available in the control region or the active MODBLKSA or MODBLKSB data set was not APF-authorized.

ABENDU3420

DFSQRST0

Explanation
IMS detected existing messages for a transaction that does not exist. The transaction definition does not match the message activity indicated on the IMS system log.

Analysis
ABENDU3420 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.

ABENDU3421

DBFCPRC0, DFSCPSM0, DFSIINM0, DFSTMII0

Explanation
IMS detected inconsistent control blocks during the IMS initialization or during the /MODIFY COMMIT command.

Analysis
ABENDU3421 is a standard abend issued by DFSIINM0 or DFSTMII0 at IMS initialization, and by DBFCPRC0 or DFSCPSM0 during /MODIFY COMMIT. The DFSSMB0x or DFSRCTEx control block modules are inconsistent with the DFSPDIRx or DFSRCTEx control block modules. PSBs or return codes are referenced that do not exist. Message DFS3421X names the control block module and the control block, which could not be found.

Possible Cause
Check for errors in the IMS system definition stage 1 source statements for the PSBs or return codes referenced in message DFS3421. Check that IMS.MODBLKSA(B) (active MODBLKS data set) contains the proper set of control block modules, DFSSMB0x, DFSPDIRx, DFSDDIRx or DFSRCTEx where x is the nucleus suffix.

If the above checks indicate that no problems exist, and the abend is issued during initialization of module DFSIINM0, then the following conditionally assembled modules (Fast Path dependencies) may be incorrectly assembled or the wrong level of code for macro DFSFP in IMS.OPTIONS was used for modules, DFSCTB20, DFSCBT30, DFSCBT40, or DFSIINM0. the control blocks that represent transaction codes.

DFSTMII0

Analysis
If XXXX in message DFS3421X = PDIR or RCTE, then the contents of the following registers at abend are:

Reg3 PSB name address for PDIR, or routing code name address for RCTE.

Reg6 SMB address.

Reg15 Return code from DFSCBTS FIND Service.

If XXXX in message DFS3421X = HSMB, then the contents of the following registers at abend are:

Reg6 SMB address.

Reg15 Is one of the following reason codes:

Code	***Meaning***
X'20'	The SMB to be hashed was already in the SMB Hash Table.
X'24'	A chain in the SMB Hash Table was broken.
X'32'	The hashing routine (DFSFHSH0) or the SMB Hash Table was not in the system.

ABENDU3422

DFSIINM0, DFSTMII0, DFSCPDM0

Explanation
IMS did not acquire enough storage to move the loaded control blocks (PDIR, DDIR, SMB, or RCTE) to IPAGE storage in subpool 231 (CSA).

Analysis
ABENDU3422 is a standard abend issued by module DFSIINM0. Message DFS3422X names the type of control block for which no storage was available. Register 2 contains the return code from the DFSBCB FUNC=GET call. Register 7 contains the number of blocks requested.

This abend is:

- Issued by DFSTMII0 when DFSBCB GET fails for an SMB or the IMODULE GETMAIN for the SMB hash table fails
- Issued if no more entries can be added to the SMB hash table or if a duplicate entry from SMB was found.
- Issued by DFSCPDM0 during MODIFY processing.

For an IMODULE GETMAIN and for a DFSBCB GET failure, register 15 contains the return code. Register 15 contains the following subcodes to indicate the failure:

Code	*Meaning*
X'20'	Duplicate SMB in hash table.
X'24'	SMB hash table has a chain error.

Possible Cause
One possible cause is that there was not enough storage available for the IMS control region to increase the storage and execute IMS. Another possible cause is that there was a control block mismatch (in other words the wrong version of IMS was used to perform a mod blocks system definition).

ABENDU3476

DFSAINB0

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.

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.

ABENDU3477

DFSAINB0

Explanation
The Transaction Manager initialization detected that the primary master logical terminal (PTMO) is not defined in the system. IMS requires that the PMTO is properly specified in the system definition.

Analysis

ABENDU3477 is a standard abend issued by DFSAINB0. Verify your system definition to ensure that the PTMO is properly specified.

ABENDU3498

DFSDLR00, DFSDDLE0, DFSDXMT0, DFSDLDC0

Explanation

User environment errors detected during DL/I processing of a HALDB result in pseudoabend ABENDU3498.

Analysis

ABENDU3498 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:

Code	*Meaning*
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.

ABENDU3610

DFSKBDP0 (Batch IMS only)

Explanation

An error occurred in the IMS batch dispatcher.

Analysis

ABENDU3610 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*
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.

ABENDU4095

DFSFCST0, DFSFMOD0

Explanation

This code indicates that an IMS TCB is being terminated abnormally because a different IMS TCB in the control region has abended. The initial abend contains the abend code that describes the problem.

ABENDU4095 is normal under these circumstances. The abend code for the original abend can be found in register 15 at the time of abnormal termination.

Notices

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

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

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

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

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

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

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

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

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

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

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

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

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

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates 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. 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 IBM's application programming interfaces.

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_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming Interface Information

This book is intended to help system programmers diagnose IMS problems. This book documents information provided by IBM that is used for diagnosis, modification, or tuning.

Attention: Do not use this diagnosis, modification, or tuning information as a programming interface.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both.

BookManager
CICS
Database 2
DB2
DFSMSdss
DFSMS
Hiperspace
IBM
IMS
Language Environment
RACF
RETAIN
VTAM
z/OS

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product or service names may be trademarks or service marks of others.

Bibliography

This bibliography lists all of the information in the IMS Version 9 library.

- *z/OS MVS Programming: Assembler Services Reference, Volume 1 (ABEND-HSPSERV)*, SA22–7606
- *z/OS MVS System Messages, Volume 1*, SA22–7631
- *z/OS DFSMS Macro Instructions for Data Sets*, SC26-7408
- *z/OS Language Environment Programming Guide*, SA22-7561

IMS Version 9 Library

Title	Acronym	Order number
IMS Version 9: Administration Guide: Database Manager	ADB	SC18-7806
IMS Version 9: Administration Guide: System	AS	SC18-7807
IMS Version 9: Administration Guide: Transaction Manager	ATM	SC18-7808
IMS Version 9: Application Programming: Database Manager	APDB	SC18-7809
IMS Version 9: Application Programming: Design Guide	APDG	SC18-7810
IMS Version 9: Application Programming: EXEC DLI Commands for CICS and IMS	APCICS	SC18-7811
IMS Version 9: Application Programming: Transaction Manager	APTM	SC18-7812
IMS Version 9: Base Primitive Environment Guide and Reference	BPE	SC18-7813
IMS Version 9: Command Reference	CR	SC18-7814
IMS Version 9: Common Queue Server Guide and Reference	CQS	SC18-7815
IMS Version 9: Common Service Layer Guide and Reference	CSL	SC18-7816
IMS Version 9: Customization Guide	CG	SC18-7817
IMS Version 9: Database Recovery Control (DBRC) Guide and Reference	DBRC	SC18-7818
IMS Version 9: Diagnosis Guide and Reference	DGR	LY37-3203
IMS Version 9: Failure Analysis Structure Tables (FAST) for Dump Analysis	FAST	LY37-3204
IMS Version 9: IMS Connect Guide and Reference	CT	SC18-9287
IMS Version 9: IMS Java Guide and Reference	JGR	SC18-7821
IMS Version 9: Installation Volume 1: Installation Verification	IIV	GC18-7822
IMS Version 9: Installation Volume 2: System Definition and Tailoring	ISDT	GC18-7823
IMS Version 9: Master Index and Glossary	MIG	SC18-7826
IMS Version 9: Messages and Codes, Volume 1	MC1	GC18-7827
IMS Version 9: Messages and Codes, Volume 2	MC2	GC18-7828
IMS Version 9: Open Transaction Manager Access Guide and Reference	OTMA	SC18-7829
IMS Version 9: Operations Guide	OG	SC18-7830
IMS Version 9: Release Planning Guide	RPG	GC17-7831
IMS Version 9: Summary of Operator Commands	SOC	SC18-7832
IMS Version 9: Utilities Reference: Database and Transaction Manager	URDBTM	SC18-7833
IMS Version 9: Utilities Reference: System	URS	SC18-7834

Supplementary Publications

Title	Order number
IMS Connector for Java 2.2.2 and 9.1.0.1 Online Documentation for WebSphere Studio Application Developer Integration Edition 5.1.1	SC09-7869
IMS Version 9 Fact Sheet	GC18-7697
IMS Version 9: Licensed Program Specifications	GC18-7825

Publication Collections

Title	Format	Order number
IMS Version 9 Softcopy Library	CD	LK3T-7213

Title	Format	Order number
IMS Favorites	CD	LK3T-7144
Licensed Bill of Forms (LBOF): IMS Version 9 Hardcopy and Softcopy Library	Hardcopy and CD	LBOF-7789
Unlicensed Bill of Forms (SBOF): IMS Version 9 Unlicensed Hardcopy Library	Hardcopy	SBOF-7790
OS/390 Collection	CD	SK2T-6700
z/OS Software Products Collection	CD	SK3T-4270
z/OS and Software Products DVD Collection	DVD	SK3T-4271

Accessibility Titles Cited in This Library

Title	Order number
z/OS V1R1.0 TSO Primer	SA22-7787
z/OS V1R5.0 TSO/E User's Guide	SA22-7794
z/OS V1R5.0 ISPF User's Guide, Volume 1	SC34-4822

Program Number: 5655-J38

Printed in USA

LY37-3204-00

Spine information:

IMS

Failure Analysis Structure Tables (FAST) for Dump Analysis

Version 9