z/OS

DFSMSdfp Diagnosis Reference

GY27-7618-02

z/OS

DFSMSdfp Diagnosis Reference

GY27-7618-02

Second Edition, March 2002

This edition applies to Version 1 Release 3 of z/OS™ (5694-A01) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces GY27-7618-01.

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

IBM welcomes your comments. A form for readers' comments may be provided at the back of this publication, or you may address your comments to the following address:

International Business Machines Corporation
RCF Processing, Department M86/050
5600 Cottle Road
San Jose, CA 95193-0001
United States of America

IBMLINK from US: STARPUBS at SJEVM5
IBMLINK from Canada: STARPUBS at TORIBM
IBM Mail Exchange: USIB3VVD at IBMMAIL
Internet: starpubs@us.ibm.com

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

Make sure to include the following in your comment or note:

- Title and order number of this book
- Page number or topic related to your comment

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

© Copyright International Business Machines Corporation 1979, 2002. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures ix

Tables xi

About This Book xv
References to CICS®. xv
Required Product Knowledge. xv
Referenced Publications xv
Accessing z/OS DFSMS Books on the Internet xvii
Using LookAt to look up message explanations. xvii
Accessing Licensed Books on the Web xvii
How to Send Your Comments xviii

Summary of Changes xix
Summary of Changes for GY27-7618-02 z/OS Version 1 Release 3 xix
New Information xix
Deleted Information xix
Changed Information xix

Chapter 1. General Diagnostic Aids 1
Notational Conventions 1
Types of System Dumps 2

Chapter 2. AOM Diagnostic Aids 3
Using IPCS to Format the AOM Trace Table 3
CTRACE Subcommand 4
AOM Trace Table Abbreviations 4
AOM Service Return and Reason Codes. 7
AOM Tape Library Return and Reason Codes 11

Chapter 3. BDAM Diagnostic Aids 13
Messages and Codes Issued by BDAM Modules 13

Chapter 4. Compression Management Services Diagnostic Aids 15
Compression Services 15
Compression Services Messages 15
LOGREC Information 16
CMP Activation Return and Reason Codes 17
CMP Allocation Return and Reason Codes 18
CMP Return and Reason Codes 18

Chapter 5. DADSM/CVAF Diagnostic Aids 27
CVAF Module-to-Function Cross-Reference 27
CVAF GTF Trace of Events 27
CVAF Format of GTF Trace Records. 29
CVAF GTF Trace Events 29
CVAF Debug Trap Options 33
CVAF Debug Example 1 36
CVAF Debug Example 2 37
CVAF Debug Example 3 37
CVAF Debug Example 4 37
CVAF Debug Example 5 37
CVAF—VTOC Index Error Message 38

© Copyright IBM Corp. 1979, 2002

Explanation 38
System Action 38
Programmer Response 38
CVAF Dumps 39
CVSTAT Field Codes 40
Format-7 DSCB 45
DADSM Module-to-Function Cross-Reference 45
DADSM Return and Diagnostic Codes 47
DADSM CREATE (Allocate) Return and Diagnostic Codes 47
DADSM Extend to New Volume Function Return and Diagnostic Codes . . . 65
DADSM Extend to Same Volume Diagnostic Codes 65
DADSM Extend Return Codes 68
DADSM ISJF Function Return and Diagnostic Codes 70
DADSM LSPACE Return and Diagnostic Codes 71
DADSM OBTAIN Function Return Codes 73
DADSM Partial Release Return and Diagnostic Codes 74
DADSM RENAME Return and Diagnostic Codes 82
Status Codes from RENAME. 90
DADSM SCRATCH Return and Diagnostic Codes 91
Status Codes from SCRATCH 99
DADSM UPDATE Return and Diagnostic Codes 101
DADSM Common Functions 103
DADSM Dumps 105

Chapter 6. DASD Common Services Diagnostic Aids 107
DASD Common Services Error Recovery and Detection 107
DASD Common Services—Problem Determination Area (PDA). 107
DASD Common Services Return and Reason Codes 108
Common Filter Services Return and Reason Codes 108
DASD Calculation Services Return, Reason, and Status Codes 109
Device Information Services Return, Reason, and Response Codes 110

Chapter 7. DESERV Macro Diagnostic Aids 113
Return Code DESRC_SEVE 113
Reason Codes returned by the DESERV macro 113
DESERV ALL Function Reason Codes. 113
DESERV GET Function Reason Codes 114
DESERV PUT Function Reason Codes 114
DESERV GET_ALL Function Reason Codes 114
DESERV GET_NAMES Function Reason Codes 115
DESERV RELEASE Function Reason Codes 115

Chapter 8. Distributed FileManager/MVS Diagnostic Aids 117
Distributed FileManager/MVS Errors. 117
Basic Error Processing 117
Resource Problems. 118
System Logic Errors 118
DDM Data Stream Errors. 119
Unexpected Source Termination 119
Communication Errors. 119
Distributed FileManager/MVS Messages 120
Distributed FileManager/MVS System Messages 120
DDM Reply Messages Returned by Distributed FileManager/MVS 125
Distributed FileManager/MVS Problem Determination Tools 127
Trace Facility for Distributed FileManager/MVS 127
TRACE Command for DFM. 128

DFM Trace Options 129
Formatting the DFM Trace Table 129
IPCS Dump Formatting with SMSXDATA for DFM 130
Formatting LOGREC Records 132
Slip Trapping for a Specific Reply Message 132
GTF Tracing of Input/Output Buffers. 134
Distributed FileManager/MVS SRVDGN Data Items 135
Distributed FileManager/MVS Implementation Specific Codes 136
Distributed FileManager/MVS General Error Codes 137
Distributed FileManager/MVS Communications Error Codes 139
Distributed FileManager/MVS Agent Error Codes 140
Distributed FileManager/MVS File/Directory Error Codes 141
Distributed FileManager/MVS Locking Error Codes 141
Distributed FileManager/MVS Access Method Error Codes 141

Chapter 9. ISMF Diagnostic Aids 143
ISMF Execution Options 143
Determining How ISMF Processes Abends and Generates Dumps 143
Using the Logging/Abend Control Entry Panel 144
Using the ERTB Display Panel 145
ISMF's Use of the ISPF Log 148
ISMF Return Codes 151
ISMF Reason Codes 151

Chapter 10. Media Manager Diagnostic Aids 187
A Sample Summary Dump 187
Media Manager Return Codes. 190

Chapter 11. OAM Diagnostic Aids. 195
Retrieving Information from the OAM DB2 Databases for Diagnosis 196
OSREQ Return and Reason Codes. 196
OAM Macro Return and Reason Codes 206
LCS External Services (CBRXLCS) 212
CBRXLCS Return and Reason Codes 212
Library Automation Communication Services (LACS) Return and Reason Codes 218
OAM Records Written to SYS1.LOGREC. 234
Software Error Records 234
Hardware Error Records 234
OAM Dump. 241
OAM IPCS Processing 242
Diagnosing DB2 Problems 244
Call Attachment Facility (CAF) Trace 245
DB2 Trace 245
DB2 Log Data. 245
Asynchronous Adapter Device Driver 245
SCSI Adapter Device Driver. 246
SCSI Adapter Completion Code 246

Chapter 12. OPEN/CLOSE/EOV (Common) Diagnostic Aids. 247
Using the Transfer Control Table 247
Locating an Object Module in a Storage Dump. 247
Adding a Temporary Change in the Maintenance Patch Area of an O/C/EOV Module 248
Using Error Records for Debugging 248
Abend Error Recording Initiated by O/C/EOV 248

Abend Error Recording by Problem Determination Routines 249
Abend Error Recording from the O/C/EOV System Recovery Routines 250
Abend Error Recording in SYS1.LOGREC 253
Problem Determination Optional Work Area Trace 255
DCB for EXCP, SAM, and BPAM 258
Cross-Reference 274
DCB for ISAM 283
Cross-Reference 292
DCB for BDAM 298
Cross-Reference 305
DCB for BTAM and QTAM 309
Cross-Reference 316
DCB for TCAM 320
Cross-Reference 328
DCB for GAM 331
Cross-Reference 334
IOB 335
Cross-Reference 345
DEB 350
Cross-Reference 365

Chapter 13. HFS Diagnostic Aids 375
HFS Component Trace 375
TRACE Command for HFS Component Trace 375
HFS Trace Options 376
HFS CTRACE Writer Options 377
Formatting the HFS Component Trace 381
IPCS VERBEXIT SMSXDATA Dump Formatting 381
HFS Records Written to SYS1.LOGREC 382
Out-of-Space During Sync Processing 388
HFS Return and Reason Codes 388
HFS Common Return Codes 389
HFS Common Reason Codes 390
IMF Reason Codes 390
SSF Reason Codes 390

Chapter 14. PDSE Diagnostic Aids 391
Operator Commands for Analyzing and Repairing PDSE Problems 391
Definitions 391
ANALYSIS Command 392
FREELATCH Command 394
Recommended Usage 395
PDSE Component Trace 398
TRACE Command for PDSE Component Trace 398
PDSE Trace Options 399
PDSE CTRACE Writer Options 400
Formatting the PDSE Component Trace 403
IPCS VERBEXIT SMSXDATA Dump Formatting 403
PDSE Records Written to SYS1.LOGREC 405
PDSE Return and Reason Codes 406
PDSE Common Return Codes. 407
PDSE Common Reason Codes 407
BMF Reason Codes 408
CDM Reason Codes 408
CLM Reason Codes 410
FAMS Reason Codes 410

IMF Reason Codes 411
SSF Reason Codes. 411

Chapter 15. SAM Diagnostic Aids 413
Extended Format I/O Routine Diagnostic Information 413
Interrupt Control Block (ICB) 416
Additional SYNAD Information (BSAM and QSAM) 417

Chapter 16. Storage Management Subsystem Diagnostic Aids 419
Storage Management Subsystem Message-Indicated Failure Source Identification 419
Storage Management Subsystem IPCS Dump Formatter (SMSDATA) 420
SMSDATA Verb Exit Functions. 420
Validating Control Block Chains 420
Formatting Control Blocks 420
Invoking the Storage Management Subsystem IPCS Dump Formatter (SMSDATA) 422
SMSDATA Formatting Examples 425
Storage Management Subsystem Trace Facility 431
The Storage Management Subsystem Address Space Trace Table 431
Trace Event Options 431
Storage Management Subsystem Records Written to SYS1.LOGREC 432
Locating the Intermodule Communications Area (IMC) 435
Storage Management Subsystem Storage Dump Contents 435
Sample of a Storage Management Subsystem SUMDUMP 435
Storage Management Subsystem Return and Reason Codes 437
ACS Translator Return Codes 437
ACS Executor Return Codes 438
Storage Management Subsystem Initialization Return Codes 438
Storage Management Subsystem Interface Return Codes 439
Storage Management Subsystem Reason Codes 439

Chapter 17. Utility Service Diagnostic Aids 465
IEBCOPY (PDSE Version) 465
Maintenance Level Message 465
Messages from DFSMSdfp Subcomponents. 465
WTO/WTP of Critical Messages 465
A SNAP/TRACE Data Set 466
Debug Keywords in the PARM Field 466
Extension of Normal Messages with Additional Information 467
Additional Debugging Messages 468
User Abend after Certain Errors 468
Abend after a Disastrous Error 468
Consolidated Data Set Information in Dumps 468
Reversion to Channel Programs That Are Not ECKD (ESCON Support) 468

Chapter 18. VIO Diagnostic Aids 469
Analyzing Diagnostic Information Recorded by Recovery Routines 469
Analyzing SYS1.DUMP Data Sets 469
Analyzing the SYS1.LOGREC Data Set 469
Messages Issued by VIO. 471

Chapter 19. VSAM Diagnostic Aids 473
Access Method Services (AMS) Diagnostic Aids 473
Trace Tables 473
Dump Points 474

TEST Option 474
Catalog Management Diagnostic Aids 477
Sample DIAGNOSE Output 477
VSAM OPEN/CLOSE/End-of-Volume (O/C/EOV) Diagnostic Aids 479
Generalized Trace Facility 479
Interactive Problem Control System (IPCS) 481
VSAM OPEN/CLOSE/End-of-Volume Return and Reason Codes 482
VSAM Record-Level Sharing (RLS) Diagnostic Aids 482
VSAM RLS Component Trace 483
SMSVSAM Abends 485
Activating the SMSVSAM Address Space 486
Terminating the SMSVSAM Address Space 486
Console Dumps 486
SMSVSAM Initialization Errors 486
Sharing Control Problems 487
VSAM RLS Hang Conditions 488
VSAM RLS Deadlock and Timeout Problems 488
VSAM Record-Level Sharing Return and Reason Codes 488
Return Codes from SMSVSAM 488
Return Codes from SMPM_CFPurge 491
Return Codes from SMPM_CFQuery 491
VSAM Record Management (R/M) Diagnostic Aids 492
Control Block Information 492
UPAD Exit or WAITX 492
RLSWAIT Exit 493
Error Conditions 493
Exclusive Control Error Analysis 494
ABEND0CX Error Analysis 495
Damaged Data Sets (Non-RLS Access) 495
Damaged Indexes 496
Damaged Data Control Intervals 497
Physical I/O Errors 497
VSAM Record Management Trace Facility (Non-RLS Access) 498
VSAM Record Management Return and Reason Codes 504
Return Codes from the Record-Management (Request) Macros 504
Control Block Manipulation Return Codes 516

Appendix. Accessibility 519
Using assistive technologies 519
Keyboard navigation of the user interface 519

Notices 521
Programming Interface Information 522
Trademarks 522

Glossary 523

Index 529

Figures

1. Sample CVAF Dump Title Line 39
2. Sample DADSM Dump Title Line 105
3. Sample AMBLIST listing 133
4. Server Diagnostic Area (SRVDGN) Format 136
5. ISMF—Profile Option Menu 144
6. ISMF—Entry Panel for Logging and Abend Parameters 145
7. ISMF—ERTB Display Panel 146
8. ISMF—Symptom Dump in the ISPF Log 147
9. ISPF—Symptom Dump in the ISPF Log 148
10. ISMF—Detailed Error Entries in the ISPF Log 149
11. ISMF—ISPF Log with ISMF Inter-Module Trace Entries 150
12. ISMF—ISPF Log with ISMF Trace-Point Entries 150
13. Media Manager—Summary Dump 188
14. Storage Dump Showing the Transfer Control Table for Module IFG0202J 247
15. Output from the GTF Data Set Resulting from Problem Determination Abend Error Recording Routine (213–04 Abend) 250
16. Contents of Recovery Routine Control Block Save Area (RRCBSA) 251
17. Storage Dump Containing the RRCBSA Passed by the O/C/EOV Recovery Routines 252
18. Output from GTF Data Set Resulting from the Optional Work Area Trace 258
19. Formatted SYS1.LOGREC Entry Example 383
20. Formatted SYS1.LOGREC Entry 405
21. SMSDATA Formatting of the SSIVT Control Block 426
22. SMSDATA Formatting of the SMS Trace Table 427
23. SMSDATA Formatting of an SMS Automatic Data Area 429
24. SMSDATA Formatting of the SMS Procedure Trace Table 430
25. Sample SMS SYS1.LOGREC Entry 433
26. SMS Sample SUMDUMP—Eye-catchers in EBCDIC Section 436
27. Example of TEST Option Output 476
28. VSAM Record Management—How to Find a Damaged Data Set 495

© Copyright IBM Corp. 1979, 2002

Tables

1. AOM—CTRACE Subcommand . . . 4
2. AOM Trace Table Abbreviations . . . 4
3. AOM DASD Return and Reason Codes . . . 7
4. AOM Tape Library Return and Reason Codes . . . 11
5. Messages and Codes Issued by BDAM Modules . . . 13
6. Compression Management Services Return Code Format . . . 16
7. CMP Reason Code Module Identifiers . . . 17
8. CMP Activation Return and Reason Codes . . . 17
9. CMP Allocation Return and Reason Codes . . . 18
10. CMP Return and Reason Codes . . . 18
11. CVAF Module-to-Function Cross-Reference . . . 27
12. Format of GTF Trace Records . . . 29
13. Format of Trace Data for Write DSCB . . . 29
14. Format of Trace Data for Write VIR . . . 30
15. Format of Trace Data for Add Index . . . 30
16. Format of Trace Data for Delete Index . . . 30
17. Format of Trace Data for Allocate Space . . . 31
18. Format of Trace Data for Release Space . . . 31
19. Format of Trace Data for Allocate VTOC Index Space . . . 31
20. Format of Trace Data for Release VTOC Index Space . . . 32
21. Format of Trace Data for Allocate VTOC Space . . . 32
22. Format of Trace Data for Release VTOC Space . . . 32
23. Function and Subfunction Keywords for DADSM and CVAF Functions . . . 46
24. DADSM CREATE Return Codes . . . 48
25. DADSM CREATE Diagnostic Information (Register 0) . . . 50
26. Abbreviation Descriptions for DADSM CREATE Diagnostic Information (Register 0) . . . 64
27. DADSM Extend Diagnostic Information . . . 65
28. Abbreviation Descriptions for DADSM EXTEND Diagnostic Information . . . 68
29. DADSM End-of-Volume Return Codes . . . 68
30. DADSM Catalog Management Return Codes . . . 69
31. DADSM PDSE Management Return Codes . . . 69
32. DADSM ISJF Return Codes . . . 70
33. DADSM ISJF Diagnostic Information . . . 70
34. DADSM LSPACE Return Codes . . . 71
35. DADSM LSPACE Diagnostic Information (Register 0) . . . 72
36. Abbreviation Descriptions for DADSM LSPACE Return and Diagnostic Codes . . . 73
37. DADSM—LSPACE Parameter List Diagnostic Information Fields . . . 73
38. DADSM OBTAIN Function Return Codes . . . 73
39. DADSM Partial Release Function Return Codes . . . 75
40. DADSM PARTREL Diagnostic Information (Register 0) . . . 76
41. Abbreviation Descriptions for PARTREL Diagnostic Information (Register 0) . . . 82
42. DADSM RENAME Return Codes . . . 83
43. DADSM RENAME Diagnostic Information (Register 0) . . . 84
44. Abbreviation Descriptions for DADSM RENAME Diagnostic Information (Register 0) . . . 89
45. Status Codes from RENAME Macro Instruction . . . 90
46. DADSM SCRATCH Return Codes (R15) . . . 91
47. DADSM SCRATCH Diagnostic Information (Register 0) . . . 92
48. Abbreviation Descriptions for DADSM SCRATCH Diagnostic Information (Register 0) . . . 99
49. Status Codes from SCRATCH Macro Instruction . . . 100
50. Secondary Status Codes . . . 100
51. DADSM UPDATE Diagnostic Information (Register 0) . . . 101
52. Abbreviation Descriptions for DADSM UPDATE Diagnostic Information (Register 0) . . . 103
53. DADSM Common Diagnostic Codes . . . 104

© Copyright IBM Corp. 1979, 2002

54. PDA Information . . . 107
55. Nonzero Return Codes and Applicable Reason Codes . . . 108
56. Nonzero Return Codes and Applicable Reason and Status Codes . . . 109
57. Nonzero Return Codes and Applicable Reason and Response Codes . . . 110
58. Message Returned during Conversation Initialization . . . 125
59. Messages Returned by Non-Control DSSs by Communications Manager/Parser . . . 125
60. Reply Messages Returned for Control DSSs by Communications Manager . . . 125
61. DDM Reply Messages Returned by Distributed FileManager/MVS . . . 126
62. Valid Options for this Trace Facility . . . 129
63. Distributed FileManager/MVS Control Blocks Formatted by SMSXDATA . . . 131
64. Distributed FileManager/MVS SRVDGN Data Items . . . 136
65. Distributed FileManager/MVS General Error Codes . . . 137
66. Distributed FileManager/MVS Communications Error Codes . . . 139
67. Distributed FileManager/MVS Agent Error Codes . . . 140
68. Distributed FileManager/MVS File/Directory Error Codes . . . 141
69. Distributed FileManager/MVS Locking Error Codes . . . 141
70. Distributed FileManager/MVS Access Method Error Codes . . . 141
71. ISMF—Standard Return Codes . . . 151
72. OSREQ Return Codes . . . 196
73. OSREQ Return and Reason Codes . . . 197
74. OAM Macro Return Codes . . . 206
75. OAM Macro Reason Codes . . . 206
76. CBRXLCS Return and Reason Codes . . . 212
77. CBRXLIB Return Codes . . . 215
78. CBRXVOL Return Codes . . . 216
79. CBRXTCF Return Codes . . . 217
80. CBRXLACS FUNC(MOUNT) Return and Reason Codes . . . 218
81. CBRXLACS FUNC(DEMOUNT) Return and Reason Codes . . . 222
82. CBRXLACS FUNC(WAIT) Return and Reason Codes . . . 224
83. CBRXLACS FUNC(VERIFY) Return and Reason Codes . . . 226
84. CBRXLACS FUNC(CANCEL) Return and Reason Codes . . . 228
85. CBRXLACS FUNC(WTO) Return and Reason Codes . . . 230
86. CBRXLACS FUNC(DOM) Return and Reason Codes . . . 231
87. CBRXLACS FUNC(ERRTEXT) Return and Reason Codes . . . 232
88. 9246 Optical Disk Library Long OBR . . . 234
89. 9247 Optical Disk Drive Long OBR . . . 235
90. 3995 Library Long OBR Format . . . 237
91. 3995 Optical Disk Drive Long OBR . . . 238
92. 3995 Library MDR . . . 239
93. Function Call Return Codes in the AX Register . . . 245
94. Function Call Return Codes in the AX Register . . . 246
95. SCSI Adapter Completion Code . . . 246
96. O/C/EOV Recovery Routine Parameter List . . . 253
97. Valid Trace Facility Options . . . 376
98. Functional Areas of HFS Management . . . 377
99. HFS Common Return Codes . . . 389
100. Valid Trace Facility Options . . . 399
101. Functional Areas of PDSE Management . . . 399
102. PDSE Common Return Codes . . . 407
103. Extended Format I/O Routine . . . 413
104. SAM Interrupt Control Block (ICB)—Description and Format . . . 416
105. SAM Diagnostic Codes . . . 417
106. Control Blocks Formatted by SMSDATA . . . 421
107. Valid Options with the SELECT Keyword for the SMSDATA Verb . . . 431
108. SMS—ACS Translator Return Codes . . . 438
109. SMS—ACS Executor Return Codes . . . 438

110. SMS Initialization Return Codes 438
111. SMS Interface Return Codes 439
112. X'F61' Record Information 480
113. Valid Trace Facility Options 484
114. Initialization Errors 486
115. SMSVSAM Return and Reason Codes 489
116. SMPM_CFPurge Return and Reason Codes 491
117. SMPM_CFQuery Return and Reason Codes 491
118. IDCAMS-Detected Messages 496
119. Predefined Trace IDs, Modules, and Functions 499
120. PARM1 Subparameter Bits. 500
121. PARM1 Subparameter Bits. 501
122. PARM1 Subparameter Bits. 501
123. PARM1 Subparameter Bits. 501
124. PARM1 Subparameter Bits. 502
125. PARM1 Subparameter Bits. 502
126. PARM2 Subparameter Bits. 502
127. Return Codes from the Record-Management(Request) Macros 504
128. Return Codes from the Record-Management(Request) Macros 505
129. Return Codes from the Record-Management(Request) Macros - R15=0 505
130. Function Codes For Logical and Physical Errors 506
131. Contents of Registers When a LERAD Routine Gets Control 506
132. Logical-Error Reason Codes found in the RPL Feedback Field when R15=8 and LERAD is not active. 507
133. Contents of Registers When a SYNAD Routine Gets Control 513
134. Physical-Error Reason Codes in the RPL Feedback Field from a Request Macro 513
135. Format of Physical-Error Messages 514
136. Control Block Manipulation Return Codes 517
137. Control Block Manipulation Error Codes 517

About This Book

This book helps you gather *detailed* DFSMSdfp™ failure-related data. You typically gather this data after identifying a failure and determining that an APAR record does not exist for it in the IBM® software support database. It is not intended to enable you to do in-depth problem diagnosis. *z/OS DFSMSdfp Diagnosis Guide*, GY27-7617, helps you initially to identify, describe, report, and resolve DFSMSdfp failures.

When the diagnosis procedures in *z/OS DFSMSdfp Diagnosis Guide* or the IBM Support Center directs you to this book, you should use the reference information herein to gather and interpret failure-related data that supplements any data obtained from the initial failure.

References to CICS®

In this book, when the term "CICS," is used, if refers to the CICS/MVS®, CICS/ESA®, and to the CICS Transaction Server for z/OS. In the sections describing VSAM RLS or the SMSVSAM address space, CICS refers to only the CICS Transaction Server, which is required for VSAM RLS. Citations are to the CICS Transaction Server for z/OS library; CICS/MVS and CICS/ESA users should use the corresponding books in those libraries.

Required Product Knowledge

This book is intended to be used with *z/OS DFSMSdfp Diagnosis Guide*, GY27-7617.

Referenced Publications

The following publications are referenced in this book.

Publication Title	Order Number
DB2 Command Reference	SC26-9934
DB2 Diagnosis Guide and Reference	LY37-3740
DB2 Messages and Codes	GC26-9940
DB2 Utility Guide and Reference	SC26-9945
EREP User's Guide	GC35-0151
EREP Reference	GC35-0152
z/OS DFSMS Access Method Services	SC26-7394
z/OS DFSMS Installation Exits	SC26-7396
z/OS DFSMS Macro Instructions for Data Sets	SC26-7408
z/OS DFSMS OAM Application Programmer's Reference	SC35-0425
z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support	SC35-0426
z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries	SC35-0427
z/OS DFSMS: Using Data Sets	SC26-7410
z/OS DFSMSdfp Advanced Services	SC26-7400

© Copyright IBM Corp. 1979, 2002

Publication Title	Order Number
z/OS DFSMSdfp Diagnosis Guide	GY27-7617
z/OS DFSMSdfp Storage Administration Reference	SC26-7402
z/OS ISPF Reference Summary	SC34-4816
z/OS ISPF User's Guide	SC34-4822
z/OS MVS Diagnosis: Reference	GA22-7588
z/OS MVS Diagnosis: Tools and Service Aids	GA22-7589
z/OS MVS Dump Output Messages	SA22-7590
z/OS MVS Initialization and Tuning Reference	SA22-7592
z/OS MVS IPCS Commands	SA22-7594
z/OS MVS IPCS Customization	SA22-7595
z/OS MVS IPCS User's Guide	SA22-7596
z/OS MVS JCL Reference	SA22-7597
z/OS MVS JCL User's Guide	SA22-7598
z/OS MVS Programming: Assembler Services Reference ABE-HSP	SA22-7606
z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN	SA22-7609
z/OS MVS Programming: Authorized Assembler Services Reference ENF-IXG	SA22-7610
z/OS MVS Programming: Authorized Assembler Services Reference LLA-SDU	SA22-7611
z/OS MVS Programming: Authorized Assembler Services Reference SET-WTO	SA22-7612
z/OS MVS Routing and Descriptor Codes	SA22-7624
z/OS MVS System Codes	SA22-7626
z/OS MVS System Commands	SA22-7627
z/OS MVS System Messages, Vol 1 (ABA-AOM)	SA22-7631
z/OS MVS System Messages, Vol 2 (ARC-ASA)	SA22-7632
z/OS MVS System Messages, Vol 3 (ASB-BPX)	SA22-7633
z/OS MVS System Messages, Vol 4 (CBD-DMO)	SA22-7634
z/OS MVS System Messages, Vol 5 (EDG-GFS)	SA22-7635
z/OS MVS System Messages, Vol 6 (GOS-IEA)	SA22-7636
z/OS MVS System Messages, Vol 7 (IEB-IEE)	SA22-7637
z/OS MVS System Messages, Vol 8 (IEF-IGD)	SA22-7638
z/OS MVS System Messages, Vol 9 (IGF-IWM)	SA22-7639
z/OS MVS System Messages, Vol 10 (IXC-IZP)	SA22-7640
z/OS TSO/E Command Reference	SA22-7782
z/OS TSO/E Primer	SA22-7787
z/OS TSO/E User's Guide	SA22-7794

Accessing z/OS DFSMS Books on the Internet

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

From this Web site, you can link directly to the z/OS softcopy books by selecting the Library icon. You can also link to IBM Direct to order hardcopy books.

Using LookAt to look up message explanations

LookAt is an online facility that allows you to look up explanations for z/OS messages, system abends, and some codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can access LookAt from the Internet at:

`http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookat.html`

or from anywhere in z/OS where you can access a TSO command line (for example, TSO prompt, ISPF, z/OS UNIX System Services running OMVS).

To find a message explanation on the Internet, go to the LookAt Web site and simply enter the message identifier (for example, `IAT1836` or `IAT*`). You can select a specific release to narrow your search. You can also download code from the *z/OS Collection*, SK3T-4269 and the LookAt Web site so you can access LookAt from a PalmPilot (Palm VIIx suggested).

To use LookAt as a TSO command, you must have LookAt installed on your host system. You can obtain the LookAt code for TSO from a disk on your *z/OS Collection*, SK3T-4269 or from the LookAt Web site. To obtain the code from the LookAt Web site, do the following:

1. Go to http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookat.html.
2. Click the **News** button.
3. Scroll to **Download LookAt Code for TSO and VM.**
4. Click the ftp link, which will take you to a list of operating systems. Select the appropriate operating system. Then select the appropriate release.
5. Find the **lookat.me** file and follow its detailed instructions.

To find a message explanation from a TSO command line, simply enter: **lookat** *message-id*. LookAt will display the message explanation for the message requested.

Note: Some messages have information in more than one book. For example, IEC192I has routing and descriptor codes listed in *z/OS MVS Routing and Descriptor Codes*. For such messages, LookAt prompts you to choose which book to open.

Accessing Licensed Books on the Web

z/OS licensed documentation in PDF format is available on the Internet at the IBM Resource Link Web site at:

`http://www.ibm.com/servers/resourcelink`

Licensed books are available only to customers with a z/OS license. Access to these books requires an IBM Resource Link Web userid and password, and a key code. With your z/OS order you received a memo that includes this key code.

To obtain your IBM Resource Link Web userid and password log on to:
`http://www.ibm.com/servers/resourcelink`

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

1. Log on to Resource Link using your Resource Link user ID and password.
2. Click on **User Profiles** located on the left-hand navigation bar.
3. Click on **Access Profile.**
4. Click on **Request Access to Licensed books.**
5. Supply your key code where requested and click on the **Submit** button.

If you supplied the correct key code you will receive confirmation that your request is being processed. After your request is processed, you will receive an e-mail confirmation.

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

To access the licensed books:

1. Log on to Resource Link using your Resource Link userid and password.
2. Click on **Library**.
3. Click on **zSeries**.
4. Click on **Software**.
5. Click on **z/OS**.
6. Access the licensed book by selecting the appropriate element.

How to Send Your Comments

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

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

  Be sure to include the name of the book, the part number of the book, version and product name, and if applicable, the specific location of the text you are commenting on (for example, a page number or a table number).
- Fill out one of the forms at the back of this book and return it by mail or by giving it to an IBM representative. If the form has been removed, address your comments to IBM Corporation, RCF Processing Department M86/050, 5600 Cottle Road, San Jose, California 95193-0001, U.S.A.

Summary of Changes

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

You might notice changes in the style and structure of some content in this book—for example, headings that use uppercase for the first letter of initial words only or procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our books.

Summary of Changes for GY27-7618-02 z/OS Version 1 Release 3

This book contains information previously presented in *z/OS V1R1 DFSMSdfp Diagnosis Reference* (GY27-7618-01).

The following sections summarize the changes to that information.

New Information

This edition includes the following new information:

- New return codes and reason codes are added to various sections to help you diagnose problems.
- PDSE diagnostic aids are expanded to enhance the two operator commands: ANALYSIS and FREELATCH.

Deleted Information

The section ″SMS Trace for OAM Diagnosis Data″ in the OAM chapter was removed. The trace facility that was described in that section is not reliable and is no longer used.

The information that was in the chapter ″Linkage Editor/Loader/Binder Diagnostic Aids″ was moved to *z/OS MVS Diagnosis: Reference* (program management chapter).

Changed Information

Descriptions of return codes and reason codes are improved throughout the diagnostic aids sections.

© Copyright IBM Corp. 1979, 2002

Chapter 1. General Diagnostic Aids

This chapter describes notational conventions for the syntax diagrams in this manual and the types of system dumps that you might use to diagnose problems and failures.

Notational Conventions

This book describes various commands that you can use to obtain and print dumps. A uniform notation describes the syntax of these commands. This notation is not part of the language; it is merely a way of describing the syntax of the commands. The command syntax definitions in this book use the following conventions:

[ ] Brackets enclose an optional entry. You can, but need not, include the entry. Examples follow:
- **[***length***]**
- **[MF=E]**

| An OR sign (a vertical bar) separates alternative entries. You must specify one, and only one, of the entries unless you allow an indicated default. Examples follow:
- **[REREAD|LEAVE]**
- **[***length***|'S']**

{ } Braces enclose alternative entries. You must use one, and only one, of the entries. Examples follow:
- **BFTEK={S|A}**
- **{K|D}**
- **{***address***|S|O}**

Sometimes alternative entries are shown in a vertical stack of braces. An example follows:

```
MACRF={{(R[C|P])   }
       {(W[C|P|L]) }
       {(R[C],W[C])}}
```

In the preceding example, you must choose only one entry from the vertical stack.

... An ellipsis indicates that the entry immediately preceding the ellipsis can be repeated. For example:
- **(***Dcbaddr***,[(***options***)],...)**

UPPERCASE BOLDFACE
Uppercase boldface type indicates entries that you must code exactly as shown. These entries consist of keywords and the following punctuation symbols: commas, parentheses, and equal signs. Examples follow:
- **CLOSE , , , ,TYPE=T**
- **MACRF=(PL,PTC)**

UNDERSCORED UPPERCASE BOLDFACE
Underscored uppercase boldface type indicates the default used if you do not specify any of the alternatives. Examples follow:
- **[EROPT={ACC|SKP|ABE}]**
- **[BFALN={F|D}]**

© Copyright IBM Corp. 1979, 2002

lowercase italic

Lowercase italic type indicates a value that you supply, according to the specifications and limits for the parameter. Examples follow:

- *number*
- *image-id*
- *count*

Types of System Dumps

The following table describes the various types of dumps. If a dump is known by more than one name, the table also mentions the alternate name.

For a sample of the kind of information in the formatted portion of a typical SVC dump, see the section that describes the abend keyword in *z/OS DFSMSdfp Diagnosis Guide*.

Dump	Data Set	Type	Print By	Caused By
SYSABEND	Defined by JCL (SYSOUT=A)	Formatted	Normally SYSOUT=A	An abend condition (and only taken if the step contains a SYSABEND DD statement).
SYSUDUMP	Defined by JCL (SYSOUT=A)	Formatted	Normally SYSOUT=A	An abend condition (and only taken if the step contains a SYSUDUMP DD statement).
SYSMDUMP	Defined by JCL (DSN=xxx)	Machine readable	Normally read via IPCS	An abend condition (and only taken if the step contains a SYSMDUMP DD statement).
STANDALONE	Defined by installation (tape or disk)	Machine readable	Printed and formatted via IPCS	Operator doing an IPL of the standalone dump program.
DYNAMIC (SVC Dump or SDUMP)	SYS1.DUMPxx	Machine readable	Printed and formatted via IPCS	Operator entering DUMP command.
FRR (SVC Dump or SDUMP)	SYS1.DUMPxx	Machine readable	Printed and formatted via IPCS	DFSMS ESTAE exit or functional recovery routine detecting an error. (For example, an ABEND071 “PSW restart key depressed” causes an FRR to take a dump.)

Notes:

1. For a complete description of the IPCS dump formatter, see *z/OS MVS IPCS User’s Guide*.
2. The `IPCS SUMDUMP` option provides a useful summary of failure-related information from SYS1.DUMP*xx* data sets.
3. For detailed information about diagnosing dumps, see *z/OS MVS Diagnosis: Tools and Service Aids*.

Chapter 2. AOM Diagnostic Aids

The Asynchronous Operations Manager (AOM) manages the interface to the host for the control operations performed by the 3990 Storage Control, the IBM TotalStorage™ Enterprise Automated Tape Library (3494), and the 3495 Tape Library Dataserver.

AOM performs the following functions:

- Issues two types of control operations:
  - Synchronous I/O operations—those I/O operations that are complete when the host receives both channel end and device end.
  - Asynchronous I/O operations—those operations that are not complete when the host receives both channel end and device end. The following takes place to complete an asynchronous operation:
    - The operation is validated.
    - The storage control sends channel end and device end to the host.
    - The storage control performs the operation.
    - Optionally, the storage control raises an attention interrupt to the host, signalling that the operation is complete.
    - The host processes the attention message to determine the operations status.
- Processes attention interrupts from the storage control to the host, notifying the host when certain asynchronous operations complete.
- Processes device state transition interrupts from the storage control to the host notifying all attached hosts when the subsystem or a device changes status. The storage control sends a device state transition interrupt to the host. This interrupt is unsolicited status of attention, device end, and unit exception. (A control unit end or busy might also be merged into this status.)

Using IPCS to Format the AOM Trace Table

This section provides the following information:

- How to use the interactive problem control system (IPCS) to format the AOM trace table
- Descriptions of some of the trace table entries that are used in AOM
- List of AOM service return codes and reason codes

AOM provides a component wraparound trace table with information that is useful in diagnosing failures. For systems prior to DFSMS 1.1, use the `IPCS VERBEXIT` subcommand to format and analyze the trace table. For DFSMS 1.1 systems, use the `IPCS CTRACE` subcommand to format and analyze the trace table.

© Copyright IBM Corp. 1979, 2002

CTRACE Subcommand

Table 1 shows the syntax of the CTRACE subcommand. For a full explanation of IPCS, the CTRACE subcommand, and its parameters and defaults, see *z/OS MVS IPCS Commands*.

Table 1. AOM—CTRACE Subcommand

Operation	Parameters
CTRACE COMP(SYSAOM)	**{SHORT|FULL}** **{GMT|LOCAL}** **{START(***mm/dd/yy,hh.mm.ss.dddddd***)}** **{STOP(***mm/dd/yy,hh.mm.ss.dddddd***)}** **{DSNAME(***dsname***)|DATASET(***dsname***)}** **{FILE(***ddname***)|DDNAME(***ddname***)}** **{PRINT|NOPRINT}** **{TERMINAL|NOTERMINAL}** **{TEST|NOTEST}**

When the IPCS input source is a dump or active storage, the CTRACE COMP(SYSAOM) subcommand displays the AOM trace entries that the wraparound AOM trace table contains. To increase the size of the AOM trace, use a CTRACE output writer.

To use a CTRACE output writer, you must take the following steps:

1. Create a CTRACE writer procedure (named CTWTR in the example)
2. Start AOM CTRACE by entering the operator TRACE command:

```
 TRACE CT,ON,COMP=SYSAOM
*04 ITT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND.
 R 4,WTR=CTWTR,END
```

3. Start the CTRACE writer by entering the operator TRACE command:

```
TRACE CT,WTRSTART=CTWTR
```

4. To look at CTRACE data, first STOP the CTRACE writer with the TRACE operator command:

```
TRACE CT,WTRSTOP=CTWTR
```

5. Enter the IPCS CTRACE COMP(SYSAOM) subcommand and use the CTRACE WTR data set as the SOURCE value.

AOM Trace Table Abbreviations

Table 2 describes some of the abbreviations that are used to describe the data that is contained in the AOM trace table.

Table 2. AOM Trace Table Abbreviations

Abbreviation	Description
AAQE	AAQE address.
ABCO	Abend code.
AECB	Asynchronous attention queue element (AAQE) event control block.
AFLG	Contents of AAQEFLG1.
AFGL	Contents of AAQEFLG1.
ASID	Address space identification.

Table 2. AOM Trace Table Abbreviations (continued)

Abbreviation	Description
CCOD	Identifies the module that generated the trace entry. AOMIPCS translates CCOD to NAME when formatting trace table entries.
CECB	Channel end/device end event control block.
CCWP	Address of the channel program.
CPGM	Channel program.
DAT1	Data determined by the type of operation.
DAT2	Continuation of data determined by the type of operation.
DEV	Device identification.
DEVI	Device identification number.
DSEB	Address of the device service exit parameter list for sense subsystem status.
DSEF	Address of the device service exit buffer for read configuration data.
DSEP	Address of the device service exit parameter list for read device characteristics.
ECOD	Event code. AOMIPCS translates ECOD to TRC***n*** when formatting trace table entries. ***n*** is a hex number from 1 to A.
EFLG	Footprint® flags.
FLG1	Contents of SERVFLG1.
FLG2	Contents of AAQEFLG2.
FLG3	Contents of SERVFLG3.
FLGS	Contents of SERVFLGS.
FRC	Function return code.
FRRF	AOMQERR FRR footprint.
FUNC	AOMQMGR function code.
HEAD	Address of the first trace table entry.
ICOD	Input/Output Supervisor completion code.
IFLA	Input/Output Supervisor flag byte A.
INDX	Configuration table index.
IOSB	Input/Output Supervisor control block address.
IOST	Input/Output Supervisor status.
IOSS	Input/Output Supervisor status.
IOSN	First two bytes of sense information.
LEN	Length of trace table entry.
LPUM	Last path used mask.
NEXT	Address of the next available trace table entry.
NEND	End of the general section of the trace table entry.
MSG	Asynchronous message-ID.
MSID	Message identification.
PATH	Path mask.
PTHM	Path mask.
PTHI	Channel path identification.

Table 2. AOM Trace Table Abbreviations (continued)

Abbreviation	Description
QUEL	QUEL address.
RETP	Return address.
REQT	Request type.
RETC	AOM Return code.
REAS	AOM Reason code.
RSDP	Read subsystem data address.
RSSD	Read subsystem data.
RG00	Values in register 0.
RG01	Values in register 1.
RG13	Values in register 13.
RG14	Values in register 14.
RG15	Values in register 15.
RSD	Read subsystem data address.
RSDD	RSD data.
SSDP	Address of the subsystem data area.
SSDL	Length of the subsystem data area.
SSDN	Needed length of the subsystem data area.
SENS	Sense data.
SEQN	Trace table sequence number.
SRVL	Pointer to SERVL
SSID	3990 Subsystem identification number.
SRV1	SERVL used by AOMDRVR for the original asynchronous I/O request.
SRV2	SERVL used by AOMDTRM for the read subsystem data channel program.
TCB	Task control block address.
TAIL	Address of the last trace table entry.
TIME	Asynchronous timeout value.
TYPE	Identifies the trace table entry type.
TRID	Identifies where the trace was issued within the module. AOMIPCS translates TRID to EP and LOC when formatting trace table entries.
TOD	Time of day.
UCBN	Device number.
UCBP	Unit control block address.
UCB	Unit control block address.
UNIT	Device number.
VNBR	Identifies the format of the variable trace data.
VLEN	Length of the following trace data.
VDAT	Format-dependent information for each trace table entry.

AOM Service Return and Reason Codes

The Asynchronous Operations Manager service return codes and reason codes specify the condition that occurred. Table 3 lists the explanation and actions to take for each code.

Table 3. AOM DASD Return and Reason Codes

Return Code	Reason Code	Description
0 (X'00')		SUCCESSFUL: The request to the service completed without error.
4 (X'04')		REQUEST FAILED: The request to the service completed with the reason code specified below.
	1 (X'01')	Some or all of the eligible volumes for allocation are unknown to the system.
	2 (X'02')	Some or all of the separation volumes are unknown to the system.
	3 (X'03')	Some or all of the volumes in both the eligible and separation lists are unknown to the system.
	4 (X'04')	INSUFFICIENT STORAGE: GETMAIN failure occurred within the service. This may indicate that the region size for your IDCAMS job or the region size for TSO/E needs to be increased. IDCAMS should have issued IDC2011I when this error code was returned. Refer to IDC2011I for details on the error condition. If the GETMAIN failure is for a system component, then a system message indicating the failure should appear on the system console, and you should follow the directions for the system component's message.
	8 (X'08')	INVALID PARAMETER LIST: Internal error. IDCAMS should have issued IDC31601I or IDC31550I for this error condition. Refer to this message for further details on the error. If the error was received for a system component, this error code indicates an internal error in that component.
	12 (X'0C')	UNAUTHORIZED REQUEST: IDCAMS was not authorized for this request. Possible reasons are that IDCAMS was: invoked by a TSO CALL command; not link-edited as authorized; not residing in an authorized library. This should not occur with any system component modules. Contact your system programmer.
	16 (X'10')	UNABLE TO ESTABLISH ESTAE: Internal error. IDCAMS should issue message IDC2391I when this error occurs. Refer to this message for further details on the error condition. If this error occurs with any other system component, refer to messages issued by that component (for example, SMS) for details on that component's recovery actions.
	20 (X'14')	INVALID CCW OR DATA AREA POINTER: Internal error. IDCAMS should issue message IDC31602 when this error occurs. Refer to this message for further details on the error condition. This is an internal error with all other system components. Refer to messages that the other components issue.
	24 (X'18')	INVALID REQUEST TYPE FOR CONTROL UNIT: Internal error.

Table 3. AOM DASD Return and Reason Codes (continued)

Return Code	Reason Code	Description
	28 (X'1C')	SUBSYSTEM DATA AREA NOT LARGE ENOUGH: Internal error.
	32 (X'20')	UNABLE TO LOCATE SSSCB ENTRY: Internal error.
	36 (X'24')	REQUEST BUFFER SIZE NOT A MULTIPLE OF 8: Internal error.
	40 (X'28')	CONFIGURATION TABLE INDEX TOO HIGH: Internal error.
	44 (X'2C')	SSSCBUPD VALUE(S) INVALID: Internal error.
	48 (X'30')	UNABLE TO OBTAIN COMPLETE SSSCB ENTRY: Internal error.
	52 (X'34')	DEVICE NUMBER INVALID: Internal error.
	56 (X'38')	UCB ADDRESS OR DEVICE NUMBER NEEDED FOR RACF® CHECK: Internal error.
	60 (X'3C')	INVALID RACF LEVEL: Internal error.
	64 (X'40')	INVALID COMMIT/DISCARD EXTENTS OR DISCARD PSFFLAG: Internal error.
	68 (X'44')	UNABLE TO DESTAGE DATA ON A COMMIT: Internal error.
	72 (X'48')	INVALID CFW TOKEN: Internal error.
	76 (X'4C')	ONLY ONE ACCESS CODE AVAILABLE: Internal error.
	80 (X'50')	INVALID EYE-CATCHER IN SERVL OR UNABLE TO ACCESS LAST BYTE IN SERVL: Internal error.
	84(X'54')	UCB ADDRESS OR DEVICE NUMBER POINTS TO NON-DASD DEVICE: Internal error.
	88 (X'58')	SSDL MISSING: Internal error.
	92 (X'5C')	REQTYPE DRVR PARM(S) NOT ALLOWED: Internal error.
	96 (X'60')	SSDP AND SSDL NOT ALLOWED: Internal error.
	100 (X'64')	REQUEST BUFFER HAS ONLY ONE SUB-BUFFER: Internal error.
	104 (X'68')	INDIRECT ADDRESS LISTS NOT ALLOWED IN DRVR CHANNEL PROGRAM: Internal error.
	108 (X'6C')	TIC MUST POINT AT *-8: Internal error.
	112 (X'70')	UCB ADDRESS OR DEVICE NUMBER MISSING: Internal error.
	116 (X'74')	SSDP MISSING: Internal error.
	120 (X'78')	COUNT MISSING FROM EXTENT BUFFER: Internal error.
	124 (X'7C')	NO 3990 MODEL 3 IN THE SYSTEM: Internal error.
	128 (X'80')	INVALID REQUEST FOR REQTYPE MULTI: Internal error.
	132 (X'84')	INVALID EYE-CATCHER IN INPUT BUFFER: Internal error.
	136 (X'88')	SSDP POINTS TO AREA NOT OWNED BY CALLER: Internal error.

Table 3. AOM DASD Return and Reason Codes (continued)

Return Code	Reason Code	Description
	140 (X'8C')	DEVICE TO BE USED FOR AOMSERV REQUEST DOES NOT POINT TO THE BASE UCB OF A MULTIPLE EXPOSURE D/T3350: AOMSERV caller error. UCB pointer must be corrected to point at the base UCB for the device.
8 (X'08')		OPERATION FAILED: The request to the service completed with the reason code specified below.
	1 (X'01')	AOM01039 was passed a SERVL in which either SERVVPTR or SEVSPTR is zero.
	2 (X'02')	AOM01039 was passed a SERVL in which either the eligible or separation volume count is zero.
	3 (X'03')	The chain of pointers needed to locate the SSCT contains a zero pointer.
	4 (X'04')	I/O ERROR EXECUTING CALLER'S CHANNEL PROGRAM: An I/O error occurred. IDCAMS issues message IDC31559I when sense data is available. Otherwise, IDCAMS issues message IDC2375I. Refer to that message for further information on the error condition. For other system components (for example, SMS), refer to messages issued by those components detailing actions taken as the result of this I/O error. Refer to SYS1.LOGREC for information regarding the specific I/O error that occurred.
	5 (X'05')	Volume List storage is not owned by caller. Set by IGX00039 ESTAE routine.
	6 (X'06')	Separation List storage is not owned by caller. Set by IGX00039 ESTAE routine.
	8 (X'08')	ASYNCHRONOUS OPERATION PENDING: The asynchronous operation is still in progress, and IDCAMS should still be waiting for completion. For other system components (for example, SMS), refer to messages issued by those components detailing actions taken as the result of this return/reason code.
	12 (X'0C')	ASYNCHRONOUS OPERATION COMPLETED IN ERROR: IDCAMS should issue message IDC31610I for this error condition. Refer to that message for details on the error. For other system components (for example, SMS), refer to messages issued by those components detailing actions taken as the result of this return/reason code.
	16 (X'10')	ASYNCHRONOUS OPERATION FAILED: IDCAMS should issue message IDC31610I for this error condition. Refer to that message for details on the error. For other system components (for example, SMS), refer to messages issued by those components detailing actions taken as the result of this return/reason code.

Table 3. AOM DASD Return and Reason Codes (continued)

Return Code	Reason Code	Description
	20 (X'14')	ASYNCHRONOUS OPERATION TIME OUT: The asynchronous operation is still in progress, but the timer for IDCAMS waiting for the operation has expired. When the asynchronous operation completes, if it 'Fails' or 'Completes In Error' message AOM000I will be issued to the system console. If the operation is successful, the status will be correctly reflected in the storage control and in all attached host systems. Generally this error will occur while establishing or reestablishing a dual copy pair and contention from other hosts or the host trying to do the operation causes time to be taken away from the 3990 copy operation. For other system components (for example, SMS), refer to messages issued by those components detailing actions taken as the result of this return/reason code.
	24 (X'18')	REQUEST LOST BY SUBSYSTEM: The storage control lost knowledge of the asynchronous operation. Refer to a merged LOGREC listing for more details. Running an IDCAMS LISTDATA report will give you the current state of the subsystem and device. After determining the current state, you may need to rerun your IDCAMS job to establish the new states that you desire. For other system components (for example, SMS), refer to messages issued by those components detailing actions taken as the result of this return/reason code.
	28 (X'1C')	DESTAGE-ALL DID NOT COMPLETE SUCCESSFULLY: message AOM001I is issued for a failing subsystem. Refer to message AOM001I for the problem determination information required to isolate the devices attached to the subsystem that have pinned data.
	32 (X'20')	I/O ERROR EXECUTING INTERNAL CHANNEL PROGRAM: Internal error.
	36 (X'24')	I/O ERROR AFTER RE-DRIVING CHANNEL PROGRAM TO ESTABLISH A DUAL COPY PAIR: Internal error.
	40 (X'28')	THE DEVICE TO BE USED FOR THE AOM DRIVER REQUEST HAS NO LOGICAL PATHS ONLINE OR THE I/O OPERATION COMPLETED WITH CONDITION CODE 3 (Device not operational): Ensure that the correct device is being referenced and that a minimum of one logical and one physical path is available.
	44 (X'2C')	AOM RECEIVED INTERVENTION REQUIRED IN RESPONSE TO AN I/O OPERATION TO DEVICE: Ensure that the correct device is being referenced and is available to system. If the device is available, follow your installation's procedures for resolving hardware problems.
12 (X'0C')		ASYNCHRONOUS OPERATIONS MANAGER ERROR: The request to the service completed with the reason code specified below.
	1 (X'01')	Load module AOM01039 could not be loaded by module IGX00039. SERVFRC will contain the failure return code from LOAD. This code is set by module IGX00039.
	2 (X'02')	AOM01039 was passed a SERVL in which either the eligible or separation volume count is zero.

Table 3. AOM DASD Return and Reason Codes (continued)

Return Code	Reason Code	Description
	4 (X'04')	INTERNAL QUEUE ERROR: Internal error.
	8 (X'08')	ABNORMAL TERMINATION: Internal error.
	12 (X'0C')	BRANCH ENTRY ADDRESS FOR ASYNCHRONOUS OPERATIONS MANAGER NOT FOUND IN SSSVT: Internal error.

AOM Tape Library Return and Reason Codes

Table 4 describes AOM tape library return and reason codes.

Table 4. AOM Tape Library Return and Reason Codes

Return Code	Error Type	Reason Code	Description
X'00'	Success	X'0000'	Request was successful
X'04'	Warning	X'0401'	Affinity list is in pool list order (not ordered by hardware).
		X'0402'	Asynchronous order was deleted by REQTYPE(DELETE).
		X'0403'	REQTYPE(LISTEN) and a listen request already exists for the library.
X'08'	Invalid request	X'0800'	Unsupported request type.
		X'0801'	SERVL eye-catcher invalid.
		X'0802'	Store into SERVL program checked.
		X'0805'	Invalid category code (SERVCAT1).
		X'0808'	UCB or Device number is required for the request.
		X'0809'	Library-ID is required for the request.
		X'0810'	Exit routine is required for the request.
X'0C'	Library or Device unavailable	X'0C00'	There are no libraries in the system.
		X'0C01'	Libraries in the system, but configuration not 'SET'.
		X'0C02'	No devices available for library I/O.
		X'0C03'	Request requires I/O and library is OFFLINE.
		X'0C04'	Request requires I/O and library is not operational.
		X'0C05'	LSSCB not found for library specified by SERVLID.
		X'0C06'	LSSDP not found for UCB or device number.
X'10'	System Service failure	X'1001'	Unable to get LOCAL lock. SERVFRC contains the return code causing the failure.
		X'1002'	GETMAIN failure. SERVFRC contains the return code causing the failure.
		X'1003'	Unable to get CMS lock. SERVFRC contains the return code causing the failure.

Table 4. AOM Tape Library Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description
		X'1004'	FREEMAIN failure. SERVFRC contains the return code causing the failure.
		X'1005'	UCBPIN failure.
		X'1006'	UCBLOOK failure.
		X'1008'	UCB UNPIN failure.
X'14'	Permanent I/O error	X'1401'	Asynchronous operation lost by hardware.
		X'14xx'	I/O error. xx=AOMDRVR return code.
X'18'	AOM service failure	X'1801'	AOMQUE create failed.
		X'1802'	AOMQUE locate failed.
		X'1803'	AOMQUE delete failed.
X'1C'	AOM FRR entered.	X'1C00'	IGX00044 FRR retry successful.
		X'1C01'	IGX00044 FRR retry failed.

Chapter 3. BDAM Diagnostic Aids

This chapter provides a description of messages and codes that are issued by Basic Access Method (BDAM) modules to help you diagnose BDAM problems.

Messages and Codes Issued by BDAM Modules

Table 5 shows the messages and completion codes that BDAM modules issue. For the meaning of a specific code, see *z/OS MVS System Codes*. For the meaning of a specific message (shown in parentheses), see the applicable system messages volume:

- *z/OS MVS System Messages, Vol 1 (ABA-AOM)*
- *z/OS MVS System Messages, Vol 2 (ARC-ASA)*
- *z/OS MVS System Messages, Vol 3 (ASB-BPX)*
- *z/OS MVS System Messages, Vol 4 (CBD-DMO)*
- *z/OS MVS System Messages, Vol 5 (EDG-GFS)*
- *z/OS MVS System Messages, Vol 6 (GOS-IEA)*
- *z/OS MVS System Messages, Vol 7 (IEB-IEE)*
- *z/OS MVS System Messages, Vol 8 (IEF-IGD)*
- *z/OS MVS System Messages, Vol 9 (IGF-IWM)*
- *z/OS MVS System Messages, Vol 10 (IXC-IZP)*

Table 5. Messages and Codes Issued by BDAM Modules

System Completion Code	Message Number
001	IEC020I
013	IEC141I
014	IEC208I
020	IEC138I
135	IEC903I
139	IEC905I
235	IEC902I
239	IEC904I
335	
339	
435	
535	

© Copyright IBM Corp. 1979, 2002

Chapter 4. Compression Management Services Diagnostic Aids

This chapter explains the following diagnostic aids, which are provided for Compression Management Services (CMP):

- Messages that are issued during compression
- LOGREC information
- Common return and reason codes for activation failures
- Common return and reason codes for allocation failures
- CMP return and reason codes

Compression Services

The compression manager startup function is invoked during the SMS2 address space initialization. The compression services global control-block structure is initialized at SMS startup. Bit DFACMPSC is set in the DFA. SMS initializes compression services if the CVTCMPSC bit is on and the compression global structure is not built completely or not available.

Compression Services Messages

CMP001I **DFSMS COMPRESSION SERVICES AVAILABLE**

Compression services is available (CVTCMPSC on). The compression services structure is initialized without error.

CMP002I **LIMITED DFSMS COMPRESSION SERVICES AVAILABLE**

Compression services completed only partial initialization. Only existing compressed format data sets will be processed. No new compressed format data sets will be allocated.

The most common cause of this condition is when not all DBBs are successfully loaded due to an error.

CMP003I **DFSMS COMPRESSION ACTIVATION FAILED**

Compression services is not available. The compression services global control-block structure could not be built, or an unexpected abend occurred during validation of the structure.

The most common cause of this condition is when CMP cannot access the SYS1.DBBLIB data set. The user should make sure SYS1.DBBLIB exists and is cataloged. SYS1.DBBLIB is built during DFSMS installation.

There might be a LOGREC entry for the CMP003I message with the return and reason codes for the error.

© Copyright IBM Corp. 1979, 2002

LOGREC Information

CMP writes a symptom record in the LOGREC for two conditions:

- At the end of each module, CMP SVC writes a LOGREC entry if it detects that the return code is greater than or equal to RC_WARN (X'08') and the called module has not recorded.
- When the recovery routine gets control during an unexpected exception condition, it will record a LOGREC entry before taking the dump. This is to ensure that some serviceability information is recorded in LOGREC, even if the dump is not successful.

The LOGREC contains this recorded information:

- The failing module identifier
- The return and reason codes
- The control block(s) that are available at the time of record
- The module that actually performed the record (CMPSVCSY)

Return Code Format

As Table 6 shows, the Compression Management Services return code is four bytes. The high-order three bytes are zero. The low-order byte contains the return code.

For example:

RC = 000000**XX**

In the example, **XX** is the return code in hexadecimal format.

Table 6. Compression Management Services Return Code Format

Return Code	Meaning	Name
X'00000000'	Successful	RC_SUCC
X'00000004'	Informational	RC_INFO
X'00000008'	Warning	RC_WARN
X'0000000C'	Parameter error	RC_PARM
X'00000010'	Caller error	RC_CALR
X'00000014'	Environmental error	RC_ENVR
X'00000018'	I/O error	RC_IOER
X'0000001C'	Media error	RC_MEDE
X'00000020'	Data set logical error	RD_DSLE
X'00000024'	System error	RC_SEVE

Reason Code Format

The Compression Management Services reason code is four bytes.

Byte 1 Function identifier

X'5F' indicates Compression Management Services.

Byte 2 Module identifier

See Table 7 for a list of module identifiers.

Bytes 3 and 4
Reason code that the describes the specific error

See Table 10 on page 18 for a complete list of CMP reason codes.

Table 7. CMP Reason Code Module Identifiers

Identifier Code	Module Name
X'01'	CMPSVCAL (allocation)
X'02'	CMPSVCOP (open)
X'03'	CMPSVCCM (compress)
X'04'	CMPSVCDE (decompress)
X'05'	CMPSVCCL (Close)
X'07'	CMPSVCPC (PC router)
X'08'	CMPSVCBD (Build Dictionary)
X'09'	CMPSVCPD (PDSE environment)
X'10'	CMPSVCRC (CMP recovery)
X'11'	CMPSVCIN (Interrogation)
X'12'	CMPSVCSA (Sampling)
X'13'	CMPSVCSY (Recording information)

CMP Activation Return and Reason Codes

Table 8. CMP Activation Return and Reason Codes

Return Code	Reason Code	Explanation
X'08'	X'0820' (DEC 2080)	RS_MVS_CMS_NOTAVAIL
X'08'	X'0821' (DEC 2081)	RS_DFSMS_CMS_NOTAVAIL
X'08'	X'0822' (DEC 2082)	RS_GLBL_PARTIAL_CONSTRUCTED
X'08'	X'0823' (DEC 2083)	RS_HW_INSTR_ABSENT
X'08'	X'0825' (DEC 2085)	RS_NOT_ALL_FREED

Refer to Table 10 on page 18 for a complete explanation of these reason codes.

CMP Allocation Return and Reason Codes

The following compression services return codes and reason codes might be found in SMS message IGD17161I or IGD17162I.

Table 9. CMP Allocation Return and Reason Codes

Return Code	Reason Code	Explanation
X'0C'	X'083E' (DEC 2110)	RS_NOT_SUPPORTED
X'0C'	X'083F' (DEC 2111)	RS_NO_BENEFIT
X'0C'	X'0840' (DEC 2112)	RS_CISIZE_TOO_SMALL

Refer to Table 10 for a complete explanation of these reason codes.

CMP Return and Reason Codes

The following is a complete list of Compression Management Services return and reason codes.

Table 10. CMP Return and Reason Codes

Return Code	Reason Code	Explanation
RC_INFO X'04'	**Reason Codes**	
X'04'	DEC 2048	RS_SAMPLING_STILL The compression operation has finished successfully. CMP is still in the sampling phase. The input data has not been compressed. The output buffer contains an image copy of the input buffer. The compression operation has finished successfully.
X'04'	DEC 2049	RS_DICT_TOKEN_DERIVED The compression operation has finished successfully. The Dictionary Token has been determined and returned in the storage pointed to by the DICTIONARY_TOKEN_PTR. The input data has not been compressed. The output buffer contains an image copy of the input buffer.
X'04'	DEC 2050	RS_COMPRESSION_REJECTED The compression operation has finished successfully. A rejection dictionary token has been created and returned in the field pointed to by the DICTIONARY_TOKEN_PTR. The caller should stop calling the CMS_COMPRESS service because the statistics from interrogation and sampling indicate that this is not a good compression candidate. The output buffer contains an image copy of the input buffer.

Table 10. CMP Return and Reason Codes (continued)

Return Code	Reason Code	Explanation
X'04'	DEC 2051	RS_TARGET_X_COMPRESSED The compression operation has finished successfully. Only part of the input buffer is compressed. The output buffer has been filled during the compression operation but the input buffer has not been exhausted. The caller may call CMPSVCCM again to compress the remaining data in the input buffer. Be sure to set the CMPSC_BITNUM to zero and reset the CMPSC_TARGETLEN and CMPSC_TARGETADDR fields.
X'04'	DEC 2052	RS_TARGET_EXHAUSTED The decompression operation has finished successfully. The target contains decompressed data. However, not all the source data has been decompressed due to the running out of space in the TARGET.
X'04'	DEC 2053	RS_DATA_GREW The compression operation has finished successfully. The output buffer has been filled during the compression operation. However, the actual number of *compressed* bytes is greater than the original number of processed input bytes. CMP has copied the CMPSC_TARGETLEN number of bytes from the beginning of the input buffer into the output buffer. The CMPSC_TARGETLEN has been decremented by CMPSC_SOURCELEN. The CMPSC_TARGETADDR has been incremented by CMPSC_SOURCELEN.

RC_WARN X'08' Reason Codes

X'08'	DEC 16	RS_TEXT_GET_FAIL MPSTCGI unable to subdivide storage area for TEXT_BLK.
X'08'	DEC 32	RS_STOR_RELEASE_FAIL MPSTCGI's STORAGE RELEASE failed.
X'08'	DEC 48	RS_DEALL_DD_FAIL MPSTCGI's SYS1.DBBLIB deallocation failed.
X'08'	DEC 2080	RS_MVS_CMS_NOTAVAIL The MVS/Compression Services is not available on the system. No compression processing can be performed unless a version of MVS/ESA™ which supports compression is installed on this system.

Table 10. CMP Return and Reason Codes (continued)

Return Code	Reason Code	Explanation
X'08'	DEC 2081	RS_DFSMS_CMS_NOTAVAIL DFSMS Compression Management Services is not available on this system. Make sure that SMS is started. If SMS is started, refer to the SMF records to find out what might have happened during DFSMS Compression Management Services initialization that caused it to fail. User Response: • See CMP003I message explanation • Catalog SYS1.DBBLIB and IPL Possible causes: • The SYS1.DBBLIB data set is not cataloged. • Compression services will not start if it cannot: – Open the SYS1.DBBLB – Read the directory – Read the members
X'08'	DEC 2082	RS_GLBL_PARTIAL_CONSTRUCTED The global structure is partially created. No new allocation for compressed format data sets is allowed.
X'08'	DEC 2083	RS_HW_INSTR_ABSENT The hardware compression instruction is not on this system.
X'08'	DEC 2085	RS_NOT_ALL_FREED CMP has tried to perform the necessary resource cleanup. However, some of the resources may not be successfully freed.

RC_PARM X'0C' Reason Codes

X'0C'	DEC 2110	RS_NOT_SUPPORTED The data set is not eligible for compression because it is not one of the supported types. Only extended SAM data sets and extended VSAM KSDS data sets are supported.

Table 10. CMP Return and Reason Codes (continued)

Return Code	Reason Code	Explanation
X'0C'	DEC 2111	RS_NO_BENEFIT The data set is not eligible for compression. The data set attributes show that the data set will not benefit in space utilization or number of I/O operations from the compression. Compression might incur unnecessary overhead for its data processing. The requirements for compression are as follows: • The data set must already have met the requirements for Extended Format. • Compression Management Services requires the data set to have primary allocation of at least 5 megabytes in order to be allocated as a compressed data set. • Compression Management Services requires the data set to have a primary allocation of 8 MB if no secondary allocation is specified, in order to be allocated as a compressed data set. • Compression Management Services requires that the data set have a minimum record length of forty (40) bytes, not including the key offset or key length. **Note:** For VSAM KSDS, the CMS requirement to have a primary allocation of 5 MB is due to the amount of sampling needed to develop a dictionary token. The 5 MB allocation MUST be for the data component only. If the amount is specified at the cluster level, then the index component will use a portion of the 5 megabyte allocation. This will result in the data set not meeting the CMS space requirement, and the data set will not be eligible for compression processing. The above is also true for the 8 MB primary allocation if no secondary is specified.
X'0C'	DEC 2112	RS_CISIZE_TOO_SMALL The data set is not eligible for compression be cause the CISIZE is not big enough to accommodate one KSDS data record in its uncompressed format. The CISIZE for a nonspanned compressed KSDS must be at least 10 bytes larger than the maximum record length. The VSAM data set is not eligible for compression because the CISIZE is not large enough to contain the data record's key field. When a data record spans CIs, the record's key field must be contained within the first CI. The CISIZE for a spanned compressed KSDS must be at least fifteen (15) bytes larger than the key field. Key field: key length and displacement from the beginning of the record.

Table 10. CMP Return and Reason Codes (continued)

Return Code	Reason Code	Explanation
RC_CALR X'10'	**Reason Codes**	
X'10'	DEC 2140	RS_BAD_OPEN_TOKEN Invalid CM_OPEN_TOKEN. Sequence number in the token does not match the one in the control block, or the CMSOP block cannot be located in the CMSAS_CM_OP_CHAIN.
X'10'	DEC 2141	RS_OPERAND_NOT_PROVIDED A request to compress or decompress but some of the operands are not provided.
X'10'	DEC 2142	RS_NULL_DICT_TOKEN A request to decompress the data but the input Dictionary Token is null.
X'10'	DEC 2143	RS_MISSING_PARM A request to decompress the data but the input Dictionary Token is null.
X'10'	DEC 2144	RS_INVALID_PARAMETER Either both JFCB pointer and the AMDSB pointer are given to CMPSVCAL, or neither of the pointers is given.
X'10'	DEC 2145	RS_COMP_REJECTED_DS The input dictionary token indicates that the data set is a rejected candidate for compression.
X'10'	DEC 2146	RS_BAD_OPERAND_PROVIDED A request to compress or decompress but the target or source is not large enough to hold even one compression symbol.
X'10'	DEC 2147	RS_UNDEFINED_OPCODE This operation code specified on the CMPSVCPC interface is undefined.
X'10'	DEC 2148	RS_INCOMPLETE_OPEN_TOKEN The CMS_OPEN_TOKEN has a sequence number of X'FFFFFFFF' which indicates that NOT all the CMP control structures are in place.
X'10'	DEC 2149	RS_NULL_OPEN_TOKEN The CMS_OPEN_TOKEN is null when it should have a value. This reason code may be returned by CMPSVCCM, CMPSVCDE, or CMPSVCCL when the input open token is null.

Table 10. CMP Return and Reason Codes (continued)

Return Code	Reason Code	Explanation
RC_ENVR X'14'	**Reason Codes**	
X'14'	DEC 2180	RS_LACK_OF_STORAGE CMP cannot obtain storage for its control structure or dictionaries.
X'14'	DEC 2181	RS_TREE_GET_FAILED CMP cannot obtain storage for the LZ-tree.
X'14'	DEC 2182	RS_DICT_GET_FAILED CMS cannot obtain storage for the hardware dictionaries.
X'14'	DEC 2183	RS_TEXT_GET_FAILED CMP cannot obtain storage for the CMS_TEXT_BLK. CMS_TXT_BLK contains the UDCT data format that is required by the CMPDBCTB routine.
RC_IOER X'18'	No Reason Codes	
RC_MEDE X'1C'	No Reason Codes	
RC_DSLE X'20'	No Reason Codes	
RC_SEVE X'24'	**Reason Codes**	
X'24'	DEC 1016	RS_SET_ESTAE_FAIL MPSTCGI is unable to create its ESTAEX environment.
X'24'	DEC 1032	RS_STOR_OBTAIN_FAIL MPSTCGI's STORAGE OBTAIN failed to get the global area needed to contain the CMS global control blocks.
X'24'	DEC 1048	RS_GLBL_GET_FAIL MPSTCGI unable to subdivide storage area for GLBL_BLK.
X'24'	DEC 1064	RS_CNTL_GET_FAIL MPSTCGI unable to subdivide storage area for CNTL_BLK.
X'24'	DEC 1080	RS_SET_TOKEN_FAIL MPSTCGI's save of CMS's Global token failed.
X'24'	DEC 1096	RS_GET_TOKEN_FAIL MPSTCGI's get of CMS's Global token failed.

Table 10. CMP Return and Reason Codes (continued)

Return Code	Reason Code	Explanation
X'24'	DEC 1112	RS_ALLOC_DD_FAIL MPSTCGI's SYS1.DBBLIB allocation failed.
X'24'	DEC 1128	RS_OPEN_DIR_FAIL MPSTCGI's OPEN of SYS1.DBBLIB for BSAM input failed.
X'24'	DEC 1144	RS_OPEN_MEM_FAIL MPSTCGI's OPEN of SYS1.DBBLIB for BPAM input failed.
X'24'	DEC 1160	RS_BUFFER_GET_FAIL MPSTCGI's STORAGE OBTAIN of input buffer failed.
X'24'	DEC 1176	RS_DIR_FORMAT_ERROR MPSTCGI received unexpected EOF during directory read.
X'24'	DEC 1192	RS_DIR_READ_ERROR MPSTCGI received permanent error during directory read.
X'24'	DEC 1208	RS_INDEX_GET_FAIL MPSTCGI unable to subdivide storage area for CNTL index.
X'24'	DEC 1224	RS_LOW_OBTAIN_FAIL MPSTCGI unable to acquire storage below the line.
X'24'	DEC 1228	RS_MEMBER_FIND_ERROR MPSTCGI unable to FIND member pointed to by directory.
X'24'	DEC 1232	RS_MEMBER_READ_ERROR MPSTCGI unable to READ members control card.
X'24'	DEC 1240	RS_NO_MEMS_FOUND MPSTCGI no members found in SYS1.DBBLIB to process.
X'24'	DEC 4096	RS_BAD_OPEN_BLOCK The eye-catcher in the CM_OPEN control block is bad.
X'24'	DEC 4097	RS_DBB_NOT_LOADED_IO The dictionary cannot be built in storage. One or more of the dictionary blocks were not loaded at startup due to I/O error.
X'24'	DEC 4098	RS_DBB_NOT_LOADED_LOGIC The dictionary was not built in storage because some of the DBBs were not loaded at startup due to logic error.

Table 10. CMP Return and Reason Codes (continued)

Return Code	Reason Code	Explanation
X'24'	DEC 4099	RS_BAD_DICTIONARY The dictionary has some format error. The MVS/ESA Compression Services cannot operate properly. The MVS/ESA return code will be propagated to the caller in the low order 2 bytes of the reason code.
X'24'	DEC 4100	RS_BAD_DICT_TOKEN Invalid Dictionary Token. The Dictionary Building Block(s) cannot be located in the CMS_CNTL blocks. Or, the flag in the dictionary token indicates that the token is a rejection token but the DBBs are not NULL.
X'24'	DEC 4101	RS_INTERNAL_LOGIC_ERROR Internal CMS logic error occurred.
X'24'	DEC 4102	RS_OBTAIN_LOCK_FAILED Cannot obtain the local lock.
X'24'	DEC 4103	RS_LOCASCB_FAILED Cannot locate the ASCB.
X'24'	DEC 4104	RS_NULL_FOOTPRINT_PTR In CMPSVCRC, the footprint pointer passed in is null.
X'24'	DEC 4105	RS_SET_ASTOKEN_FAILED Cannot save the CMSAS token via the SSF IGWFTOKM macro.
X'24'	DEC 4106	RS_SET_FRR_FAILED Cannot set the recovery environment. FRR is set in CMPSVCOP, CMPSVCBD, and CMPSVCCL where they either GETMAIN or FREEMAIN storage and a local lock is required.
X'24'	DEC 4107	RS_TREE_INIT_FAILED CMPDBCTI returns a nonzero return code. It failed in initializing the LZ-tree.
X'24'	DEC 4108	RS_TREE_BUILD_FAILED CMPDBCTB returns a nonzero return code. It failed in inserting text blocks into the LZ-tree.
X'24'	DEC 4109	RS_TREE_CONV_FAILED CMPDBCTI returns a nonzero return code. It failed in converting the LZ-tree into the hardware compression format.
X'24'	DEC 4110	RS_BAD_DBB The DBB is bad. CMPSTCGI marked it as invalid in the CMS_CNTL block.

Table 10. CMP Return and Reason Codes (continued)

Return Code	Reason Code	Explanation
X'24'	DEC 4111	RS_BAD_AS_TOKEN The CMSAS token retrieved via IGWFTOKM does not point to a valid CMSAS block.
X'24'	DEC 4112	RS_BAD_TASK_TOKEN The CMSTASK token retrieved via IGWFTOKM does not point to a valid CMSTASK block.
X'24'	DEC 4113	RS_BAD_AS_BLOCK The CMSAS block has a valid eye-catcher, but some other information in it is suspicious.
X'24'	DEC 4114	RS_NULL_AS_TOKEN A null CMSAS token is returned from IGWFTOKM when a non-null CMSAS is expected.
X'24'	DEC 4115	RS_NULL_TASK_TOKEN A null CMSTASK token is returned from IGWFTOKM when a non-null CMSTASK is expected.
X'24'	DEC 4116	RS_RECOVERY_ENTERED This reason code is set whenever the recovery routine is entered.
X'24'	DEC 4117	RS_SET_TASKTOKEN_FAILED Cannot save the CMSTASK token via the SSF IGWFTOKM macro.
X'24'	DEC 4118	RS_BAD_DS_BLOCK The CMSDS block has an invalid eye-catcher.
X'24'	DEC 4119	RS_BAD_DS_PTR The pointer to CMSDS in the CMSOP block is null but the file does not have a dictionary token yet.
X'24'	DEC 4120	RS_BAD_DI_BLOCK The DI block is bad. It does not have a valid eye-catcher.
X'24'	DEC 4121	RS_CMSDS_IN_USE The caller requested to open an EMPTY data set but there exists a CMSDS that has a nonzero count.

Chapter 5. DADSM/CVAF Diagnostic Aids

This chapter provides the following major diagnostic aids for DADSM/CVAF:
- CVAF module-to-function cross-reference
- CVAF Generalized Trace Facility trace of events
- CVAF debug trap options
- CVAF-VTOC index error message
- CVAF and DADSM dump contents
- CVAF status codes in the CVPL
- Format-7 DSCB
- DADSM module-to-function cross-reference
- DADSM return codes, status codes, and diagnostic information

CVAF Module-to-Function Cross-Reference

In Table 11, the characters *xx* or *xxx* are used to represent any characters.

For a description of how to use the CVAF function and subfunction as modifier keywords, see the section describing DADSM/CVAF modifier keywords in *z/OS DFSMSdfp Diagnosis Guide*.

Table 11. CVAF Module-to-Function Cross-Reference

Module	Function	Subfunction
ICVCM*xxx*	CM	
ICVDM*xxx*	DM	
ICVDS*xxx*	DS	
ICVFI*xxx*	FI	
ICVIXA*xx*	IX	ADD
ICVIXC*xx*	IX	SRCH
ICVIXD*xx*	IX	DEL
ICVIXS*xx*	IX	SRCH
ICVSMS00	SM	
ICVLSN00	LS	
ICVVOL*xx*	VO	
ICVVRFM0	VR	

CVAF GTF Trace of Events

Generalized Trace Facility (GTF) permits tracing the following CVAF events:
- All output of VTOC index I/O operations
- All output of VTOC I/O performed by CVAF
- All modifications of a VTOC index and space maps

The CVAF program CVAFGTF sets CVAF tracing options. Ending CVAFGTF ends only the interaction with the operator, not the tracing options. With the options set, CVAF continues to create trace records until the options are reset.

© Copyright IBM Corp. 1979, 2002

DADSM/CVAF Diagnostic Aids

To invoke CVAF event tracing (program CVAFGTF), take the following steps:

1. Start GTF. For more information, see *z/OS MVS Diagnosis: Tools and Service Aids*.
2. Request the trace option USR.
3. Start CVAFGTF by entering the following command from the operator's console:

   ```
   START CVAFTR
   ```

4. Supply a specific trace command in response to the following console message that CVAFGTF issues:

   ```
   CVGTF01 ENTER CVAF TRACE COMMAND Sxxx, Pxxx, ENDT, OR DBUG
   ```

   The trace commands and their meaning are as follows:

   S*xxx* Starts CVAF trace on device *xxx*.

   P*xxx* Stops CVAF trace on device *xxx*.

   DBUG Sets CVAF debug options. See "CVAF Debug Trap Options" on page 33.

   ENDT Ends CVAFGTF.

   CVAFGTF also issues the following console messages:

   ```
   CVGTF03 - INVALID DEVICE/ADDRESS xxx
   ```

   ```
   CVGTF02 - CVAF TRACE STARTED/STOPPED ON xxx
   ```

   CVGTF04 and CVGTF05 are issued if the DEBUG command was entered, as specified in "CVAF Debug Trap Options" on page 33.

   If GTF is not active, GTF tracing by CVAF does not occur. If GTF is later invoked with the USR option and CVAFGTF options are in effect, CVAF tracing occurs.

   If the unit that is selected for tracing is a shared direct access storage device, then all sharing systems should trace CVAF activity to this device. You can accomplish this by setting the CVAF options through the CVAFTR program and invoking GTF on each of the sharing systems.

CVAF Format of GTF Trace Records

All GTF trace records have the same format, shown in Table 12.

Table 12. Format of GTF Trace Records

Offset	Length	Description
0 (X'00')	2	Length of trace record
2 (X'02')	2	Always zero
4 (X'04')	1	Application ID; FF for trace record
5 (X'05')	1	Format ID; *xx* for CVAF events
6 (X'06')	8	Time stamp
14 (X'0E')	2	Event ID; 0FAF-0FB8 for CVAF events
16 (X'10')	*nn*	Trace data; format depends on CVAF event

Use IPCS to print trace records. For information on the use of IPCS, see *z/OS MVS Diagnosis: Tools and Service Aids*.

CVAF GTF Trace Events

Ten different CVAF events are traced. The following paragraphs describe the trace data for each event. For the events numbered 3 through 10, the trace record is created when the map is updated in virtual storage. The VTOC index map is not updated until the record is written.

1. **Write DSCB**

   Recorded when CVAF writes a DSCB to the VTOC.

   Restriction: CVAF does not write format-0 and format-4 DSCBs. Therefore, no trace record is created for them.

   Event ID: 0FAF

   Table 13 describes the format of trace data for a write DSCB event.

Table 13. Format of Trace Data for Write DSCB

Offset	Length	Description
0 (X'00')	6	Volume serial number in EBCDIC
6 (X'06')	5	CCHHR of DSCB written
11 (X'0B')	1	Reserved
12 (X'0C')	140	140-byte DSCB

2. **Write VIR**

   Recorded when a VTOC index record (VIR) is written to the VTOC index. Each GTF trace record is 128 bytes long. Sixteen trace records are written for each 2048-byte VIR.

   Event ID: 0FB0

   Table 14 on page 30 describes the format of trace data for a write VIR event.

Table 14. Format of Trace Data for Write VIR

Offset	Length	Description
0 (X'00')	6	Volume serial number in EBCDIC
6 (X'06')	1	Sequence number for this 128-byte section of the VIR, starting with 0
7 (X'07')	4	Relative byte address (RBA) of the VIR
11 (X'0B')	1	Reserved
12 (X'0C')	128	128-byte section of the VIR

3. **Add Index**

   Recorded when a data set name and a CCHHR pointer are added to the VTOC index.

   Event ID: 0FB1

   Table 15 describes the format of trace data for an add index event.

Table 15. Format of Trace Data for Add Index

Offset	Length	Description
0 (X'00')	6	Volume serial number in EBCDIC
6 (X'06')	5	CCHHR of DSCB
11 (X'0B')	1	Reserved
12 (X'0C')	44	Data set name

4. **Delete Index**

   Recorded when a data set name is deleted from the VTOC index.

   Event ID: 0FB2

   Table 16 describes the format of trace data for a delete index event.

Table 16. Format of Trace Data for Delete Index

Offset	Length	Description
0 (X'00')	6	Volume serial number in EBCDIC
6 (X'06')	5	CCHHR of DSCB
11 (X'0B')	1	Reserved
12 (X'0C')	44	Data set name

5. **Allocate Space**

   Recorded when space on the volume is allocated from the volume pack space map (VPSM). If the space request fails, the trace record is not written.

   Event ID: 0FB3

   Table 17 on page 31 describes the format of trace data for an allocate space event.

Table 17. Format of Trace Data for Allocate Space

Offset	Length	Description
0 (X'00')	6	Volume serial number in EBCDIC
6 (X'06')	5	Unused
11 (X'0B')	1	Reserved
12 (X'0C')	*nn*	Table of allocated extents(see note)

Note: The table format is 4 bytes of number of extents, followed by the extent descriptors. Each extent descriptor contains a 4-byte RTA1 (beginning of extent) and a 4-byte RTA2 (end of extent + 1).

6. **Release Space**

   Recorded when space on the volume is released from the volume pack space map (VPSM).

   Event ID: 0FB4

   Table 18 describes the format of trace data for a release space event.

Table 18. Format of Trace Data for Release Space

Offset	Length	Description
0 (X'00')	6	Volume serial number in EBCDIC
6 (X'06')	5	Unused
11 (X'0B')	1	Reserved
12 (X'0C')	*nn*	Table of released extents(see note)

Note: The table format is: 4 bytes of number of extents, followed by the extent descriptors. Each extent descriptor contains a 4-byte RTA1 (beginning of extent) and a 4-byte RTA2 (end of extent + 1).

7. **Allocate VTOC Index Space**

   Recorded when a VTOC index record is allocated and marked in the VTOC index map (VIXM) as available for allocation.

   Event ID: 0FB5

   Table 19 describes the format of trace data for an allocate VTOC index space event.

Table 19. Format of Trace Data for Allocate VTOC Index Space

Offset	Length	Description
0 (X'00')	6	Volume serial number in EBCDIC
6 (X'06')	1	Unused
7 (X'07')	4	RBA of VIR allocated

8. **Release VTOC Index Space**

   Recorded when a VTOC index record is released and marked in the VIXM as unavailable for subsequent allocation.

   Event ID: 0FB6

   Table 20 describes the format of trace data for a release VTOC index space event.

Table 20. Format of Trace Data for Release VTOC Index Space

Offset	Length	Description
0 (X'00')	6	Volume serial number in EBCDIC
6 (X'06')	1	Unused
7 (X'07')	4	RBA of VIR deallocated

9. **Allocate VTOC Space**

   Recorded when a format-0 DSCB is allocated; the VTOC map of DSCBs (VMDS) is updated to indicate that the DSCB is allocated.

   Event ID: 0FB7

   Table 21 describes the format of trace data for an allocate VTOC space event.

Table 21. Format of Trace Data for Allocate VTOC Space

Offset	Length	Description
0 (X'00')	6	Volume serial number in EBCDIC
6 (X'06')	5	CCHHR of allocated format-0 DSCB

10. **Release VTOC Space**

    Recorded when a DSCB (all formats) is released and marked in the VMDS as unallocated.

    Event ID: 0FB8

    Table 22 describes the format of trace data for a release VTOC space event.

Table 22. Format of Trace Data for Release VTOC Space

Offset	Length	Description
0 (X'00')	6	Volume serial number in EBCDIC
6 (X'06')	5	CCHHR of DSCB marked as unallocated

CVAF Debug Trap Options

The CVAF table contains a debug trap area. The address of the CVAF table is in field CVTCVAF at offset 12 (X'0C') in the VS1/VS2 common extension to the communication vector table (CVT). The debug trap area can be used to acquire additional information for diagnosing a problem in CVAF.

At offset 12 (X'0C') into the CVAF table is the flag byte, CVFDTFG, which contains the bits that can be set to invoke debugging options of CVAF. The bits and their names are:

Bit 0	CVFSVWA
Bit 1	CVFENT
Bit 2	CVFEXIT
Bit 3	CVFJOB
Bit 4	CVFLOOP
Bit 5-7	Reserved

At offset 15 (X'0F') into the CVAF table is the 5-byte field CVFSTPNM. This field can be set to the last 5 bytes of a CVAF module name for which a debug trap is to be set. (The first 3 bytes of the module name are implicitly ICV.)

At offset 20 (X'14') into the CVAF table is the 8-byte field CVFJOBNM. This field can be set to the name of a job for which a debug trap is to be set. The job name must be padded on the right with blanks if the job name is less than 8 characters.

You can modify the CVAF table using the CVAFGTF program that follows.

There are two debug trap options:

1. Save CVAF module work areas

   As each module in CVAF gains control, a module work area is allocated for its use. The module work area is released when the module exits. That module work area may be allocated to a subsequent module that gains control. A few CVAF modules ICVCMW00 (and entry point ICVCMG00), ICVCMW01 (and entry point ICVCMG01), and ICVLSN00 do not obtain module work areas. The work area used by CVAF module CVAFGTF cannot be saved.

   It is desirable to make available for inspection a dump of all CVAF module work areas for all CVAF modules that gain control. Set the CVFSVWA flag bit to 1 to prevent release of a CVAF module work area when a module returns control to its caller. Subsequent module work areas will be allocated so as not to overlay work areas of modules previously in control.

   Note: All of the saved module work areas will be released when CVAF returns to its caller.

   If the CVFJOB flag bit is set to 0, the facility that saves CVAF module work areas is active for all CVAF calls. If the CVFJOB bit is set to 1 and CVFJOBNM is set to a job name, the CVAF module work area is saved only when the job name is the same as that stored in CVFJOBNM.

2. Trap module entry/exit

   You might want to stop processing either when a specific CVAF module is given control (entry) or when it returns control to its caller (exit). Set the CVFENT bit to 1 to set such a trap on entry to a module. set the CVFEXIT bit to 1 to set a trap on exit from a module. The last 5 bytes of the module name for which a trap is set are stored in the CVFSTPNM field of the CVAF table.

   The entry trap logic is contained in module ICVCMW00 when that module is called to allocate a module work area. The exit trap logic is contained in module ICVCMW01 when that module is called to release a module work area.

   Modules ICVCMW00 (and entry point ICVCMG00), ICVCMW01 (and entry point ICVCMG01), ICVLSN00, and CVAFGTF cannot be trapped this way.

   The trap can be either a program check or a write-to-operator-with-reply (WTOR) followed by a loop that waits for a reply.

   If the CVFLOOP bit is set to 0, the trap is a program check (0C1). If the CVFLOOP bit is set to 1, the trap is a write-to-operator (CVAF LOOP - jobname, modulename. Reply to continue.) followed by a loop until the reply is received. Before the reply is received, you can alter the CVFSTPNM field from the operator's console to identify another module. Reply xx,y to continue. (xx is the reply number.)

   The trap can be for all CVAF calls or for only the CVAF calls issued by a job with a specified job name. Set the CVFJOB bit to 0 to set the trap for all jobs. Set the CVFJOB bit to 1 to set the trap for only the job whose name is stored in the CVFJOBNM field of the CVAF table.

To invoke the CVAF debug trap options (program CVAFGTF), do the following procedure:

1. Enter the following command from the operator's console:

   ```
   START CVAFTR
   ```

2. CVAFGTF issues a command message prompting the operator to supply a specific trace command. The message is:

   ```
   CVGTF01 ENTER CVAF TRACE COMMAND Sxxx, Pxxx, ENDT, OR DBUG
   ```

3. Enter command DBUG. Commands Sxxx and Pxxx are explained in step 4 on page 28.
4. CVAFGTF then issues a command message prompting the operator to supply the debug trap options:

   ```
   CVGTF04 ENTER CVAF DEBUG PARAMETERS
   ```

5. Reply to this message, using the following format:

   ```
   SET/RESET,SAVE/NOSAVE,jobname,modname,ENTRY,EXIT,LOOP/PGMCHK
   ```

   The first positional parameter must be:

   SET All previously set options are disabled, and the following parameters are inspected to determine the options desired.

   RESET Subsequent parameters are ignored, and all debug trap options are removed.

The second positional parameter must be:

SAVE The save CVAF module work-area option is enabled. If *jobname* is not supplied as the third parameter, SAVE is active for all jobs. If *jobname* is supplied as the third parameter, SAVE is active only for that job.

NOSAVE The save CVAF module work-area option is disabled or not enabled.

omitted Indicated by a comma or by the first parameter being the end of the reply to CVGTF04. Operation is the same as if NOSAVE were specified.

The third positional parameter must be:

jobname Specifies a 1- to 8-character job name. This job name will be used to qualify the save CVAF module work area option, if SAVE is specified as the second parameter. The job name will also be used to qualify the module entry/exit trap if ENTRY is specified as the fifth parameter and EXIT is specified as the sixth parameter.

omitted Indicated by a comma, or by the first or second parameter being the end of the reply to CVGTF04.

The fourth positional parameter must be:

modname Specifies the last 5 characters of a CVAF module name. The first 3 characters of the module name are assumed to be ICV. For example, if DSD01 is supplied as the fourth parameter, the CVAF module will be ICVDSD01. This module will be used for the module entry/exit trap. If **modname** is supplied, ENTRY must be supplied as the fifth parameter, or EXIT must be supplied as the sixth parameter, or both.

omitted Indicated by a comma or by the third or earlier parameter being the end of the reply to CVGTF04. The module entry/exit trap option is not enabled.

The fifth positional parameter must be:

ENTRY Specifies that the module entry trap option is to be enabled. A module name must be specified as the fourth parameter, or ENTRY will be ignored. If *jobname* is supplied as the third parameter, the module entry trap is enabled only for the specified job. If PGMCHK is specified as the seventh parameter (or the seventh parameter is omitted), the module entry trap results in a 0C1 program check. If LOOP is specified as the seventh parameter, the module entry trap results in an enabled loop.

omitted Indicated by a comma or by the fourth or earlier parameter being the end of the reply to CVGTF04. The module entry/exit trap option is not enabled.

The sixth positional parameter must be:

EXIT Specifies that the module exit trap option is to be enabled. A module name must be specified as the fourth parameter, or EXIT will be ignored. If jobname is supplied as the third parameter, the module exit trap is enabled only for the specified job. If PGMCHK is specified as the seventh parameter (or the seventh parameter is omitted), the module exit trap results in a 0C1 program check. If LOOP is specified as the seventh parameter, the module exit trap results in an enabled loop.

omitted Indicated by a comma or by the fifth or earlier parameter being the end of the reply to CVGTF04. The module entry/exit trap option is not enabled.

The seventh positional parameter must be:

PGMCHK The module entry/exit trap results in a 0C1 program check when the specified module gains control. ENTRY must be specified as the fifth parameter, and EXIT must be specified as the sixth parameter.

LOOP The module entry/exit trap results in an enabled loop. The loop continues until the job time limit is reached or the job is canceled, or until CVAFGTF is invoked to change or remove the module entry/exit trap.

omitted Indicated by a comma or by the sixth or earlier parameter being the end of the reply to CVGTF04. Operation is the same as if PGMCHK had been supplied.

6. CVAFGTF then issues the following message to inform the operator of the debug trap options that are in effect:

   ```
   CVGTF05 CVAF DEBUG PARAMETERS ARE p1,p2,...,p7
   ```

   Parameters p1 through p7 have the same format as the preceding positional parameters 1 through 7. Parameters are the same as the reply to CVGTF04 if all requirements for invoking the options have been met.
7. CVAFGTF then reissues message CVGTF01 as in step 2 on page 34. If ENDT is supplied in response to CVGTF01, the CVAFGTF program terminates, but leaves any selected options set until a reset parameter is issued.

CVAF Debug Example 1

To invoke the save CVAF module work area option for all jobs, reply to message CVGTF04:

```
R nn,set,save
```

CVAFGTF replies with the following message:

```
CVGTF05 CVAF DEBUG PARAMETERS ARE SET ,SAVE  ,    ,    ,    ,    ,
```

CVAF Debug Example 2

To invoke the save CVAF module work area option for job/TESTJOB, reply to message CVGTF04:

```
R nn,set,save,testjob
```

CVAFGTF replies with the following message:

```
CVGTF05 CVAF DEBUG PARAMETERS ARE SET ,SAVE  ,TESTJOB,   ,   ,   ,
```

CVAF Debug Example 3

To invoke the save CVAF module work-area option only for job MYJOB and to cause CVAF to loop when module ICVCML00 exits for MYJOB, reply to message CVGTF04 as follows:

```
R nn,set,save,myjob,cml00,,exit,loop
```

CVAFGTF replies with the following message:

```
CVGTF05 CVAF DEBUG PARAMETERS ARE SET ,SAVE ,MYJOB  ,CML00,   ,EXIT,LOOP
```

CVAF Debug Example 4

To disable all CVAF debug trap options, reply to message CVGTF04:

```
R nn,reset
```

or

```
R nn,
```

CVAFGTF replies with the following message:

```
CVGTF05 CVAF DEBUG PARAMETERS ARE RESET,   ,   ,   ,   ,   ,
```

CVAF Debug Example 5

On entry to module ICVCML00, if the CVAF module trap for job TRAPJOB is requested but the ENTRY parameter is accidentally omitted, a module trap is not established. The reply to message CVGTF04 follows:

```
R nn,set,,trapjob,cml00
```

CVAFGTF replies with the following message:

```
CVGTF05 CVAF DEBUG PARAMETERS ARE SET ,NOSAVE,   ,   ,   ,   ,
```

This message indicates that debug trap options are not in effect. The reply to CVGTF04 might be reissued after entering DBUG to message CVGTF01 as follows:

```
R nn,set,,trapjob,cml00,entry
```

CVAFGTF replies with the following message:

```
CVGTF05 CVAF DEBUG PARAMETERS ARE SET ,NOSAVE,TRAPJOB ,CML00,ENTRY,  ,PGMCHK
```

Note that PGMCHK has been defaulted.

CVAF—VTOC Index Error Message

When CVAF detects an error in a VTOC index, it issues the following message:

IEC606I VTOC INDEX DISABLED ON dev,volser,code[,rba[,secno,offset]]

In addition, CVAF places the same return code that is in the IEC606I message into the CVSTAT field of the CVPL.

Explanation

The Common VTOC access facility (CVAF) detected a VTOC index error on the volume.

dev Specifies the device number.

volser Specifies the volume serial number of the volume.

code Specifies a number that represents the kind of VTOC index error encountered.

rba Specifies the RBA of the VIR in the VTOC index that contains a structure error indicated by **code**.

secno Specifies the section number in the VIER containing the VTOC index entry, if the VIR is a VIER.

offset Specifies the offset into the section of the VTOC index entry, if the VIR is a VIER.

System Action

The VTOC index is disabled by zeroing the index bit in the format-4 DSCB and setting the bit in the first high-level VIER that indicates an invalid VTOC index structure. A system dump is written to the SYS1.DUMPxx data set, and an entry is made in the SYS1.LOGREC data set. The VTOC is converted to nonindexed format when DADSM next allocates space on the volume. The message IEC604I (which indicates that the VTOC convert routines have been used) will be issued later.

Programmer Response

Examine the system dump and a print of the VTOC index. Use the information in message IEC606I to determine the cause of the VTOC index structure error.

CVAF Dumps

If CVAF detects an error, module ICVCME02 is called to produce an ABEND18B dump for the following conditions. The following error types relate to the error type that is identified on the title line of the dump.

- For CVAF Error Type 1:
  - The indexed VTOC is disabled.
  - Message IEC606I RC*xxx* is issued to report the error.
- For CVAF Error Type 2:
  - The W/A identified on the title line of the dump is the calling program's work area.
  - Examine the work area pointed to by *yyyyyyyy* in the following example to locate the eye-catcher **CVPL**. It might be necessary to scan backwards from the start of the W/A. The CVSTAT field (CVPL + X'07') contains a 1-byte status code describing the error. See "CVSTAT Field Codes" on page 40 for a description of these codes.
- For CVAF Error Type 3:
  - The problem is an Abend*nnn* (except ABEND18B) in either CVAF or in a function called by CVAF.
  - The W/A identified on the title line of the dump is the work area for the module identified in MODNAME on the title line.
  - Starting at the work area pointed to by *yyyyyyyy* in the following example, examine the storage preceding the W/A (scan backwards from the start of the W/A) to locate the eye-catcher **CVPL**. The CVSTAT field (CVPL + X'07') contains a 1-byte status code describing the error. See "CVSTAT Field Codes" on page 40 for a description of these codes.

The dump title line is in the following format. See Figure 1 for a sample title line.

```
MODNAME  JOBNAME  CVAF ERROR TYPE n   TCB=xxxxxxxx  W/A=yyyyyyyy
```

where:

MODNAME is the CVAF module involved:

n is the error type (1, 2, or 3)

xxxxxxxx is the TCB address

yyyyyyyy is the work area address

```
ICVCMB00  CVJOB   CVAF ERROR TYPE 1   TCB=00DC6400  W/A=00DFAB20
```

Figure 1. Sample CVAF Dump Title Line

The CVFCTN field (CVPL +X'06') and the CVSTAT field (CVPL +X'07') in the failure-related CVPL identify the failure-related CVAF function and CVAF status, respectively. For a description of function code values, see the table of CVFCTN values in the DADSM/CVAF modifier keywords section of *z/OS DFSMSdfp Diagnosis Guide*. For a description of the CVSTAT codes, see "CVSTAT Field Codes" on page 40.

CVSTAT Field Codes

Status Code	Description
0 (X'00')	No error occurred.
1 (X'01')	Data set name was not found, or VIER is empty.
2 (X'02')	Argument is outside VTOC extents or RBA range of VTOC index.
4 (X'04')	Invalid parameter was supplied (wrong key), or VRFAREA is too small.
5 (X'05')	DSN keyword was omitted.
6 (X'06')	Not authorized to perform this function.
7 (X'07')	Buffer list was omitted.
8 (X'08')	DEB was not valid, omitted, not open to VTOC, or not authorized to supply specified parameter
9 (X'09')	IOAREA=KEEP and user was not authorized, or I/O area was supplied and either user was not authorized or CVAFVOL function.
10 (X'0A')	Function is not supported on OS VTOC.
11 (X'0B')	DSCB is not format-0 DSCB and VERIFY=YES.
12 (X'0C')	MAPRCDS=YES or IXRCDS=KEEP, or both, but VTOC is nonindexed.
13 (X'0D')	IXRCDS=KEEP not specified for CVAFDSM ACCESS=IXADD or IXDLT.
14 (X'0E')	CTAREA keyword omitted.
15 (X'0F')	UCB is not valid, volume was not mounted; VIO unit, not DASD.
17 (X'11')	DSCB length invalid for the function requested: 96 bytes for CVAFDIR ACCESS=WRITE,VERIFY=YES; 140 bytes for CVAFSEQ reading in data set name sequence; 96 bytes for CVAFSEQ reading in physical sequence.
19 (X'13')	UCB was omitted and CVAF I/O area was not supplied.
22 (X'16')	Data set name was already supplied in index.
23 (X'17')	Invalid DSN was supplied (44 X'FF' is a reserved data set name).
24 (X'18')	ARG keyword was not supplied.
25 (X'19')	Conflicting or incomplete information specified in the space table for a CVAFDSM ACCESS=ALLOC, MAP=VOLUME request, or CVAFVSM ACCESS=ALLOC request.
27 (X'1B')	VTOC index is full. No free VIRs are available and a VIER split is required.
28 (X'1C')	Space keyword was omitted (CVSPACE field zero in CVPL).
29 (X'1D')	CVAFDSM ACCESS=ALLOC: no format-0 DSCB is available (MAP=VTOC), VTOC index is full (MAP=INDEX), or volume space is not available (MAP=VOLUME). CVAFVSM ACCESS=ALLOC: volume space is not available.

30 (X'1E') CVAFDSM ACCESS=ALLOC: CCHHR (MAP=VTOC) or RBA MAP=INDEX or volume space extent (MAP=VOLUME) was already allocated. CVAFVSM ACCESS=ALLOC: volume space extent was already allocated.

31 (X'1F') CVAFVSM ACCESS=ALLOC: Volume space extent invalid or outside VTOC extents (MAP=VTOC), or RBA outside VTOC index extents (MAP=INDEX), volume. CVAFDSM ACCESS=MAPDATA: Input starting RBN is invalid.

32 (X'20') End of data. CVAFDSM ACCESS=MAPDATA: no more free extents in VPSM. CVAFSEQ: no more names in index or DSCBs in VTOC. For indexed access, no DSN in a VTOC index with higher or higher-or-equal key than that supplied. For physical-sequential access, no DSCB in the VTOC has a higher argument than that supplied. For a multiple DSCB request, the last DSCB in the VTOC was read and more DSCBs were requested.

33 (X'21') EXTENTS keyword was omitted, or supplied number of extents is zero.

34 (X'22') CVAFDSM ACCESS=RLSE and format-0 DSCB is already free (MAP=VTOC), or VIER is already unallocated (MAP=INDEX), or volume space extent is already unallocated (MAP=VOLUME). CVAFVSM ACCESS=RLSE: volume space extent is already unallocated.

42 (X'2A') VRF data supplied for write too long.

43 (X'2B') Buffer list is for VIRs, but a DSCB buffer list is required.

44 (X'2C') No buffer list entry found.

45 (X'2D') Invalid DSCB buffer length (neither 96 nor 140) in buffer list entry, or VIR buffer length not equal to VIB VIR size.

46 (X'2E') Neither TTR nor CCHHR bits set in buffer list entry to be used in writing or reading a 140-byte DSCB.

47 (X'2F') More than one of the TTR, CCHHR, and RBA bits set in the buffer list entry.

48 (X'30') Both the DSCB and VIR bits set in the buffer list header.

49 (X'31') RBA bit set in a buffer list entry for a DSCB buffer list.

50 (X'32') TTR or CCHHR bit is set in buffer list entry, but buffer list header indicates that buffer list is for a VIR.

52 (X'34') Combination of MAP and COUNT is not supported.

53 (X'35') MAP was either omitted or multiply specified.

54 (X'36') Buffer list for a VIR chained to or from a buffer list for a DSCB.

55 (X'37') Unauthorized caller, and VIB was not initialized.

56 (X'38') MAPRCDS=YES was not specified but is required.

57 (X'39') Buffer list for a DSCB was supplied, but buffer list for a VIR is required (in MAPRCDS or IXRCDS buffer list address in CVPL).

58 (X'3A') Neither the VIR nor DSCB bit set in a buffer list header.

60 (X'3C') Invalid or conflicting setting of allocate option byte is in space parameter

61 (X'3D') Filter criteria list address was omitted (CVFCL=0) or ID was not FCL (CVAFFILT).

62 (X'3E') FCLCOUNT field is zero or no elements to process (CVAFFILT).

63 (X'3F') FCLCOUNT field is greater than 1 and FCL1LIST indicates a generic data set name (CVAFFILT).

64 (X'40') Insufficient number of user buffer list entries to complete this request. All data set names in the filter criteria list up to this point processed successfully. Use CVAFFILT ACCESS=RESUME function to continue processing.

65 (X'41') Buffer list entry error occurred; for example, buffer length not 140 bytes (CVAFFILT).

66 (X'42') CVAFFILT ACCESS=RESUME and filter save address (FSA) omitted (CVFSA=0), or FSA not in CVAF protect key.

67 (X'43') Invalid partially qualified data set name for generic access (CVAFFILT).

68 (X'44') Filter criteria list not in user key (CVAFFILT).

69 (X'45') Internal GETMAIN error (CVAFFILT).

70 (X'46') Direct VTOC access I/O error (CVAFFILT).

71 (X'47') Sequential VTOC access I/O error (CVAFFILT).

72 (X'48') Error occurred in CVAFFILT processing. CVAFFILT ACCESS=RESUME function is not recommended.

73 (X'49') Insufficient number of user buffer list entries to complete this request. Error in DSCB chain. One or more names in the filter criteria list not processed successfully; however, CVAFFILT ACCESS=RESUME function recommended for remaining DSCBs.

74 (X'4A') Data set name information is invalid (FCLDSNLG is either 0 or greater than 44, or FCLDSNA is either 0 or not in user key) (CVAFFILT).

75 (X'4B') Extent is beyond 65536 track boundary. For CVAFDSM ACCESS=MAPDATA MAP=VOLUME: free extents up to 65536 tracks boundary, if any, are returned in extent table of XXYYZ format. User should specify RTA4BYTE=YES and use extent table mapped by ICVEDT02.

76 (X'4C') VTOC status from format-4 DSCB does not match status indicated by CVPL flag CV1IVT.

77 (X'4D') ID of free-space DSCB is incorrect.

78 (X'4E') ID of extent table is incorrect. ICVEDT02 is expected.

79 (X'4F') ID of space table is incorrect. ICVVLSDT is expected.

80 (X'50') RTA4BYTE=YES was not specified for CVAFDSM ACCESS=MAPDATE MAP=VOLUME for an OSVTOC volume.

127 (X'7F') I/O error occurred.

128 (X'80') Reserved.

129 (X'81') The first high-level VIER as indicated in the VIXM does not have the flag bit set indicating that it is the first high-level VIER.

130 (X'82') A horizontal or vertical VIER pointer is outside the RBA range of the VTOC index.

131 (X'83') A vertical VIER pointer points to a VIR that is not a VIER (invalid ID in header).

132 (X'84') A level *n* vertical index entry pointer points to a VIER that is not at level *n* - 1.

133 (X'85') Level *n* horizontal index entry pointer points to VIER that is not at level *n*.

134 (X'86') Horizontal VIER/map pointer points to a VIR that is not a VIER/map (invalid ID in header).

135 (X'87') Horizontal map pointer points to a VIR that is not one of the first *n* VTOC index records (*n* is recorded in VIXM field VIMRCDS), or the first record in the VTOC index is not a VIXM.

136 (X'88') A level-1 index entry contains a CCHHR pointer that is outside the VTOC extent.

137 (X'89') The first high-level VIER, as indicated in the VIB, does not have the flag bit set indicating that it is the first high-level VIER. (Either this error is either recovered by updating the VIB from the VIXM, or the error is changed to 129.)

138 (X'8A') The RBA of the VTOC index VIR does not match the RBA recorded in the header of the record.

139 (X'8B') The first record of a map (VIXM, VPSM, or VMDS) is not one of the first n VTOC index records (*n* is recorded in the VIXM field, VIMRCDS).

140 (X'8C') The data set name in a level *n* + 1 VIER entry is lower than the high key of the level *n* VIER that the level *n* + 1 VIER entry points to.

141 (X'8D') First high-level VIER structure error bit is on.

142 (X'8E') I/O error indicating the VTOC index is not formatted correctly.

143 (X'8F') Either the index bit is zero, or the VSE bit is zero in the format-4 DSCB of a VTOC previously found to be an indexed VTOC.

144 (X'90') No SYS1.VTOCIX.*nnn* data set name in a VTOC whose format-4 DSCB has the index bit on, indicating the VTOC has an index.

145 (X'91') The data set name in a level *n* + 1 VIER entry is higher than the high key of the level *n* VIER to which the level *n* + 1 VIER entry points.

146 (X'92') Four or more high-level VIERs were found.

147 (X'93') Too many levels in the VTOC index. The length of the search list was exceeded.

148 (X'94') VIER is not valid because offset to last section is not valid.

149 (X'95') VIER is not valid because offset to last entry in a section is not valid.

150 (X'96') Media manager initialization failed.

151 (X'97') Level-2 or higher VIER contains fewer than two entries.

152 (X'98') RECOVER=YES was specified, but the static text module (ICVIXST0) indicates that recovery is not permitted.

153 (X'99') The format-4 DSCB on an indexed VTOC is written with either the index bit or the VSE bit zeroed on an indexed VTOC.

154 (X'9A') A space map extends over more than 20 VTOC index records.

155 (X'9B') Data set name was not found in section with key greater than or equal to the name being searched for. The VIER section containing the name is invalid.

156 (X'9C') Invalid VIER horizontal pointer. Horizontal pointer of VIER1 points to VIER2, whose high key is lower than or equal to the high key of VIER1.

157 (X'9D') Could not find entry in level-2 or higher VIER that matches the high key of the VIER.

158 (X'9E') Invalid section length or invalid number of sections in a VIER header.

159 (X'9F') The first high-level VIER pointed to by the VIB has an invalid ID in the header.

160 (X'A0') Request to write DSCB data (96 bytes) to the location of the format-4 DSCB. Data in buffer is not a valid format-4 DSCB (bytes 0-43 not X'04', or byte 44 not X'F4').

161 (X'A1') Request to write a format-4 DSCB by name (44 X'04'). Data in buffer is not a valid format-4 DSCB: ID byte (byte 44) is not X'F4'.

162 (X'A2') Request to write 140 bytes (key and data) to the location of the format-4 DSCB. Data in buffer is not a valid format-4 DSCB (bytes 0-43 not X'04', or byte 44 not X'F4'.

163 (X'A3') Request to read DSCB data (96 bytes) from the location of the format-4 DSCB. Data read into buffer is not a valid format-4 DSCB (byte 0 is not X'F4').

164 (X'A4') Request to access format-4 DSCB by name (44 X'04'), read data (96 bytes) and location (CCHHR). Data read into buffer is not a valid format-4 DSCB (byte 0 is not X'F4').

165 (X'A5') Request to write a format-4 DSCB by name (44 X'04') and return the location (CCHHR). ID byte is not valid for a format-4 DSCB (byte 44 is not X'F4').

166 (X'A6') Request to read DSCB key and data (140 bytes) from the location of the format-4 DSCB. Data read into buffer is not a valid format-4 DSCB (byte 44 is not X'F4').

Format-7 DSCB

Common Name: Format 7– –VTOC Data Set Control Block

Owning Component: DADSM (DF107)

Eye-catcher ID: X'07070707'

Subpool and Key: Not applicable

Size: 140 bytes.

Pointed to by: The CCHHR address of the first format-7 DSCB is contained in the DS4EFPTR field of the format-4 DSCB.

Function: Extent description for certain devices

Offset	Type/Value	Len	Name (Dim)	Description
0 (0)	CHARACTER	4	DS7KEYID	KEY IDENTIFIER (X'07070707')
	SPACE FOR 5 F7 ENTRIES			
4 (4)	CHARACTER	40	DS7EXTNT	SPACE FOR 5 EXTENT DESCRIPTIONS
44 (2C)	CHARACTER	1	DS7FMTID	FORMAT IDENTIFIER (X'F7')
	SPACE FOR 11 F7 ENTRIES			
45 (2D)	CHARACTER	88	DS7ADEXT	SPACE FOR 11 EXTENT DESCRIPTIONS
133 (85)	CHARACTER	2		RESERVED
	DESCRIPTION OF THE F7 FORWARD POINTER			
135 (87)	CHARACTER	5	DS7PTRDS	POINTER (CCHHR) TO NEXT FORMAT 7 DSCB OR ZERO
135 (87)		*	DS7CCPTR	TWO BYTES OF CYLINDER DATA (CC)
		*	DS7HHPTR	TWO BYTES OF HEAD DATA (HH)
		*	DS7RPTR	ONE BYTE OF RECORD DATA (R)
		*	DS7END	END OF THE DSCB 7 MAPPING
	SINGLE FORMAT 7 RTA EXTENT DESCRIPTION EACH FORMAT 7 ENTRY HAS THE FOLLOWING FORMAT: BYTES 0-3 START RTA ADDRESS BYTES 4-7 END RTA ADDRESS + 1			
	CHARACTER	8	DS7EXT	DESCRIPTION OF ONE F7 EXTENT DESCRIPTION ENTRY
	CHARACTER	4	DS7RTAST	STARTING RTA VALUE
	CHARACTER	4	DS7RTAED	ENDING RTA VALUE + 1

DADSM Module-to-Function Cross-Reference

The following tables provide the function and subfunction keywords for the related DADSM and CVAF functions. In these tables, the characters xx are used to represent any two characters.

For a description of how to use the DADSM function and subfunction keywords as modifiers, see the section describing DADSM/CVAF modifier keywords in *z/OS DFSMSdfp Diagnosis Guide*. Table 23 on page 46 lists the function and subfunction keywords for DADSM and CVAF functions.

DADSM/CVAF Diagnostic Aids

Table 23. Function and Subfunction Keywords for DADSM and CVAF Functions

Module	Function
IGCxx02G	OBTAIN
IGC0002I	SCRATCH
IGC0003B	CREATE
IGC00030	RENAME
IGCxx07H	LSPACE
IGCxx09H	PROTECT
IGGDACxx	CREATE
IGGDAExx	EXTEND
IGGDAJxx	COMMON[1]
IGGDAPxx	Partial Release
IGGDARxx	RENAME
IGGDASxx	SCRATCH
IGGDATBL	COMMON[1]
IGGDAUxx	UPDATE
IGGPEXIT	COMMON[1]
IGGPOST0	Installation Exit
IGGPRE00	Installation Exit
IGGVRFxx	VRF
IGG0CLF2	CREATE
IGG019EK	COMMON[1]
IGG020xx	PRLSE
IGG029xx	SCRATCH
IGG029CM	COMMON[1]
IGG0290F	COMMON[1]
IGG030xx	RENAME
IGG032DB	VRF
IGG032xx	CREATE
IGG055xx	EXTEND

Note:

1. This module is shared by more than one function.

DADSM Return and Diagnostic Codes

This section describes the following DADSM return and diagnostic codes:

- DADSM CREATE (Allocate) return and diagnostic codes
- DADSM Extend function return codes
- DADSM ISJF function return and diagnostic codes
- DADSM LSPACE return and diagnostic codes
- DADSM OBTAIN function return codes
- DADSM Partial Release return and diagnostic codes
- DADSM Protect function return codes
- DADSM RENAME return and diagnostic codes
- DADSM SCRATCH return and diagnostic codes

DADSM CREATE (Allocate) Return and Diagnostic Codes

This section defines the return codes and diagnostic codes returned by the CREATE function of DADSM. These return and diagnostic codes are displayed in messages. The REALLOC macro and SVC 32 interfaces to DADSM CREATE return these return codes and diagnostic codes in registers 15 and 0, respectively. Register 15 contains the applicable DADSM SVC return code, as Table 24 on page 48 shows. Register 0 contains 4 bytes of diagnostic information that consists of the DADSM error code, subfunction identifier, subfunction return code, and subfunction reason code. as Table 25 on page 50 shows.

Tip: For certain errors during DADSM Extend processing for secondary allocations on a new volume, the diagnostic information in Table 25 on page 50 describes the DADSM Extend failure.

If an error occurs, DADSM might issue message **IEC614I**, which would contain the return codes and reason codes.

Table 24 on page 48 describes the conditions that are indicated by the DADSM SVC return code.

Note: If SVC 99 returns a 47xx code in the field S99ERROR, the xx represents the DADSM return code, which are in the DADSM return codes table in Table 24 on page 48.

The field S99INFO contains the right-most 2 bytes of the DADSM diagnostic code, which are in the DADSM diagnostic information table in Table 25 on page 50.

Because only the right-most 2 bytes are returned, the columns for bytes 2 and 3 should be examined for a match with the returned value. If the user requested the “return dynamic allocation message” by the use of the fields S99EIMSG and S99EWPT of the SVC99 request block extension, then the returned message should contain the complete 4-byte DADSM diagnostic code.

Note: This is a complete list of DADSM creation return codes. Some of these codes may not apply to the REALLOC macro.

Table 24. DADSM CREATE Return Codes

DADSM SVC Return Code (R15)	Description
00 (X'00')	If the 4 bytes of diagnostic information returned in register 0 are all zeros, this indicates successful data set creation. If they are nonzero, examine Table 25 on page 50 for possible error conditions.
04 (X'04')	Duplicate data set name.
08 (X'08')	No room in VTOC or VTOC index.
12 (X'0C')	Permanent I/O error or CVAF error. The function code is invalid or not supported at this level.
16 (X'10')	Requested absolute track not available.
20 (X'14')	Requested quantity not available.
24 (X'18')	Average record length exceeds 65535 bytes.
28 (X'1C')	Invalid DSORG for ISAM index request.
32 (X'20')	No prime area requested for ISAM.
36 (X'24')	ISAM overflow area requested before prime.
40 (X'28')	ISAM space request does not start on cylinder boundary.
44 (X'2C')	Duplicate ISAM data set element.
48 (X'30')	Invalid DADSM REALLOC parameter list.
52 (X'34')	Invalid JFCB or partial DSCB pointer.
56 (X'38')	Not enough space on volume for directory.
60 (X'3C')	REALLOC ALLOC=ABS is not supported on OS VTOCs.
64 (X'40')	Invalid user label request.
68 (X'44')	Invalid UCB pointer.
72 (X'48')	VSE VTOC cannot be converted to an OS VTOC.
76 (X'4C')	No space parameter given for a new data set or zero space requested at absolute track zero.
80 (X'50')	Invalid request for ISAM index.
84 (X'54')	ISAM multivolume index is not allowed.
88 (X'58')	Invalid ISAM data set element.
92 (X'5C')	ISAM multivolume overflow is not allowed.
96 (X'60')	ABSTR and CYL requests conflict (ISAM).
100 (X'64')	CYL and CONTIG request conflict (ISAM).
104 (X'68')	Invalid space subparameter.
108 (X'6C')	Zero primary space request for ISAM data set or ABSTR request.
112 (X'70')	ISAM index request duplication.
116 (X'74')	User labels not supported.
120 (X'78')	Invalid combination of DSSIZE and MINAU in REALLOC parameter.
124 (X'7C')	DSSIZE not a multiple of MINAU.
128 (X'80')	Directory space requested is larger than primary space.
132 (X'84')	Space request must be ABSTR for a VSE volume.
136 (X'88')	Invalid FMT3 DSCB pointer.
140 (X'8C')	Index must be requested before prime area.

Table 24. DADSM CREATE Return Codes (continued)

DADSM SVC Return Code (R15)	Description
144 (X'90')	Invalid third DD statement for ISAM creation.
148 (X'94')	Overlapping extents in the VTOC.
152 (X'98')	Overlapping VSE split cylinder extents in the VTOC.
156 (X'9C')	DADSM CREATE terminated because of possible VTOC errors.
160 (X'A0')	ISAM CREATE terminated because of possible VTOC errors.
164 (X'A4')	Creation terminated because of VSE stacked pack format.
168 (X'A8')	RACDEF failed, data set already defined.
172 (X'AC')	User not authorized to define data set.
176 (X'B0')	Installation exit rejected this request with return code 8.
180 (X'B4')	Installation exit rejected the request with return code 4.
184 (X'B8')	RACF, a component of the Security Server for z/OS, redefine with modeling specified and model not found.
188 (X'BC')	Invalid FMT2 DSCB pointer.
192 (X'C0')	Requested data set creation was not allowed by SMS.
196 (X'C4')	Requested data set creation was not possible. Register 0 contains additional diagnostic information. See Table 25 on page 50 for an explanation of the diagnostic information.
200 (X'C8')	PDSE directory creation failed.
204 (X'CC')	VTOC ENQ related failure.
208 (X'D0')	I/O error occurred during the allocation of a data set. The data set will be deleted.
212 (X'D4')	Request failed due to a presence of split-cylinder data set(s) on the volume.
216 (X'D8')	For this type of data set, the primary quantity requested must not exceed 65535 tracks.

Table 25 on page 50 describes the conditions that are indicated by the diagnostic information that is returned in register 0. This is a complete list of information. Some of it may not apply to the REALLOC macro. The abbreviations that are used in Table 25 on page 50 are defined in Table 26 on page 64.

DADSM/CVAF Diagnostic Aids

Table 25. DADSM CREATE Diagnostic Information (Register 0)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'00'	X'00'	X'00'	X'80'	obsolete.
X'04'	X'03'	VDSS1	VDSS2	VDSS detected an error.
X'04'	X'04'	X'00'	X'09'	Build JFCB interface; both JFCBDRLH and JFCBLKSI are zero for a data set allocated by average block.
X'04'	X'04'	X'00'	X'0A'	Build JFCB interface; DS1SCXTV value exists, but no DS1SCXTF flag is set.
X'04'	X'09'	X'00'	X'33'	Validate JFCB extension; CVOL data sets not allowed on SMS-managed volumes (high-level qualifier of SYSCTLG).
X'04'	X'09'	X'00'	X'34'	Validate JFCB extension; unmovable data sets not supported on SMS-managed volumes.
X'04'	X'09'	X'00'	X'7E'	Validate JFCB extension; ISAM data sets not supported on SMS-managed volumes.
X'04'	X'09'	X'00'	X'7F'	Validate JFCB extension; absolute track data sets not supported on SMS-managed volumes.
X'04'	X'09'	X'00'	X'83'	Validate JFCB extension; minimum extent size invalid with MXIG or ALX.
X'04'	X'09'	X'00'	X'85'	Validate JFCB extension; AVGREC invalid when space requested in cylinders or tracks.
X'04'	X'0A'	X'00'	X'33'	Validate model DSCB extension; CVOLs (high-level qualifier of SYSCTLG) not supported on SMS-managed volumes.
X'04'	X'0A'	X'00'	X'34'	Validate model DSCB extension; unmovable data sets not on SMS-managed volumes.
X'04'	X'0A'	X'00'	X'7E'	Validate Model DSCB extension; ISAM data sets not supported on SMS-managed volumes.
X'04'	X'0A'	X'00'	X'7F'	Validate model DSCB extension; absolute track data sets not supported on SMS-managed volumes.
X'04'	X'0A'	X'00'	X'80'	Validate model DSCB extension; average block length not available for average block length data set.
X'04'	X'0A'	X'00'	X'81'	Validate model DSCB extension; partial DSCB indicates that data set is allocated by absolute track.
X'04'	X'0A'	X'00'	X'82'	Validate model DSCB extension; DACLRGEX request invalid when MXIG or ALX type of space request.
X'04'	X'0A'	X'00'	X'83'	Validate model DSCB extension; DACMINES request invalid when MXIG or ALX type of space request.
X'04'	X'0A'	X'00'	X'94'	Validate model DSCB extension; user label not allowed for PDSE.

Table 25. DADSM CREATE Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'04'	X'0D'	RACRET	RACREAS	Invoke RACF to check caller's authority. RACROUTE Request = AUTH.
X'04'	X'0E'	X'00'	X'52'	Invoke RACF to create a profile for the data set; data set protected by discrete profile, but RACF not active.
X'04'	X'0E'	RACRET	RACREAS	Invoke RACF to create a profile for the data set. RACROUTE Request = DEFINE.
X'04'	X'0F'	X'00'	X'52'	Invoke RACF to create a profile for the data set using a security model; data set protected by discrete profile, but RACF not active.
X'04'	X'0F'	RACRET	RACREAS	Invoke RACF to create a profile for the data set using a security model. RACROUTE Request=DEFINE.
X'04'	X'10'	X'00'	X'52'	Invoke RACF to create a profile for the data set using a generic security model; data set protected by discrete profile, but RACF not active.
X'04'	X'10'	RACRET	RACREAS	Invoke RACF to create a profile for the data set using a generic security model. RACROUTE Request = DEFINE.
X'04'	X'11'	X'00'	X'52'	Invoke RACF to add a volume serial number to a data set's profile; data set protected by discrete profile, but RACF not active.
X'04'	X'11'	RACRET	RACREAS	Invoke RACF to add a volume serial number to a data set's profile. RACROUTE Request = DEFINE.
X'04'	X'12'	X'00'	X'35'	Primary or secondary quantity is too large to allocate data set.
X'04'	X'14'	X'00'	X'36'	Validate DADSM CREATE request; ADSCB request—invalid FMT2 key.
X'04'	X'14'	X'00'	X'37'	Validate DADSM CREATE request; ADSCB request—invalid FMT2 ID.
X'04'	X'14'	X'00'	X'38'	Validate DADSM CREATE request; ADSCB request—FMT1 DSCB DSORG must indicate ISAM data set.
X'04'	X'14'	X'00'	X'39'	Validate CREATE request; data set name is VTOC name.
X'04'	X'14'	X'00'	X'3A'	Validate CREATE request; MDSCB indicates ISAM data set ISAM data set invalid for model DSCB request.
X'04'	X'14'	X'00'	X'3B'	Validate CREATE request; ADSCB request—space requested at absolute track 0.
X'04'	X'14'	X'00'	X'3C'	Validate DADSM CREATE request; MDSCB request—no directory quantity specified for PDS data set when primary quantity is not zero.

Table 25. DADSM CREATE Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'04'	X'14'	X'00'	X'3D'	Validate DADSM CREATE request; MDSCB request—directory quantity specified, but data set is not a PDS.
X'04'	X'14'	X'00'	X'3E'	Validate DADSM CREATE request; ADSCB request—directory quantity specified, but data set is not a PDS.
X'04'	X'14'	X'00'	X'3F'	Validate DADSM CREATE request; ADSCB request—no directory quantity specified for PDS data set when primary quantity is not zero.
X'04'	X'14'	X'00'	X'40'	Validate CREATE request; MDSCB or ADSCB address invalid.
X'04'	X'14'	X'00'	X'41'	Validate DADSM CREATE request; ADSCB request—zero space requested at absolute track 0.
X'04'	X'14'	X'00'	X'42'	Validate DADSM CREATE request; JFCB request—DSORG cannot be ISAM or PDS.
X'04'	X'14'	X'00'	X'43'	Validate DADSM CREATE request; JFCB request—DSORG cannot be graphics.
X'04'	X'14'	X'00'	X'44'	Validate DADSM CREATE request; JFCB request—directory quantity cannot be specified.
X'04'	X'14'	X'00'	X'45'	Validate DADSM CREATE request; ADSCB request—DSORG cannot be ISAM or PDS.
X'04'	X'14'	X'00'	X'46'	Validate DADSM CREATE request; ADSCB request—FMT3 DSCB present and number of extents is greater than 15.
X'04'	X'14'	X'00'	X'47'	Validate DADSM CREATE request; ADSCB request—FMT3 DSCB present and number of extents is less than 3.
X'04'	X'14'	X'00'	X'48'	Validate DADSM CREATE request; ADSCB request—no FMT3 DSCB present and number of extents is greater than 2.
X'04'	X'14'	X'00'	X'49'	Validate DADSM CREATE request; ADSCB request—primary quantity is zero.
X'04'	X'14'	X'00'	X'4A'	Validate DADSM CREATE request; primary quantity is zero, but minimum allocation unit is not zero.
X'04'	X'14'	X'00'	X'4B'	Validate DADSM CREATE request; primary quantity is less than 0.
X'04'	X'14'	X'00'	X'4C'	Validate DADSM CREATE request; minimum allocation unit is less than 0.
X'04'	X'14'	X'00'	X'4D'	Validate DADSM CREATE request; ADSCB request—invalid FMT3 key.
X'04'	X'14'	X'00'	X'4E'	Validate DADSM CREATE request; ADSCB request—invalid FMT3 ID.
X'04'	X'14'	X'00'	X'4F'	Validate DADSM CREATE request; ADSCB request—number of extents is greater than 16.

Table 25. DADSM CREATE Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'04'	X'14'	X'00'	X'50'	Validate DADSM CREATE request; ADSCB request—number of extents is less than 4.
X'04'	X'14'	X'00'	X'51'	Validate DADSM CREATE request; ADSCB request—no FMT3 DSCB present and number of extents is greater than 3.
X'04'	X'14'	X'00'	X'88'	Primary quantity is not a multiple of minimum extent quantity (DACMINES or DACJMNES).
X'04'	X'15'	PREREA1	PREREA2	DADSM CREATE IGGPRE00 preprocessing exit rejected the request.
X'04'	X'16'	X'00'	X'53'	Verify that SMS flags passed by the caller match those in the FMT4 DSCB; volume in conversion to SMS.
X'04'	X'16'	X'00'	X'54'	Verify that SMS flags passed by the caller match those in the FMT4 DSCB; FMT4 DSCB indicates volume is not SMS-managed, but data set is SMS-managed.
X'04'	X'16'	X'00'	X'55'	Verify that SMS flags passed by the caller match those in the FMT4 DSCB; FMT4 DSCB indicates volume is SMS-managed, but data set is not SMS-managed.
X'04'	X'19'	CVRET	CVSTAT	Invoke CVAFDSM for an absolute DSCB request.
X'04'	X'1A'	CVRET	CVSTAT	Invoke CVAFDSM to obtain space for data set.
X'04'	X'1B'	CVRET	CVSTAT	Invoke CVAFDSM; get FMT0 DSCB.
X'04'	X'1C'	CVRET	CVSTAT	Invoke CVAFDSM; add data set name to index and check for duplicate.
X'04'	X'21'	X'00'	X'59'	Read FMT4 DSCB and check for duplicate data set name; data set name is indexed VTOC high-water-mark key (44 bytes of X'FF').
X'04'	X'21'	X'00'	X'5A'	Read FMT4 DSCB and check for duplicate data set name; data set name is VTOC index name.
X'04'	X'21'	X'00'	X'5B'	Read FMT4 DSCB and check for duplicate data set name; MDSCB/ADSCB request—duplicate data set name.
X'04'	X'21'	X'00'	X'5C'	Read FMT4 DSCB and check for duplicate data set name; JFCB request—duplicate data set name and not ISAM data set.
X'04'	X'22'	X'00'	X'5D'	Verify enough space in VTOC or VTOC index; OS VTOC—no available FMT0 DSCBs in VTOC.
X'04'	X'22'	X'00'	X'61'	Verify enough space in VTOC or VTOC index; ADSCB request—number of tracks needed for directory is greater than the maximum (X'65535').
X'04'	X'22'	X'00'	X'66'	Verify enough space in VTOC or VTOC index; OS VTOC—less than 2 available FMT0 DSCBs in VTOC.

Table 25. DADSM CREATE Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'04'	X'22'	X'00'	X'69'	Verify enough space in VTOC or VTOC index; OS VTOC—number of available FMT0 DSCBs in VTOC is less than 3.
X'04'	X'22'	X'00'	X'6A'	Verify enough space in VTOC or VTOC index; OS VTOC—number of DSCBs needed to create FMT1, FMT3, and FMT5 DSCBs is greater than the number of available FMT0 DSCBs in VTOC.
X'04'	X'22'	X'00'	X'73'	Verify enough space in VTOC or VTOC index; less than 4 FMT0 DSCBs available in VTOC.
X'04'	X'23'	X'00'	X'4F'	Convert extent from cyl/trk format (CCHH) to relative track address (RTA); ADSCB request—number of extents is greater than 16.
X'04'	X'23'	X'00'	X'5E'	Convert extent from cyl/trk format (CCHH) to relative track address (RTA); ADSCB request—beginning CCHH of extent is less than the ending CCHH.
X'04'	X'23'	X'00'	X'5F'	Convert extent from cyl/trk format (CCHH) to relative track address (RTA); ADSCB request—cylinder value of CCHH is greater than the number of cylinders on volume.
X'04'	X'23'	X'00'	X'60'	Convert extent from cyl/trk format (CCHH) to relative track address (RTA); ADSCB request—track value of CCHH is greater than the number of tracks on the volume.
X'04'	X'23'	X'00'	X'62'	Convert extent from cyl/trk format (CCHH) to relative track address (RTA); ADSCB request—user label extent is greater than 1 track.
X'04'	X'23'	X'00'	X'63'	Convert extent from cyl/trk format (CCHH) to relative track address (RTA); ADSCB request—user label extent is not the first extent.
X'04'	X'23'	X'00'	X'64'	Convert extent from cyl/trk format (CCHH) to relative track address (RTA); ADSCB request—type of extent is split cylinder.
X'04'	X'23'	X'00'	X'65'	Convert extent from cyl/trk format (CCHH) to relative track address (RTA); ADSCB request—extent sequence number is invalid.
X'04'	X'25'	X'00'	X'57'	Verify space requested is available; requested absolute track address is greater than the last free track.
X'04'	X'25'	X'00'	X'67'	Verify space requested is available; part of requested absolute extent is not free space.
X'04'	X'25'	X'00'	X'68'	Verify space requested is available; requested absolute track address is less than the first free track.
X'04'	X'25'	X'00'	X'6B'	Verify space requested is available; not enough space in first extent for directory.

Table 25. DADSM CREATE Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'04'	X'25'	X'00'	X'95'	Verify space requested is available; device track capacity is too small for PDSE.
X'04'	X'25'	X'00'	X'96'	Verify space requested is available; directory space is larger than primary space.
X'04'	X'25'	X'00'	X'97'	Requested quantity exceeds 65535 tracks.
X'04'	X'25'	X'00'	X'98'	Verify space requested is available; no extents on this volume qualify for ALX request.
X'04'	X'25'	X'00'	X'99'	Verify space requested is available; no extent on this volume can hold CONTIG request.
X'04'	X'25'	X'00'	X'9A'	Verify space requested is available; insufficient free space on volume.
X'04'	X'25'	X'00'	X'9B'	Verify space requested is available; volume is too fragmented for request.
X'04'	X'25'	X'00'	X'9F'	Requested quantity not available.
X'04'	X'25'	X'00'	X'A0'	Requested quantity not available.
X'04'	X'25'	X'00'	X'A1'	Requested quantity not available.
X'04'	X'25'	X'00'	X'A2'	Requested absolute track not available.
X'04'	X'27'	X'00'	X'5D'	Write FMT3 DSCB; OS VTOC—no available FMT0 DSCBs in VTOC.
X'04'	X'2C'	X'00'	X'00'	Create VIO data set.
X'04'	X'2C'	X'00'	X'3B'	Create VIO data set; ADSCB request—space requested at absolute track 0.
X'04'	X'2C'	X'00'	X'41'	Create VIO data set; ADSCB request—zero space requested at absolute track 0.
X'04'	X'2C'	X'00'	X'49'	Create VIO data set; ADSCB request—primary quantity is zero.
X'04'	X'2C'	X'00'	X'6C'	Create VIO data set; total space needed for data set is equal to the available space but extra track needed for user label.
X'04'	X'2D'	X'00'	X'6D'	Validate ISAM request; imbedded directory space invalid on DD statement for INDEX element.
X'04'	X'2D'	X'00'	X'6E'	Validate ISAM request; either first or only DD statement and ISAM element name not blank, PRIME, INDEX, or OVFLOW.
X'04'	X'2D'	X'00'	X'6F'	Validate ISAM request; more than one DD statement and ISAM element name not blank, PRIME, INDEX, or OVFLOW.
X'04'	X'2D'	X'00'	X'70'	Validate ISAM request; type of space must be CYL or ABSTR, not OVFLOW.
X'04'	X'2D'	X'00'	X'71'	Validate ISAM request; for imbedded INDEX, only one volume can be specified.

Table 25. DADSM CREATE Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'04'	X'2D'	X'00'	X'72'	Validate ISAM request; if 3 DD statements, element names must be specified in the order of INDEX, PRIME, OVFLOW.
X'04'	X'2D'	X'00'	X'74'	Validate ISAM request; processing first DD statement—duplicate data set name.
X'04'	X'2D'	X'00'	X'75'	Validate ISAM request; second or third DD statement—duplicate data set name found on volume other than one of the volumes allocated to first DD statement.
X'04'	X'33'	X'00'	X'59'	Validate ISAM multivolume request; data set name is indexed VTOC high-water-mark key (44 bytes of X'FF').
X'04'	X'33'	X'00'	X'74'	Validate ISAM multivolume request; processing first DD statement—duplicate data set name.
X'04'	X'33'	X'00'	X'75'	Validate ISAM multivolume request; second or third DD statement—duplicate data set name found on volume other than one of the volumes allocated to first DD statement.
X'04'	X'33'	X'00'	X'76'	Validate ISAM multivolume request; space allocated on multiple volumes is less than the requested space.
X'04'	X'38'	X'00'	X'77'	Find space for user label; CYL request—no FMT5 extent found to satisfy user label space request.
X'04'	X'38'	X'00'	X'78'	Find space for user label; TRK request—no FMT5 extent found to satisfy user label space request.
X'04'	X'39'	X'00'	X'00'	VTOC conversion; build FMT5 DSCBs.
X'04'	X'3D'	X'00'	X'00'	VTOC conversion; IGG0325P requested FMT5 space update.
X'04'	X'46'	CDMRSN1	CDMRSN2	Invoke CDM to create an PDSE directory; directory could not be created. For explanation see “CDM Reason Codes” on page 408.
X'04'	X'47'	X'00'	X'8B'	VTOC ENQ not held by caller.
X'04'	X'47'	X'00'	X'8D'	VTOC ENQ not held exclusively by caller.
X'04'	X'47'	ENQRC	X'8C'	Verification of ENQ held by caller failed. RESERVE RET=TEST was used.
X'04'	X'4D'	CDMRET	CDMRSN	Invoke CDM to create a PDSE directory or an HFS directory; directory could not be created. For explanation see “CDM Reason Codes” on page 408.
X'04'	X'4E'	X'00'	X'97'	Requested space exceeds 65535 tracks for a data set type that is limited to 65535 tracks per volume. Striped (extended format), PDSE, VSAM, and HFS data sets can exceed 65535 tracks.
X'04'	X'4F'	COMMRC	COMMRS	System error. A DADSM Common service detected an error during free-space search.

Table 25. DADSM CREATE Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'04'	X'50'	COMMRC	COMMRS	System error. A DADSM Common service detected an error during free-space search.
X'04'	X'54'	COMMRC	COMMRS	System error. A DADSM Common service detected an error during free-space search.
X'04'	X'55'	COMMRC	COMMRS	System error. A DADSM Common service detected an error during free-space DSCB update.
X'04'	X'56'	COMMRC	COMMRS	System error. A DADSM Common service detected an error during free-space DSCB update.
X'04'	X'57'	COMMRC	COMMRS	System error. A DADSM Common service detected an error during absolute DSCB conversion.
X'04'	X'58'	COMMRC	COMMRS	System error. A DADSM Common service detected an error during Format 1 or Format 3 DSCB build.
X'04'	X'59'	COMMRC	COMMRS	System error. A DADSM Common service detected an error during Format 1 or Format 3 DSCB build.
X'04'	X'5A'	COMMRC	COMMRS	System error. A DADSM Common service detected an error during free-space search.
X'04'	X'5B'	X'00'	X'9F'	Requested quantity not available.
X'04'	X'5B'	X'00'	X'A0'	Requested quantity not available.
X'04'	X'5B'	X'00'	X'A1'	Requested quantity not available.
X'04'	X'5C'	X'00'	X'9F'	Requested quantity not available.
X'04'	X'5C'	X'00'	X'A2'	Requested absolute track not available.
X'04'	X'5D'	COMMRC	COMMRS	System error. A DADSM Common service detected an error during format 1 or format 3 DSCB build.
X'04'	X'5E'	COMMRC	COMMRS	System error. A DADSM Common service detected an error during format 1 or format 3 DSCB build.
X'04'	X'60'	X'00'	X'35'	Setup_jfcb_info, convert primary KB in DACRE to tracks. The calculated number of tracks exceeds one word.
X'04'	X'61'	X'00'	X'35'	Setup_jfcb_info, convert secondary KB in DACRE to tracks. The calculated number of tracks exceeds one word.
X'04'	X'62'	X'00'	X'35'	Setup_jfcb_info, convert primary KB in DACRE to blocks for average block. The calculated number of blocks will exceed one word.
X'04'	X'63'	X'00'	X'35'	Setup_jfcb_info, convert secondary KB in DACRE to blocks for average block. The calculated number of blocks will exceed one word.
X'04'	X'64'	X'00'	X'35'	Setup_jfcb_info, convert primary KB in DACRE to blocks for average record. The calculated number of blocks will exceed one word.
X'04'	X'65'	X'00'	X'35'	Setup_jfcb_info, convert secondary KB in DACRE to blocks for average record. The calculated number of blocks will exceed one word.

Table 25. DADSM CREATE Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'04'	X'66'	X'00'	X'35'	Setup_jfcb_info, convert primary in JFCB to blocks for average record. The calculated number of blocks will exceed one word.
X'04'	X'67'	X'00'	X'35'	Setup_jfcb_info, convert secondary in JFCB to blocks for average record. The calculated number of blocks will exceed one word.
X'04'	X'68'	X'00'	X'35'	Setup_mdscb_info, convert primary KB or MB in DACRE to blocks. The calculated number of blocks will exceed one word.
X'04'	X'69'	X'00'	X'81'	Setup_mdscb_info, invalid type of space request.
X'04'	X'6A'	X'00'	X'35'	Setup_mdscb_info, convert primary KB or MB in DACRE to tracks. The calculated number of tracks exceeds one word.
X'04'	X'6E'	X'00'	X'35'	Setup_jfcb_info, convert blocks to PDSE blocks for avg. block request.
X'04'	X'6F'	X'00'	X'35'	Setup_mdscb_info, convert blocks to PDSE blocks for avg. block request.
X'04'	X'75'	X'00'	X'A8'	IXVTOC. The minimum allocation quantity specified is greater than the available space.
X'04'	X'76'	X'00'	X'A8'	OSVTOC. The minimum allocation quantity specified is greater than the available space.
X'04'	X'F1'	X'00'	X'00'	Split cylinder extent encountered during VTOC conversion. Message IEC602I issued containing CCHHR of offending DSCB and starting CCHH of the extent in question.
X'08'	X'02'	X'00'	X'07'	Logical error on an SSI call; invalid SSOB format, length.
X'08'	X'13'	SYSCDE1	SYSCDE2	ABEND encountered.
X'08'	X'17'	CVRET	CVSTAT	Invoke CVAFDIR; read index maps, high level VIER, and FMT4 DSCB.
X'08'	X'18'	CVRET	CVSTAT	Invoke CVAFVRF; check for VRF data in VIXM.
X'08'	X'19'	CVRET	CVSTAT	Invoke CVAFDSM for an absolute DSCB request.
X'08'	X'1A'	CVRET	CVSTAT	Invoke CVAFDSM to obtain space for data set.
X'08'	X'1B'	CVRET	CVSTAT	Invoke CVAFDSM; get FMT0 DSCB.
X'08'	X'1C'	CVRET	CVSTAT	Invoke CVAFDSM; add data set name to index and check for duplicate data set name.
X'08'	X'1D'	CVRET	CVSTAT	Invoke CVAFVRF; release buffers.
X'08'	X'1E'	CVRET	CVSTAT	Invoke CVAFVRF; write VRF data to index map.
X'08'	X'1F'	CVRET	CVSTAT	Invoke CVAFDIR; write buffer lists to the volume.
X'08'	X'20'	CVRET	CVSTAT	Invoke CVAFDIR; write FMT1 DSCB.
X'08'	X'21'	X'00'	X'56'	Read FMT4 DSCB and check for duplicate data set name; I/O error during VRF recovery routine execution.

Table 25. DADSM CREATE Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'08'	X'21'	X'00'	X'58'	Read FMT4 DSCB and check for duplicate data set name; OS VTOC—data set already exists on volume, but FMT1 DSCB ID is invalid.
X'08'	X'21'	ECBCC, SNS0	CHSTAT, SNS1	Read FMT4 DSCB and check for duplicate data set name.
X'08'	X'24'	ECBCC, SNS0	CHSTAT, SNS1	Read FMT5 DSCB.
X'08'	X'26'	ECBCC, SNS0	CHSTAT, SNS1	Write FMT1 DSCB.
X'08'	X'27'	ECBCC, SNS0	CHSTAT, SNS1	Write FMT3 DSCB.
X'08'	X'28'	ECBCC, SNS0	CHSTAT, SNS1	Write FMT5 DSCB.
X'08'	X'29'	ECBCC, SNS0	CHSTAT, SNS1	Read FMT0 DSCB.
X'08'	X'2A'	ECBCC, SNS0	CHSTAT, SNS1	Write FMT0 DSCB.
X'08'	X'2B'	ECBCC, SNS0	CHSTAT, SNS1	Write FMT4 DSCB.
X'08'	X'2E'	ECBCC, SNS0	CHSTAT, SNS1	Write FMT2 DSCB.
X'08'	X'2F'	CVRET	CVSTAT	Invoke CVAFDSM; add data set name to index.
X'08'	X'30'	CVRET	CVSTAT	Invoke CVAFDIR; write FMT2 DSCB.
X'08'	X'31'	CVRET	CVSTAT	Invoke CVAFDIR; write FMT3 DSCB.
X'08'	X'32'	CVRET	CVSTAT	Invoke CVAFDIR; check for duplicate data set name.
X'08'	X'33'	X'00'	X'56'	Validate ISAM multivolume request; I/O error during VRF recovery routine execution.
X'08'	X'34'	X'00'	X'58'	OS VTOC channel program issued to read FMT4, read FMT5, check for duplicate data set name, and read FMT1; OS VTOC—data set already exists on volume, but FMT1 DSCB ID is invalid.
X'08'	X'34'	ECBCC, SNS0	CHSTAT, SNS1	OS VTOC channel program issued to read FMT4, read FMT5, check for duplicate data set name, and read FMT1.
X'08'	X'35'	ECBCC, SNS0	CHSTAT, SNS1	Continue duplicate name search for VSE volume
X'08'	X'36'	ECBCC, SNS0	CHSTAT, SNS1	OS VTOC channel program issued to read FMT4, read FMT5, check for duplicate data set name, and read FMT1, after IGGVRF00 disabled the VTOC index.
X'08'	X'37'	ECBCC, SNS0	CHSTAT, SNS1	OS VTOC channel program issued to read FMT4, read FMT5, check for duplicate data set name, and read FMT1, after IGGVRF00 disabled the VTOC index.
X'08'	X'39'	ECBCC, SNS0	CHSTAT, SNS1	VTOC conversion; build FMT5 DSCB(s).
X'08'	X'3A'	ECBCC, SNS0	CHSTAT, SNS1	VTOC conversion; build FMT0 DSCB.
X'08'	X'3F'	ECBCC, SNS0	CHSTAT, SNS1	VTOC conversion; write FMT0 DSCB.
X'08'	X'40'	ECBCC, SNS0	CHSTAT, SNS1	VTOC conversion; write FMT4 DSCB.
X'08'	X'41'	ECBCC, SNS0	CHSTAT, SNS1	VTOC conversion; write FMT5 DSCB.
X'08'	X'44'	ECBCC, SNS0	CHSTAT, SNS1	VTOC conversion; read FMT5 DSCB.

Table 25. DADSM CREATE Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'08'	X'48'	CNVRET	X'00'	PDS directory build. Convert TTR.
X'08'	X'49'	ECBCC, SNS0	CHSTAT, SNS1	PDS directory build. EXCP write of PDS directory.
X'08'	X'4A'	TRKCALCRC	X'9C'	Invoke TRKCALC for track capacity.
X'08'	X'4B'	CNVRT	X'00'	EOF mark write. Convert TTR.
X'08'	X'4C'	ECBCC, SNS0	CHSTAT, SNS1	EOF mark write. EXCP write of EOF mark.
X'08'	X'52'	CVRET	CVSTAT	I/O or CVAF error detected during read of first format 7 DSCB.
X'08'	X'53'	X'00'	X'9D'	VTOC in error. Identifier of first format 7 DSCB is invalid.
X'08'	X'5F'	CVRET	CVSTAT	I/O or CVAF error detected during read of first format 7 DSCB.
X'08'	X'6B'	X'00'	X'00'	Processing selected extents. All free extents processed but not all selected extents.
X'08'	X'6C'	X'00'	X'A3'	Merging selected extents with FS for rear or complete allocation.
X'08'	X'6D'	X'00'	X'A3'	Merging selected extents with FS for front or middle allocation.
X'08'	X'72'	X'04'	DCSSTAT	DASDCALC failed getting half track BLKSIZE to be used in calculating space for a compressed data set.
X'08'	X'72'	DCSRET	DCSRSN	DASDCALC failed getting half track BLKSIZE to be used in calculating space for a compressed data set.
X'08'	X'F0'	X'00'	X'00'	VTOC Conversion after DIRF prevented by customer zap to module IGG0325Z.
X'0C'	X'01'	X'00'	X'01'	Validate input parameters; passed DEB address is zero.
X'0C'	X'01'	X'00'	X'02'	Validate input parameters; passed UCB address is zero.
X'0C'	X'01'	X'00'	X'03'	Validate input parameters; data set name is VTOC index name.
X'0C'	X'01'	X'00'	X'05'	Validate input parameters; data set name is FMT4 DSCB name.
X'0C'	X'01'	X'00'	X'06'	Validate input parameters; data set name to be extended is a VSAM data set.
X'0C'	X'01'	X'00'	X'08'	Validate input parameters; invalid request flag; only the bit specifying “Extend to new volume” may be set on.
X'0C'	X'05'	X'00'	X'0D'	Validate DADSM REALLOC parameter list; RALPDSCB is zero.
X'0C'	X'05'	X'00'	X'0E'	Validate DADSM REALLOC parameter list; RALUCB is zero.

Table 25. DADSM CREATE Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'0C'	X'05'	X'00'	X'0F'	Validate DADSM REALLOC parameter list; RALUCB is dummy SMS UCB.
X'0C'	X'05'	X'00'	X'10'	Validate DADSM REALLOC parameter list; RALFLAG contains nonzero reserved flags.
X'0C'	X'05'	X'00'	X'11'	Validate DADSM REALLOC parameter list; RALABS not supported for this version of the parameter list.
X'0C'	X'05'	X'00'	X'40'	Invalid PD1FMTID or DS1FMTID field.
X'0C'	X'05'	X'00'	X'8A'	VTOCENQD keyword of REALLOC macro invalid for ALLOC=ABS.
X'0C'	X'06'	X'00'	X'18'	Validate branch entry parameter list; DACEXPTR is zero.
X'0C'	X'06'	X'00'	X'19'	Validate branch entry parameter list; DACJFCB, DACMDSCB, or DACADSCB not set (interface flags).
X'0C'	X'06'	X'00'	X'1A'	Validate branch entry parameter list; more than one interface flag specified.
X'0C'	X'06'	X'00'	X'1B'	Validate branch entry parameter list; VIO not supported for MDSCB or ADSCB interface.
X'0C'	X'06'	X'00'	X'1C'	Validate branch entry parameter list; DACRSRBT must be zero.
X'0C'	X'06'	X'00'	X'1D'	Validate branch entry parameter list; DACRSRV1 must be zero.
X'0C'	X'06'	X'00'	X'1E'	Validate branch entry parameter list; DACRSRV2 must be zero.
X'0C'	X'06'	X'00'	X'1F'	Validate branch entry parameter list; DACRSRV3 must be zero.
X'0C'	X'06'	X'00'	X'20'	Validate branch entry parameter list; DACRSRV4 must be zero.
X'0C'	X'06'	X'00'	X'21'	Validate branch entry parameter list; DACRSRVF must be zero.
X'0C'	X'06'	X'00'	X'22'	Validate branch entry parameter list; DACUCB is zero.
X'0C'	X'06'	X'00'	X'23'	Validate branch entry parameter list; DACUCB is dummy SMS UCB.
X'0C'	X'06'	X'00'	X'24'	Validate branch entry parameter list; DACGENMP set but DACSECMP is zero.
X'0C'	X'06'	X'00'	X'25'	Validate branch entry parameter list; DACPROFM set but DACSECMP is not zero.
X'0C'	X'06'	X'00'	X'7C'	Validate branch entry parameter list; DACRSRV5 must be zero.
X'0C'	X'06'	X'00'	X'86'	Validate branch entry parameter list; DACLRGEX is on, must be off in ADSCB interface.
X'0C'	X'06'	X'00'	X'90'	Validate branch entry parameter list; PDSE cannot be VIO.

Table 25. DADSM CREATE Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'0C'	X'06'	X'00'	X'91'	Validate branch entry parameter list; DACRSRV6 must be zero.
X'0C'	X'06'	X'00'	X'92'	Validate branch entry parameter list; DACSKRSV must be zero.
X'0C'	X'06'	X'00'	X'9E'	Validate branch entry parameter list; HFS data set bit without PDSE bit.
X'0C'	X'06'	X'00'	X'A5'	Validate branch entry parameter list; creation of an HFS data set is requested, but the necessary security environment is not active. HFS data sets cannot be created unless a security package such as RACF is active.
X'0C'	X'06'	X'00'	X'A6'	The maximum extent value is specified, but X37 reduction is not specified.
X'0C'	X'06'	X'00'	X'A7'	Minimum allocation quantity is supplied but the 'best fit' is not specified.
X'0C'	X'07'	X'00'	X'12'	Check PQTY and MINAU when rounding to a cylinder boundary; invalid PQTY or MINAU for request requiring cylinder boundaries.
X'0C'	X'08'	X'00'	X'13'	Validate passed UCB address; register 1 does not contain a pointer to a fullword containing the UCB address.
X'0C'	X'08'	X'00'	X'14'	Validate passed UCB address; fullword pointed to by register 1 contains dummy SMS UCB.
X'0C'	X'09'	X'00'	X'26'	Validate JFCB extension; DACJFCBP is zero.
X'0C'	X'09'	X'00'	X'27'	Validate JFCB extension; DACIJRS1 must be zero.
X'0C'	X'09'	X'00'	X'28'	Validate JFCB extension; DACDSABP is zero for an ISAM data set.
X'0C'	X'09'	X'00'	X'29'	Validate JFCB extension; DACSECVL set but DACVOLSQ or DACDSSNM is zero.
X'0C'	X'09'	X'00'	X'31'	Validate JFCB extension; DACAVGRC value is invalid.
X'0C'	X'09'	X'00'	X'32'	Validate JFCB extension; DACSECVL cannot be set for VIO data sets.
X'0C'	X'09'	X'00'	X'79'	Validate JFCB extension; DACIJRS2 must be zero.
X'0C'	X'09'	X'00'	X'7A'	Validate JFCB extension; DACIJRS3 must be zero.
X'0C'	X'09'	X'00'	X'7B'	Validate JFCB extension; DACAVRCP is on, but DACAVGRC is zero.
X'0C'	X'09'	X'00'	X'7D'	Validate JFCB extension; DACLRGEX on condition is only valid for cylinder or track requests.
X'0C'	X'09'	X'00'	X'87'	Validate JFCB extension; If DACLRGEX is on, JFCBCTRI = X'40' is invalid.
X'0C'	X'09'	X'00'	X'93'	Validate JFCB extension; KEYLEN must be zero for PDSEs.

Table 25. DADSM CREATE Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'0C'	X'09'	X'00'	X'A9'	JFCB. The minimum allocation space value is greater than the primary space allocation value.
X'0C'	X'0A'	X'00'	X'2A'	Validate model DSCB extension; DACDSCBM is zero.
X'0C'	X'0A'	X'00'	X'2B'	Validate model DSCB extension; must set space request flag.
X'0C'	X'0A'	X'00'	X'2C'	Validate model DSCB extension; more than one space request flag set.
X'0C'	X'0A'	X'00'	X'2D'	Validate model DSCB extension; DACIMRS1 must be zero.
X'0C'	X'0A'	X'00'	X'2E'	Validate model DSCB extension; DACIMRS2 must be zero.
X'0C'	X'0A'	X'00'	X'7D'	Validate model DSCB extension; DACLRGEX is on, but DACMBRQ and DACKBRQ are both off.
X'0C'	X'0A'	X'00'	X'84'	Validate model DSCB extension; DACDSSNM cannot be blank for SMS-managed data sets.
X'0C'	X'0A'	X'00'	X'93'	Validate model DSCB extension; KEYLEN must be zero for PDSEs.
X'0C'	X'0A'	X'00'	X'A9'	MDSCB. The minimum allocation space value is greater than the primary space allocation value.
X'0C'	X'0B'	X'00'	X'2F'	Validate absolute DSCB extension; DACDSCB1 is zero.
X'0C'	X'0C'	X'00'	X'30'	Validate model DSCB extension; DACRTPTR cannot be zero for VIO data sets.
X'0C'	X'71'	UCBCAPTRC	UCBCAPTRS	IOSCAPU failed when attempting to capture the UCB passed by the caller.
X'0C'	X'74'	X'00'	X'80'	The user value specified, DACXTMAX, exceeds the allowable extents for a data set.
X'10'	X'05'	X'00'	X'0B'	Validate DADSM REALLOC parameter list; invalid RALPLID field.
X'10'	X'05'	X'00'	X'0C'	Validate DADSM REALLOC parameter list; invalid RALNGTH field.
X'10'	X'06'	X'00'	X'15'	Validate branch entry parameter list; invalid DACPLID field.
X'10'	X'06'	X'00'	X'16'	Validate branch entry parameter list; invalid DACPVER field.
X'10'	X'06'	X'00'	X'17'	Validate branch entry parameter list; invalid DACPLEN field.

Table 26 on page 64 describes abbreviations for DADSM CREATE diagnostic information.

Table 26. Abbreviation Descriptions for DADSM CREATE Diagnostic Information (Register 0)

Abbreviation	Description
CDMRET	Return code from CDM. For explanation see “CDM Reason Codes” on page 408.
CDMRSN	Reason code from CDM. For explanation see “CDM Reason Codes” on page 408.
CDMRSN1,2	CDM reason code (low order 2 bytes). For explanation see “CDM Reason Codes” on page 408.
CHSTAT	Channel status from I/O. For a description of the status, see *z/OS DFSMSdfp Advanced Services*.
CNVRET	Return code from TTR Convert Routine. For a description of the return code, see *z/OS DFSMSdfp Advanced Services*.
COMMRC	DADSM common return code. For a description of that return code, see Table 53 on page 104.
COMMRS	DADSM common reason code. For a description of that reason code, see Table 53 on page 104.
CVRET	Return code from the CVAF function. For a description of the return code, see *z/OS DFSMSdfp Advanced Services*.
CVSTAT	CVSTAT field of the CVAF parameter list. For a description of the fields, see “CVSTAT Field Codes” on page 40.
DCSRET	See return codes in “DASD Calculation Services Return, Reason, and Status Codes” on page 109.
DCSRSN	See reason codes in “DASD Calculation Services Return, Reason, and Status Codes” on page 109.
DCSSTAT	See status codes in “DASD Calculation Services Return, Reason, and Status Codes” on page 109.
ECBCC	ECB completion code. For a description of the return code, see *z/OS DFSMSdfp Advanced Services*.
ENQRC	Return code from ENQ function. For a description of the return code, see *z/OS MVS Programming: Assembler Services Reference ABE-HSP*.
PREREA1,2	DADSM IGGPRE00 preprocessing exit reason code (low order 2 bytes). For a description of the reason code, see *z/OS DFSMS Installation Exits*.
RACREAS	RACF reason code. See Diagnostic Code for request type.
RACRET	RACF return code. See Diagnostic Code for request type.
SNS0	First sense byte from the IOB. For a description of the sense byte, see *z/OS DFSMSdfp Advanced Services*
SNS1	Second sense byte from the IOB. For a description of the sense byte, see *z/OS DFSMSdfp Advanced Services*
SYSCDE1,2	System completion code (low order 2 bytes). For a description of the completion code, see *z/OS DFSMS Macro Instructions for Data Sets*.
TRKCALCRC	Return code from TRKCALC routine. For a description of the return code, see the return codes from the TRKCALC macro in *z/OS DFSMSdfp Advanced Services*.
UCBCAPTRC	Return code from IOSCAPU service.
UCBCAPTRS	Reason code from IOSCAPU service.
VDSS1,2	VDSS error reason code (low order 2 bytes). For a description of the VDSS1 and VDSS2 field values, convert the reason code to decimal and see “Storage Management Subsystem Reason Codes” on page 439.

DADSM Extend to New Volume Function Return and Diagnostic Codes

For some errors during secondary allocation on a new volume, DADSM Extend might issue message **IEC614I**. See “DADSM CREATE (Allocate) Return and Diagnostic Codes” on page 47 and Table 25 on page 50 for a description of the DADSM Extend diagnostic information.

DADSM Extend to Same Volume Diagnostic Codes

This section defines the diagnostic codes that are returned by the EXTEND to same volume function of DADSM. These diagnostic codes might be displayed in messages.

If an error occurs, DADSM might issue a message **IEC614I**, which would contain a return code and a diagnostic code.

Note: This is a complete list of DADSM Extend to Same Volume diagnostic codes.

In Table 27, the Extend diagnostic code (Extend to same volume only) might appear in an **IEC614I** message. The abbreviations that are used in Table 27 are defined in Table 28 on page 68.

Table 27. DADSM Extend Diagnostic Information

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'04'	X'15'	X'00'	X'00'	AVG REC/BLK length is greater than 65535.
X'04'	X'16'	X'00'	X'00'	Neither AVG REC or AVG BLK with BLKSIZE of zero.
X'04'	X'17'	X'00'	X'17'	AVG REC pre-check determines that the calculated number of blocks will exceed one word.
X'04'	X'17'	X'00'	X'18'	The calculated number of blocks will exceed one word; conversion to AVGREC=K bytes results in an overflow condition.
X'04'	X'17'	X'00'	X'19'	The calculated number of blocks will exceed one word; conversion to AVGREC=M bytes results in an overflow condition.
X'04'	X'21'	X'00'	X'10'	Validate space request size. Data set exceeds 65534 tracks.
X'04'	X'21'	X'00'	X'11'	Validate space request size. Data set exceeds size of volume.
X'04'	X'21'	X'00'	X'12'	Validate space request size. Data set exceeds maximum allowable size.
X'04'	X'22'	X'00'	X'15'	Search for space in free-space DSCBs. Space could not be obtained from up to 5 extents.
X'04'	X'22'	X'00'	X'16'	Search for space in free-space DSCBs. No free-space in first F5/F7 DSCB.
X'04'	X'25'	CVRET	CVSTAT	Indexed VTOC path. CVAFVSM Allocate for Preferred RTA request.
X'04'	X'26'	CVRET	CVSTAT	Indexed VTOC path. CVAFVSM Allocate track request for Non-Preferred RTA request.

Table 27. DADSM Extend Diagnostic Information (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'04'	X'27'	CVRET	CVSTAT	Indexed VTOC path. CVAFVSM Allocate cylinder request for Non-Preferred RTA request.
X'04'	X'39'	X'00'	X'00'	VTOC conversion; build FMT5 DSCBs.
X'04'	X'3D'	X'00'	X'00'	VTOC conversion; IGG0325P requested FMT5 space update.
X'04'	X'4A'	X'00'	X'00'	Requested secondary size exceeds size of volume.
X'04'	X'F1'	X'00'	X'00'	Split cylinder extent encountered during VTOC conversion. Message IEC602I issued containing CCHHR of offending DSCB and starting CCHH of the extent in question.
X'08'	X'10'	X'00'	X'13'	Validate F1 DSCB for PDSE data set. DS1FMTID is not X'F1'.
X'08'	X'10'	X'00'	X'14'	Validate F1 DSCB for PDSE data set. PDSE flag not set.
X'08'	X'11'	X'00'	X'13'	Validate F1 DSCB for non-PDSE data set. DS1FMTID is not X'F1'.
X'08'	X'1B'	CVRET	CVSTAT	CVAFDIR read F1 DSCB for PDSE data set.
X'08'	X'1C'	CVRET	CVSTAT	CVAFDIR read F1 DSCB for non-PDSE data set.
X'08'	X'1D'	CVRET	CVSTAT	CVAFDIR read F3 DSCB.
X'08'	X'1E'	X'00'	X'00'	DEB was not provided for data set that requires one.
X'08'	X'1F'	X'00'	X'00'	Conversion routine for CCHHR to TTR (CVTPRLTV) failed.
X'08'	X'24'	ECBCC SNS0	CHSTAT SNS1	VTOC path invokes EXCP to read F5/F5 DSCB.
X'08'	X'25'	CVRET	CVSTAT	Indexed VTOC path. CVAFVSM Allocate for Preferred RTA request.
X'08'	X'26'	CVRET	CVSTAT	Indexed VTOC path. CVAFVSM Allocate track request for Non-Preferred RTA request.
X'08'	X'27'	CVRET	CVSTAT	Indexed VTOC path. CVAFVSM Allocate cylinder request for Non-Preferred RTA request.
X'08'	X'29'	COMMRC	COMMRS	DADSM Common function. Convert from RTA to CCHH to build F1/F3 DSCB.
X'08'	X'2A'	CVRET	CVSTAT	Indexed VTOC path. Convert Catalog request. CVAFVSM Allocate ABST request.
X'08'	X'2D'	COMMRC	COMREA	Convert CCHH to RTA to obtain the preferred RTA for the data set.
X'08'	X'30'	ECBCC SNS0	CHSTAT SNS1	OSVTOC path - Invoke EXCP to read F5 DSCB.
X'08'	X'31'	ECBCC SNS0	CHSTAT SNS1	OSVTOC path - Invoke EXCP to read F7 DSCB.
X'08'	X'32'	X'00'	X'00'	OSVTOC path - Invalid track allocation for ICF Extend.
X'08'	X'34'	COMMRC	COMREA	OSVTOC path - Convert F5 DSCB into work area.
X'08'	X'35'	COMMRC	COMREA	OSVTOC path - Convert F7 into work area.
X'08'	X'36'	COMMRC	COMREA	OSVTOC path - Convert F5 into work area.

Table 27. DADSM Extend Diagnostic Information (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'08'	X'37'	COMMRC	COMREA	OSVTOC path - Convert F7 into work area.
X'08'	X'38'	COMMRC	COMREA	OSVTOC path - Convert F5 DSCB into work area.
X'08'	X'39'	ECBCC, SNS0	CHSTAT, SNS1	VTOC conversion; build FMT5 DSCB(s).
X'08'	X'3A'	ECBCC, SNS0	CHSTAT, SNS1	VTOC conversion; build FMT0 DSCB.
X'08'	X'3F'	ECBCC, SNS0	CHSTAT, SNS1	VTOC conversion; write FMT0 DSCB.
X'08'	X'40'	ECBCC, SNS0	CHSTAT, SNS1	VTOC conversion; write FMT4 DSCB.
X'08'	X'41'	ECBCC, SNS0	CHSTAT, SNS1	VTOC conversion; write FMT5 DSCB.
X'08'	X'44'	ECBCC, SNS0	CHSTAT, SNS1	VTOC conversion; read FMT5 DSCB.
X'08'	X'45'	COMMRC	COMREA	OSVTOC path - Convert F7 DSCB into work area.
X'08'	X'46'	ECBCC SNS0	CHSTAT SNS1	OSVTOC path - Invoke EXCP to read F5 DSCB.
X'08'	X'47'	ECBCC SNS0	CHSTAT SNS1	OSVTOC path - Invoke EXCP to read F7 DSCB.
X'08'	X'48'	ECBCC SNS0	CHSTAT SNS1	OSVTOC path - Invoke EXCP to write F5 DSCB.
X'08'	X'49'	ECBCC SNS0	CHSTAT SNS1	OSVTOC path - Invoke EXCP to write F7 DSCB.
X'08'	X'4F'	X'00'	X'00'	OSVTOC path invokes EXCP to search for an F0 DSCB that will be used to create a new free-space DSCB.
X'08'	X'50'	CVRET	CVSTAT	IXVTOC path - Invoke CVAFDIR to write buffer lists for VRF data.
X'08'	X'51'	CVRET	CVSTAT	IXVTOC path - Invoke CVAFDIR to write reordered buffer lists.
X'08'	X'52'	CVRET	CVSTAT	IXVTOC path - Invoke CVAFDIR to zero VRF data.
X'08'	X'53'	CVRET	CVSTAT	Invoke CVAFDIR to write F1 DSCB.
X'08'	X'54'	CVRET	CVSTAT	Invoke CVAFDIR to write F3 DSCB.
X'08'	X'55'	CVRET	CVSTAT	Invoke CVAFDIR to write last F3 DSCB.
X'08'	X'56'	CVRET	CVSTAT	IXVTOC path - Invoke CVAFDSM to get F0 DSCB for FN DSCB.
X'08'	X'57'	CVRET	CVSTAT	OSVTOC path - Invoke CVAFDIR to get F0 DSCB for FN DSCB.
X'08'	X'58'	CVRET	CVSTAT	IXVTOC path - Invoke CVAFDSM to get F0 DSCB for F3 DSCB.
X'08'	X'59'	CVRET	CVSTAT	OSVTOC path - Invoke CVAFDIR to get F0 DSCB for F3 DSCB.
X'08'	X'61'	COMMRC	COMREA	Convert the EDT02 to EDT01.
X'08'	X'62'	COMMRC	COMMRS	DADSM common function. OSVTOC path, convert F7 DSCB into work area.
X'08'	X'63'	COMMRC	COMMRS	DADSM common function. OSVTOC path, convert F5 DSCB into work area.
X'08'	X'F0'	X'00'	X'00'	VTOC Conversion after DIRF prevented by customer zap to module IGG0325Z.

DADSM/CVAF Diagnostic Aids

Table 28 describes abbreviations for DADSM EXTEND diagnostic information.

Table 28. Abbreviation Descriptions for DADSM EXTEND Diagnostic Information

Abbreviation	Description
CHSTAT	Channel status from I/O. For a description of the status, see *z/OS DFSMSdfp Advanced Services*.
COMMRC	DADSM Common Function return code. For a description of that return code, see Table 53 on page 104.
COMMRS	DADSM Common Function reason code. For a description of that reason code, see Table 53 on page 104.
CVRET	Return code from the CVAF function. For a description of the return code, see *z/OS DFSMSdfp Advanced Services*.
CVSTAT	CVSTAT field of the CVAF parameter list. For a description of the fields, see “CVSTAT Field Codes” on page 40.
ECBCC	ECB completion code. For a description of the code, see *z/OS DFSMS: Using Data Sets*.
PREREA1,2	DADSM IGGPRE00 preprocessing exit reason code (low order 2 bytes). For a description of the reason code, see *z/OS DFSMS Installation Exits*.
SNS0	First sense byte from the IOB. For a description of the sense byte, see *z/OS DFSMSdfp Advanced Services*
SNS1	Second sense byte from the IOB. For a description of the sense byte, see *z/OS DFSMSdfp Advanced Services*
TRKCALCRC	Return code from TRKCALC routine. For a description of the return code, see the return codes from the TRKCALC macro in *z/OS DFSMSdfp Advanced Services*.

DADSM Extend Return Codes

Table 29, Table 30 on page 69, and Table 31 on page 69 describe DADSM extend return codes.

End-of-Volume

Table 29. DADSM End-of-Volume Return Codes

Return Code (R15)	Description
1 (X'01')	Successful secondary allocation on the current volume.
4 (X'04')	Successful secondary allocation on the new volume.
-1 (X'-01')	Error in secondary allocation on the current volume. • Fewer than three free VTOC records (format-0 DSCBs) • Quantity not available • Too many extents (more than 16) for the data set on this volume
-4 (X'-04')	Error in secondary allocation on the new volume. • Duplicate name on the new volume • Fewer than three free VTOC records (format-0 DSCBs) • Quantity not available
-8 (X'-08')	Unable to convert VTOC to standard format, or a volume with the alternate tracks on cylinder 0 was encountered (VSE allocation).
-12 (X'-0C')	Unable to define the new volume to RACF.
-16 (X'-10')	Permanent I/O error or unexpected CVAF error return code.
-20 (X'-14')	Installation rejected the request.

DADSM Catalog Management Return Codes

Table 30. DADSM Catalog Management Return Codes

Return Code (R15)	Description
2(X'02')	The 'open for input' indicator was found on the catalog while a partial release request was being processed. Partial release processing cannot occur if there are other users opened to the data set.
4(X'04')	A partial release request was attempted for a VSAM data set that is not an Extended Format VSAM cluster. Partial release processing is only supported for Extended Format data sets.
6(X'06')	The data set name passed to a VSAM partial release of a VSAM data set.
8(X'08')	A DADSM error occurred during a partial release of a VSAM data set.
10(X'0A')	An attempt was made to perform partial release processing against a data set defined with guaranteed space. Guaranteed space data sets are not supported for partial release processing.
129 (X'81')	Return code for catalog caller. Successful secondary allocation on the current volume.
161 (X'A1')	Return code for catalog caller. Successful secondary allocation on the current volume.
-8 (X'-08')	Unable to convert the VTOC to standard format.
-16 (X'-10')	Permanent I/O error or unexpected CVAF error return code.
-20 (X'-14')	Installation exit rejected the request.
-129 (X'-81')	Return code for catalog caller. Error in secondary allocation on the new volume. • Fewer than three free VTOC records (format-0 DSCBs) • Quantity not available • Too many extents for the data space on this volume
-161 (X'-A1')	Return code for catalog caller. Error in secondary allocation on the new volume. • Fewer than three free VTOC records (format-0 DSCBs) • Quantity not available • Too many extents for the data space on this volume

DADSM PDSE Management Return Codes

Table 31. DADSM PDSE Management Return Codes

Return Code (R15)	Description
3 (X'03')	Successful secondary allocation for PDSE on the current volume.
-3 (X'-03')	Error in secondary allocation for PDSE.
-8 (X'-08')	Unable to convert VTOC to standard format, or a volume with the alternate tracks on cylinder 0 was encountered (VSE allocation).
-16 (X'-10')	Permanent I/O error or unexpected CVAF error return code.
-20 (X'-14')	Installation rejected the request.

DADSM ISJF Function Return and Diagnostic Codes

For some errors during secondary allocation on the current volume, the DADSM ISJF (Invoke SJF) function may issue message **IEC614I**. Table 32 shows the return codes that might appear in this message.

Table 32. DADSM ISJF Return Codes

Return Code (R15)	Description
00 (X'00')	All requested keywords specified for data set.
04 (X'04')	Not all requested keywords specified for data set.
08 (X'08')	SJF failure.
12 (X'0C')	Invalid parameter or request.
16 (X'10')	Invalid parameter list.

Table 33 describes the diagnostic information that appears in message **IEC614I**, which contains the return code, subfunction identifier, subfunction return code, and subfunction reason code.

Table 33. DADSM ISJF Diagnostic Information

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Return Code	**Subfunction Identifier**	**Subfunction Return Code[1]**	**Subfunction Reason Code[1]**	**Description**
X'08'	X'02'	SJFRS1	SJFRS2	Invoke SJF; SJF return code = RC04 and SJF reason code is not 'keyword not specified'.
X'08'	X'03'	SJFRC	X'00'	Invoke SJF; received error return code other than RC20.
X'08'	X'04'	SJFRS1	SJFRS2	Invoke SJF; SJF return code = RC20.
X'0C'	X'01'	X'00'	X'04'	Verify parameters; DDNAME address is zero.
X'0C'	X'01'	TYPE	X'05'	Verify parameters; request type invalid.
X'0C'	X'01'	X'00'	X'06'	Verify parameters; request is for AVGREC value but output variable address is zero.
X'0C'	X'01'	X'00'	X'07'	Verify parameter list; TCB address is zero.
X'0C'	X'01'	X'00'	X'08'	Verify parameter list; data set name address is zero.
X'0C'	X'01'	X'00'	X'09'	Verify parameter list; volume serial address is zero.
X'10'	X'01'	X'00'	X'01'	Verify parameter list; eye-catcher invalid.
X'10'	X'01'	X'00'	X'02'	Verify parameter list; version invalid.
X'10'	X'01'	X'00'	X'03'	Verify parameter list; length invalid.

Note:

1. The following key describes the preceding abbreviations. For an explanation of the SJF return and reason codes, contact your programming support personnel .

Abbreviation	Description
SJFRC	Return code from SJF
SJFRS1,2	Reason code from SJF (2 bytes)
TYPE	Invalid request type passed to the ISJF function

DADSM LSPACE Return and Diagnostic Codes

Table 34 describes the DADSM LSPACE return codes.

Table 34. DADSM LSPACE Return Codes

Return Code (R15)	Description
0 (X'00')	Successful LSPACE processing.
4 (X'04')	I/O error while reading DSCBs, or unexpected CVAF error return code. I/O Timeout error while reading volume label.
8 (X'08')	Last allocation on this volume made under VSE, or a DADSM function has prematurely terminated and VTOC errors may exist.
12 (X'0C')	UCB address invalid. UCB not for direct access device. UCB not-ready bit is on, indicating device not ready; UCBVTOC=0, indicating volume not mounted. If a message area was specified, the reason associated with the return code is stored there.
16 (X'10')	Invalid message area address or SMF indicator.

Table 35 describes additional diagnostic information that is returned by LSPACE in register 0. The abbreviations that are used in Table 35 are defined in Table 36. If the return code is not 16, it is also returned in the LSPACE parameter list at the locations that are identified in Table 37 on page 73.

DADSM/CVAF Diagnostic Aids

Table 35. DADSM LSPACE Diagnostic Information (Register 0)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Return Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
04 (X'04')	X'06'	ECBCC	X'00'	Error reading volume label.
04 (X'04')	X'07'	AOMRC	AOMRSC	Error reading volume label with AOM.
04 (X'04')	X'07'	AOMRC	IOSCOD	Error reading volume label due to a timeout condition.
04 (X'04')	X'07'	AOMRC	X'52'	Start pending in subchannel.
04 (X'04')	X'07'	AOMRC	X'53'	I/O timed out canceled by IOS.
04 (X'04')	X'80'	DIRRET	CVSTAT	Error getting format-4/space maps.
04 (X'04')	X'81'	DSMRET	CVSTAT	Error getting free extents.
04 (X'04')	X'82'	DSMRET	CVSTAT	Error getting format-0 count.
04 (X'04')	X'83'	DSMRET	CVSTAT	Error getting VIR count.
04 (X'04')	X'84'	VRFRET	CVSTAT	Error checking for VRF.
08 (X'08')	X'80'	X'00'	X'00'	Error: DIRF bit found on or the free space indicators (FMT5/FMT7 DSCBs) are invalid.
12 (X'0C')	X'01'	X'00'	X'03'	Invalid LSPACE flag.
12 (X'0C')	X'01'	X'00'	X'04'	Not authorized for SMF.
12 (X'0C')	X'01'	X'00'	X'05'	Bad MSG/EXPMSG/DATA area storage key.
12 (X'0C')	X'01'	X'00'	X'06'	Bad format-4 area storage key.
12 (X'0C')	X'01'	X'00'	X'07'	UCB not found.
12 (X'0C')	X'01'	X'00'	X'08'	UCB not direct access device.
12 (X'0C')	X'01'	X'00'	X'09'	UCB VTOC pointer is zero.
12 (X'0C')	X'02'	X'00'	X'00'	Invalid UCB status.
12 (X'0C')	X'03'	ENQRET	X'00'	Failed ENQ on SYSZDMNT.
16 (X'10')	X'01'	X'00'	X'01'	Bad parameter list storage key.
16 (X'10')	X'01'	X'00'	X'02'	Bad parameter list ID.

Table 36. Abbreviation Descriptions for DADSM LSPACE Return and Diagnostic Codes

Abbreviation	Description
CVSTAT	CVSTAT field of CVAF parameter list. For a description of the field, see "CVSTAT Field Codes" on page 40.
DIRRET	Return code from CVAFDIR. For a description of the return code, see *z/OS DFSMSdfp Advanced Services*.
DSMRET	Return code from CVAFDSM. For a description of the return code, see *z/OS DFSMSdfp Advanced Services*
ECBCC	ECB completion code. For a description of the completion code, see *z/OS DFSMS: Using Data Sets*.
ENQRET	Return code from ENQ. For a description of the return code, see *z/OS MVS Programming: Assembler Services Reference ABE-HSP*.
N/A	Not Applicable.
VRFRET	Return code from CVAFVRF.

Table 37. DADSM—LSPACE Parameter List Diagnostic Information Fields

LSPACE Parameter List Offset	Diagnostic Information
8 (X'08')	LSPACE return code
9 (X'09')	LSPACE subfunction
10 (X'0A')	LSPACE subfunction return code
11 (X'0B')	LSPACE subfunction reason code

DADSM OBTAIN Function Return Codes

Table 38 describes DADSM OBTAIN function return codes.

Table 38. DADSM OBTAIN Function Return Codes

Return Code (R15)	Description
0 (X'00')	Successful completion of OBTAIN routine.
4 (X'04')	The required volume was not mounted.
8 (X'08')	The F1 DSCB was not found in the VTOC of the specified volume.
12 (X'0C')	A permanent I/O error or an invalid format-1 or format-4 DSCB was encountered when processing the specified volume, or an unexpected CVAF error return code was encountered.
16 (X'10')	The pointer to the work area passed by the caller was invalid.
20 (X'14')	The absolute address passed (seek option) was not within the boundaries of the VTOC.

DADSM Partial Release Return and Diagnostic Codes

Control returns to the instruction following the last instruction generated by the PARTREL macro. Register 15 contains the applicable DADSM return code, as shown in Table 39.

DADSM PARTREL returns 4 bytes of diagnostic information in register 0. See Table 40 on page 76 for a description of this information.

If an error occurs, DADSM might issue message **IEC614I**, which would contain failure-related status information.

- If an error results from a CVAF function, the subfunction return code field contains the CVAF return code, and the subfunction reason code field contains the CVAF status code (CVSTAT).
- If an error results from the execution of an EXCP channel program (DADSM sometimes uses EXCP channel programs to access DSCBs), the subfunction return code and reason code fields contain either:
  - The ECB completion code and the CSW channel status, if the ECB completion code is not X'41' and the channel status is not zero.
  - Two sense bytes from the IOB, if the ECB completion code is X'41' and there is no channel status.
- If an error results from an RACF invocation, the subfunction return code and reason code fields contain the RACF return code and reason code.

Table 39 describes the conditions that are indicated by the DADSM return codes.

Tip: This is a cumulative list of DADSM partial release return codes. Some of these codes may not apply to the PARTREL macro.

Table 39. DADSM Partial Release Function Return Codes

Return Code (R15)	Description
0 (X'00')	Function successful.
2 (X'02')	Unable to find extent in the VTOC.
4 (X'04')	Unable to find extent in the VTOC.
8 (X'08')	Either the required SYSZTIOT or SYSDSN ENQ failed, or an unrelated DEB indicates that another DCB is open to the data set.
12 (X'0C')	Invalid parameter list.
16 (X'10')	One of the following conditions occurred: • A permanent I/O error occurred. • CVAF provided an unexpected return code. • An installation exit rejected the request. • An I/O error occurred while the tracks to be released were being erased.
20 (X'14')	DSN, or DSN pointer is invalid.
24 (X'18')	Invalid UCB pointer.
28 (X'1C')	Specified DSORG is not supported.
32 (X'20')	No room in the VTOC.
36 (X'24')	Invalid TIOT=NOENQ request; caller does not have exclusive use of SYSDSN.
40 (X'28')	An error occurred while SMS was processing the request.
44 (X'2C')	'CLOSE' was the caller of partial release. The customer rejected the partial release request by passing a nonzero return code back to partial release from the preprocessing exit (IGGPRE00).

Table 40 describes the conditions that are indicated by the diagnostic information that is returned in register 0. The abbreviations that are used in Table 40 are defined in Table 41 on page 82.

DADSM/CVAF Diagnostic Aids

Table 40. DADSM PARTREL Diagnostic Information (Register 0)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Return Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'00'	X'16'	SYSCDE1	SYSCDE2	PARTREL ESTAE received control after function completed.
X'00'	X'1A'	SYSCDE1	SYSCDE2	PARTREL ESTAE received control after function completed.
X'04'	X'04'	X'00'	X'01'	Close/checkpoint called PARTREL for a PDSE.
X'04'	X'08'	ENQRET	X'0C'	ENQ SYSZTIOT RET=USE failed.
X'04'	X'09'	ENQRET	X'0E'	ENQ SYSDSN RET=USE failed.
X'04'	X'09'	ENQRET	X'0F'	ENQ SYSDSN RET=CHNG failed.
X'04'	X'1B'	CNVTRET	X'17'	DS1LSTAR not in DEB.
X'04'	X'27'	X'00'	X'0B'	VTOC full.
X'04'	X'28'	X'00'	X'14'	DSN invalid; VTOC index data set.
X'04'	X'29'	ENQRET	X'0C'	ENQ SYSZTIOT RET=USE failed.
X'04'	X'2A'	ENQRET	X'0D'	ENQ SYSZTIOT RET=CHNG failed.
X'04'	X'2B'	ENQRET	X'0E'	ENQ SYSDSN RET=USE failed.
X'04'	X'2C'	ENQRET	X'0F'	ENQ SYSDSN RET=CHNG failed.
X'04'	X'2E'	X'00'	X'13'	DSORG not supported.
X'04'	X'2E'	X'00'	X'15'	DSN invalid; data set not found.
X'04'	X'2E'	X'00'	X'16'	DSN invalid; not format-1 DSCB.
X'04'	X'30'	X'00'	X'11'	DS1LSTAR not found in format-1 extents.
X'04'	X'30'	X'00'	X'12'	DS1LSTAR not found in format-1/format-3 extents.
X'04'	X'32'	X'00'	X'11'	DS1LSTAR not found in format-1 extents.
X'04'	X'32'	X'00'	X'12'	DS1LSTAR not found in format-1/format-3 extents.
X'04'	X'33'	X'00'	X'10'	Duplicate DEB found.
X'04'	X'34'	CNVTRET	X'17'	DS1LSTAR not in DEB.
X'04'	X'36'	X'00'	X'11'	DS1LSTAR not found in format-1 extents.
X'04'	X'37'	X'00'	X'11'	DS1LSTAR not found in format-1 extents.
X'04'	X'3F'	X'00'	X'12'	DS1LSTAR not found in format-1/format-3 extents.
X'04'	X'41'	PRERET	X'1A'	Preprocessing installation exit reject.
X'04'	X'43'	PRERET	X'1A'	Preprocessing installation exit reject.
X'04'	X'4D'	ERASRET	X'1B'	Error occurred while erasing released extents.
X'04'	X'53'	X'00'	X'0B'	VTOC full.
X'08'	X'02'	CVRET	CVSTAT	Invoke CVAFDIR; error occurred while reading format-1 DSCB (key and data).
X'08'	X'03'	CVRET	CVSTAT	Invoke CVAFDIR; error occurred while reading format-1 DSCB (using key).
X'08'	X'0A'	CVRET	CVSTAT	Invoke CVAFDIR; error occurred while reading format-3 DSCB.
X'08'	X'0C'	CDMRET	CDMREAS	SMS error occurred during connect directory to PDSE.

Table 40. DADSM PARTREL Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Return Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'08'	X'0D'	CDMRET	CDMRSN	SMS error occurred during connect directory to an HFS data set.
X'08'	X'0E'	CDMRET	CDMREAS	SMS error occurred during disconnect directory from PDSE.
X'08'	X'16'	SYSCDE1	SYSCDE2	PARTREL ESTAE received control before function completed.
X'08'	X'1A'	SYSCDE1	SYSCDE2	PARTREL ESTAE received control before function completed.
X'08'	X'1C'	SNS0, ECBSTAT	SNS1, ECBSTAT	I/O error occurred while reading first format-1 DSCB (key and data).
X'08'	X'21'	CVRET	CVSTAT	Invoke CVAFDIR; error occurred while reading format-4 DSCB with space maps.
X'08'	X'22'	CVRET	CVSTAT	Invoke CVAFVRF; error occurred while reading VRF data.
X'08'	X'23'	X'00'	X'0A'	Error occurred while attempting VTOC recovery.
X'08'	X'25'	CVRET	CVSTAT	Invoke CVAFDIR; error occurred while reading format-4 DSCB without space maps.
X'08'	X'27'	SNS0, ECBSTAT	SNS1, ECBSTAT	I/O error occurred while reading format-4 DSCB.
X'08'	X'28'	SNS0, ECBSTAT	SNS1, ECBSTAT	I/O error occurred while writing format-4 DSCB with DIRF bit set.
X'08'	X'2E'	CVRET	CVSTAT	Invoke CVAFDIR; error occurred while reading format-1 DSCB (using key).
X'08'	X'30'	SNS0, ECBSTAT	SNS1, ECBSTAT	I/O error occurred while reading format-3 DSCB.
X'08'	X'30'	X'03'	X'04'	CCHH to RTA conversion routine encountered a zero (0) input value, this is invalid.
X'08'	X'30'	X'03'	X'18'	CCHH to RTA conversion routine encountered an END CCHH value that was less than the START CCHH value.
X'08'	X'30'	X'03'	X'1C'	CCHH to RTA conversion routine encountered a START 'CC' value that was not less than the maximum number of cylinders on the volume.
X'08'	X'30'	X'03'	X'20'	CCHH to RTA conversion routine encountered a END 'CC' value which was not less than the maximum number of cylinders on the volume.
X'08'	X'30'	X'03'	X'24'	CCHH to RTA conversion routine encountered a START 'HH' value which was not less than the maximum number of tracks per cylinder.
X'08'	X'30'	X'03'	X'28'	CCHH to RTA conversion routine encountered an END 'HH' value which was not less than the maximum number of tracks per cylinder.
X'08'	X'30'	X'04'	X'04'	RTA to CCHH conversion routine encountered a zero (0) input value, this is invalid.

Table 40. DADSM PARTREL Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Return Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'08'	X'30'	X'04'	X'14'	RTA to CCHH conversion routine encountered an input RTA which could not be converted into a 2-byte field. (RTA is too large).
X'08'	X'30'	X'04'	X'10'	RTA to CCHH conversion routine encountered an end RTA value which was to higher than the start RTA value.
X'08'	X'30'	X'04'	X'1C'	RTA to CCHH conversion routine converted a start 'CC' which was not smaller than the maximum number of cylinders on the volume.
X'08'	X'30'	X'04'	X'20'	RTA to CCHH conversion routine converted an end 'CC' which was not smaller than the maximum number of cylinders on the volume.
X'08'	X'31'	X'03'	X'04'	CCHH to RTA conversion routine encountered a zero (0) input value, this is invalid.
X'08'	X'31'	X'03'	X'18'	CCHH to RTA conversion routine encountered an END CCHH which was less than the START CCHH value.
X'08'	X'31'	X'03'	X'1C'	CCHH to RTA conversion routine encountered a START 'CC' value which was not less than the maximum number of cylinders on the volume.
X'08'	X'31'	X'03'	X'20'	CCHH to RTA conversion routine encountered a END 'CC' value which was not less than the maximum number of cylinders on the volume.
X'08'	X'31'	X'03'	X'24'	CCHH to RTA conversion routine encountered a START 'HH' value which was not less than the maximum number of tracks per cylinder.
X'08'	X'31'	X'03'	X'28'	CCHH to RTA conversion routine encountered an END 'HH' value which was not less than the maximum number of tracks per cylinder.
X'08'	X'31'	X'04'	X'04'	RTA to CCHH conversion routine encountered a zero (0) input value, this is invalid.
X'08'	X'31'	X'04'	X'14'	RTA to CCHH conversion routine encountered an input RTA which could not be converted into a 2-byte field. (RTA is too large).
X'08'	X'31'	X'04'	X'10'	RTA to CCHH conversion routine encountered an end RTA value that was higher than the start RTA value.
X'08'	X'31'	X'04'	X'1C'	RTA to CCHH conversion routine converted a start 'CC' that was not smaller than the maximum number of cylinders on the volume.
X'08'	X'31'	X'04'	X'20'	RTA to CCHH conversion routine converted an end 'CC' that was not smaller than the maximum number of cylinders on the volume.
X'08'	X'3B'	SNS0, ECBSTAT	SNS1, ECBSTAT	I/O error occurred while reading first format-3 DSCB.
X'08'	X'3D'	SNS0, ECBSTAT	SNS1, ECBSTAT	I/O error occurred while reading second through tenth format-3 DSCB.

Table 40. DADSM PARTREL Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Return Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'08'	X'3F'	X'03'	X'04'	CCHH to RTA conversion routine encountered a zero (0) input value, this is invalid.
X'08'	X'3F'	X'03'	X'18'	CCHH to RTA conversion routine encountered an END CCHH value that was less than the START CCHH value.
X'08'	X'3F'	X'03'	X'1C'	CCHH to RTA conversion routine encountered a START 'CC' value that was not less than the maximum number of cylinders on the volume.
X'08'	X'3F'	X'03'	X'20'	CCHH to RTA conversion routine encountered an END 'CC' value which was not less than the maximum number of cylinders on the volume.
X'08'	X'3F'	X'03'	X'24'	CCHH to RTA conversion routine encountered a START 'HH' value which was not less than the maximum number of tracks per cylinder.
X'08'	X'3F'	X'03'	X'28'	CCHH to RTA conversion routine encountered an END 'HH' value which was not less than the maximum number of tracks per cylinder.
X'08'	X'41'	X'06'	X'01'	X'Converting from 4-byte EDT table to a 2-byte EDT table. Invalid eye-catcher detected.'
X'08'	X'41'	X'06'	X'02'	X'Converting from 4-byte EDT table to a 2-byte EDT table. Invalid control block 'level.''
X'08'	X'41'	X'06'	X'03'	X'Converting from 4-byte EDT table to a 2-byte EDT table. 2-byte EDT is not large enough to contain all entries indicated in the 4-byte EDT.'
X'08'	X'41'	X'06'	X'04'	X'Converting from 4-byte EDT table to a 2-byte EDT table. Input RTA is too large to fit into a 2-byte field.'
X'08'	X'42'	SNS0, ECBSTAT	SNS1, ECBSTAT	I/O error occurred while reading a format-3 DSCB before releasing its space.
X'08'	X'43'	X'06'	X'01'	X'Converting from 4-byte EDT table to a 2-byte EDT table. Invalid eye-catcher detected.'
X'08'	X'43'	X'06'	X'02'	X'Converting from 4-byte EDT table to a 2-byte EDT table. Invalid control block 'level.''
X'08'	X'43'	X'06'	X'03'	X'Converting from 4-byte EDT table to a 2-byte EDT table. 2-byte EDT is not large enough to contain all entries indicated in the 4-byte EDT.'
X'08'	X'43'	X'06'	X'04'	X'Converting from 4-byte EDT table to a 2-byte EDT table. Input RTA is too large to fit into a 2-byte field.'
X'08'	X'44'	CVRET	CVSTAT	Invoke CVAFVRF; error occurred while recording VRF space used.
X'08'	X'45'	CVRET	CVSTAT	Invoke CVAFDSM; error occurred while updating VMDS with released format-3 DSCB.
X'08'	X'46'	SNS0, ECBSTAT	SNS1, ECBSTAT	I/O error occurred while writing updated format-3 DSCB.
X'08'	X'47'	SNS0, ECBSTAT	SNS1, ECBSTAT	I/O error occurred while zeroing released format-3 DSCB.

Table 40. DADSM PARTREL Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Return Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'08'	X'49'	CVRET	CVSTAT	Invoke CVAFVRF; error occurred while recording VRF space used.
X'08'	X'4A'	CVRET	CVSTAT	Invoke CVAFDSM; error occurred while releasing volume space from VPSM.
X'08'	X'4A'	CVRET	CVSTAT	Invoke CVAFDSM; error occurred while writing VPSM to release space.
X'08'	X'4B'	CVRET	CVSTAT	Invoke CVAFDIR; error occurred while writing updated format-1 DSCB.
X'08'	X'4E'	CVRET	CVSTAT	Invoke CVAFDIR; error occurred while reading format-4 DSCB and space maps.
X'08'	X'50'	X'00'	X'19'	Error occurred while attempting VTOC recovery.
X'08'	X'53'	SNS0, ECBSTAT	SNS1, ECBSTAT	I/O error occurred while reading format-4 DSCB.
X'08'	X'54'	SNS0, ECBSTAT	SNS1, ECBSTAT	I/O error occurred while writing format-4 DSCB to set DIRF bit.
X'08'	X'55'	CVRET	CVSTAT	Invoke CVAFDIR; error occurred while writing VIXM.
X'08'	X'57'	SNS0, ECBSTAT	SNS1, ECBSTAT	I/O error occurred while writing format-1 DSCB.
X'08'	X'58'	X'07'	X'01'	X'Sorting Extent Descriptor table. Eye-catcher is invalid.'
X'08'	X'58'	X'07'	X'02'	X'Sorting Extent Descriptor table. Invalid control block level number.'
X'08'	X'59'	SNS0, ECBSTAT	SNS1, ECBSTAT	I/O error occurred while reading format-5 DSCB.
X'08'	X'5B'	X'00'	X'32'	X'Attempting to convert an F5/F7 DSCB into an EDT. Input DSCB is NOT an F5 or F7.'
X'08'	X'5B'	X'0A'	X'01'	X'Attempting to convert an F7 DSCB into an EDT. Space allocated for the EDT is too small.'
X'08'	X'5B'	X'08'	X'01'	X'Attempting to convert an F5 DSCB into an EDT. Space allocated for the EDT is too small.'
X'08'	X'5D'	SNS0, ECBSTAT	SNS1, ECBSTAT	I/O error occurred while reading format-5 DSCB.
X'08'	X'5D'	X'00'	X'32'	X'Attempting to convert an F5/F7 DSCB into an EDT. Input DSCB is NOT an F5 or F7.'
X'08'	X'5D'	X'0A'	X'01'	X'Attempting to convert an F7 DSCB into an EDT. Space allocated for the EDT is too small.'
X'08'	X'5D'	X'08'	X'01'	X'Attempting to convert an F5 DSCB into an EDT. Space allocated for the EDT is too small.'
X'08'	X'5E'	X'00'	X'3C'	X'Attempting to convert an F5/F7 DSCB into an EDT. Input DSCB is NOT an F5 or F7.'
X'08'	X'5E'	X'0B'	X'01'	X'Attempting to convert an EDT into a F7 DSCB. EDT eye-catcher is invalid.'
X'08'	X'5E'	X'0B'	X'02'	X'Attempting to convert an EDT into a F7 DSCB. EDT control block 'level' is invalid.'

Table 40. DADSM PARTREL Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Return Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'08'	X'5E'	X'09'	X'01'	X'Attempting to convert an EDT into a F5 DSCB. EDT eye-catcher is invalid.'
X'08'	X'5E'	X'09'	X'02'	X'Attempting to convert an EDT into a F5 DSCB. EDT control block 'level' is invalid.'
X'08'	X'5E'	SNS0, ECBSTAT	SNS1, ECBSTAT	I/O error occurred while writing format-5 DSCB.
X'08'	X'60'	SNS0, ECBSTAT	SNS1, ECBSTAT	I/O error occurred while reading format-0 DSCB.
X'08'	X'62'	SNS0, ECBSTAT	SNS1, ECBSTAT	I/O error occurred while writing format-4 DSCB to clear the DIRF bit.
X'08'	X'63'	CVRET	CVSTAT	Invoke CVAFVRF; error occurred while clearing VRF data.
X'08'	X'64'	CVRET	CVSTAT	Invoke CVAFDIR; error occurred while releasing VTOC space maps.
X'08'	X'65'	CVRET	CVSTAT	Invoke CVAFDIR; error occurred while writing format-4 DSCB to disable index.
X'08'	X'6B'	MSSRET	X'00'	Error occurred while relinquishing MSS extents.
X'0C'	X'17'	X'00'	X'05'	PARTREL back-end parameter list contains no caller flag.
X'0C'	X'17'	X'00'	X'06'	PARTREL back-end parameter list contains more than one caller flag.
X'0C'	X'17'	X'00'	X'07'	PARTREL back-end parameter list contains invalid UCB pointer.
X'0C'	X'17'	X'00'	X'08'	PARTREL back-end parameter list contains invalid close work area pointer.
X'0C'	X'17'	X'00'	X'09'	PARTREL back-end parameter list contains invalid PARTREL work area pointer.
X'0C'	X'72'	UCBCAPTRC	UCBCAPTRS	IOSCAPU failed when attempting to capture the UCB passed by the caller.
X'0C'	X'73'	UCBCAPTRC	UCBCAPTRS	IOSCAPU failed when attempting to capture the UCB returned from CATALOG.
X'10'	X'17'	X'00'	X'02'	PARTREL back-end parameter list contains invalid eye-catcher.
X'10'	X'17'	X'00'	X'03'	PARTREL back-end parameter list contains invalid version.
X'10'	X'17'	X'00'	X'04'	PARTREL back-end parameter list contains invalid length.

Table 41. Abbreviation Descriptions for PARTREL Diagnostic Information (Register 0)

Abbreviation	Description
CDMRET	Return code from CDM.
CDMRSN	Return code from CDM.
CDMREAS	Return code from CDM.
CNVTRET	Return code from the CVTPCNVT conversion routine (TTR to CCHHR). For a description of the return code, see *z/OS DFSMSdfp Advanced Services*.
CVRET	Return code from the CVAF function. For a description of the return code, see *z/OS DFSMSdfp Advanced Services*.
CVSTAT	CVSTAT field of the CVAF parameter list. For a description of the fields, see “CVSTAT Field Codes” on page 40.
ECBCC	ECB completion code. For a description of the code, see *z/OS DFSMS: Using Data Sets*.
ENQRET	Return code from ENQ. For a description of the return code, see *z/OS MVS Programming: Assembler Services Reference ABE-HSP*.
ERASRET	Return code from the erase function. For a description of the return code, see *z/OS DFSMS Macro Instructions for Data Sets*.
PRERET	Return code from the IGGPRE00 preprocessing exit module. For a description of the return code, see *z/OS DFSMS Installation Exits*.
SNS0	First sense byte from the IOB. For a description of the sense byte, see *z/OS DFSMSdfp Advanced Services*.
SNS1	Second sense byte from the IOB. For a description of the sense byte, see *z/OS DFSMSdfp Advanced Services*.
SYSCDE1,2	System completion code (low order 2 bytes). For a description of the completion code, see *z/OS DFSMS Macro Instructions for Data Sets*.
UCBCAPTRC	Return code from IOSCAPU service.
UCBCAPTRS	Reason code from IOSCAPU service.

DADSM RENAME Return and Diagnostic Codes

When control returns to the instruction following the SVC 30 generated by the RENAME macro, register 15 contains the applicable DADSM SVC return codes, as Table 42 on page 83 shows. Register 0 contains 4 bytes of diagnostic information that consists of the DADSM error code, subfunction identifier, subfunction return code, and subfunction reason code, as Table 43 on page 84 shows. Each volume’s volume list entry contains the rename status code shown in Table 45 on page 90.

If an error occurs, DADSM issues message **IEC614I**, which contains the failure-related return and reason codes.

To determine whether the data set has been successfully renamed on each volume where it resides, you must examine the rename status code, that is, the last byte of each entry in the volume list.

- If an error results from a CVAF function, the subfunction return code field contains the CVAF return code, and the subfunction reason code field contains the CVAF status code (CVSTAT).
- If an error results from the execution of an EXCP channel program (DADSM uses EXCP channel programs to access DSCBs on OS VTOC volumes), the subfunction return code and reason code fields contain either:
  - The ECB completion code and the CSW channel status, if the ECB completion code is not X'41' and the channel status is not zero
  - Two sense bytes from the IOB, if the ECB completion code is X'41' and there is no channel status
- If an error results from a RACF invocation, the subfunction return code and reason code fields contain the RACF return code and reason code.

Table 42 describes the conditions indicated by the DADSM SVC return code.

Table 42. DADSM RENAME Return Codes

DADSM SVC Return Code	Description
00 (X'00')	Function successful.
04 (X'04')	No volume containing any part of data set was mounted, nor was there a unit available for mounting. The data set may be a VIO data set and cannot be renamed.
08 (X'08')	An unusual condition was encountered on one or more volumes.
12 (X'0C')	One of the following conditions occurred: • The DADSM RENAME parameter list is not valid. • The volume list is not valid. • At entry to RENAME, register 0 was not zero and did not point to a valid UCB. The address must be that of a UCB, not a UCB copy. The RENAME status code will not have been set.

Table 43 on page 84 describes the conditions that are indicated by the diagnostic information that is returned in register 0. The abbreviations used in Table 43 on page 84 are defined in Table 44 on page 89.

DADSM/CVAF Diagnostic Aids

Table 43. DADSM RENAME Diagnostic Information (Register 0)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'00'	X'0A'	X'00'	X'01'	Validate DADSM RENAME parameter list; the address of the DADSM parameter list is zero.
X'00'	X'0A'	X'00'	X'02'	Validate DADSM RENAME parameter list; either the address of the old or new data set names is zero or the address of the volume list is zero.
X'00'	X'0A'	X'00'	X'03'	Validate DADSM RENAME parameter list; the address of the volume list is invalid.
X'00'	X'0A'	X'00'	X'04'	Validate DADSM RENAME parameter list; the count of volumes in the volume list is less than or equal to 0.
X'00'	X'0A'	X'00'	X'21'	Validate DADSM RENAME parameter list; storage protection validation failed for the parameter list.
X'00'	X'0A'	X'00'	X'22'	Validate DADSM RENAME parameter list; storage protection validation failed for the old data set name.
X'00'	X'0A'	X'00'	X'23'	Validate DADSM RENAME parameter list; storage protection validation failed for the new data set name.
X'04'	X'06'	X'00'	X'1A'	Verify of SMS flags failed; FMT1 DSCB indicates that the data set is SMS-managed, but the FMT4 DSCB is not SMS-managed.
X'04'	X'06'	X'00'	X'1B'	Verify of SMS flags failed; caller indicates that the data set is SMS-managed, but the FMT1 DSCB does not.
X'04'	X'06'	X'00'	X'1C'	Verify of SMS flags failed; caller indicates no BCS entry exists for data set, but the FMT1 DSCB does not.
X'04'	X'06'	X'00'	X'1D'	Verify of SMS flags failed; caller indicates that the data set is cataloged, but the FMT1 DSCB does not.
X'04'	X'06'	X'00'	X'1E'	Verify of SMS flags failed; caller indicates that the volume is not SMS-managed, but the FMT4 DSCB indicates that it is either SMS or being converted to SMS.
X'04'	X'06'	X'00'	X'20'	Verify of SMS flags failed; caller indicates that the data set is not SMS-managed, but the FMT1 DSCB indicates that it is SMS.
X'04'	X'07'	BPRERET	X'00'	Invoke the IGGDARU2 preprocessing exit module; if the module rejects the request, the subfunction return code field contains the IGGDARU2 preprocessing exit return code.
X'04'	X'08'	VDSS1	VDSS2	VDSS detected an error.
X'04'	X'0B'	X'00'	X'24'	Validate DADSM RENAME request; rename not supported for a VIO data set.

Table 43. DADSM RENAME Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'04'	X'0B'	X'00'	X'27'	Validate DADSM RENAME request; blank volume serial number and format of the data set name indicate a possible VIO data set, but the data set was not found on the DSAB chain.
X'04'	X'0B'	X'00'	X'2E'	Validate DADSM RENAME request; an open DEB was found for the data set being renamed.
X'04'	X'0B'	X'00'	X'30'	Validate DADSM RENAME request; cannot rename a VTOC index data set to a non-VTOC index type of name on an indexed VTOC volume.
X'04'	X'0B'	X'00'	X'31'	Validate DADSM RENAME request; cannot rename a non-VTOC index type of name to a VTOC index data set on an indexed VTOC volume.
X'04'	X'0B'	ENQRET	X'25'	Validate DADSM RENAME request; enqueue on TIOT failed.
X'04'	X'0B'	ENQRET	X'26'	Validate DADSM RENAME request; enqueue on SYSDSN failed.
X'04'	X'0B'	ENQRET	X'46'	Validate DADSM RENAME request; enqueue on SYSDSN failed. Also, the caller has appropriate RACF READ authority to be able to specify that the data set being renamed is not the data set in use. See *z/OS DFSMSdfp Advanced Services*.
X'04'	X'0C'	X'00'	X'28'	Locate volume to be processed; the volume serial number from the volume list is zero.
X'04'	X'0C'	X'00'	X'29'	Locate volume to be processed; volume not mounted on any online, shareable DASD unit.
X'04'	X'0C'	X'00'	X'2A'	Locate volume to be processed; device type of volume from the volume list does not match the UCB device type of unit on which the volume resides.
X'04'	X'0D'	X'00'	X'2B'	Invoke IGG0300F to demount/mount/verify volume; volume could not be demounted.
X'04'	X'0D'	X'00'	X'2C'	Invoke IGG0300F to demount/mount/verify volume; volume could not be mounted.
X'04'	X'0E'	X'00'	X'2F'	Check caller's authority and rename DSCB; SECLOADA installation exit rejected the request.
X'04'	X'0E'	X'00'	X'32'	Check caller's authority and rename DSCB; caller not functionally authorized to rename the VTOC index data set.
X'04'	X'0E'	X'00'	X'33'	Check caller's authority and rename DSCB; data set with new name already exists on indexed VTOC volume.
X'04'	X'0E'	X'00'	X'34'	Check caller's authority and rename DSCB; data set with new name already exists on OS VTOC volume.
X'04'	X'0E'	X'00'	X'35'	Check caller's authority and rename DSCB; attempt to rename a data set to the VTOC index name failed, because a data set whose name begins with 'SYS1.VTOCIX.' already exists on the volume.

Table 43. DADSM RENAME Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'04'	X'0E'	X'00'	X'36'	Check caller's authority and rename DSCB; the new data set name is indexed VTOC high-water-mark key (44 bytes of X'FF').
X'04'	X'0E'	X'00'	X'37'	Check caller's authority and rename DSCB; indexed VTOC—old data set name in index but no FMT1 DSCB was found.
X'04'	X'0E'	X'00'	X'38'	Check caller's authority and rename DSCB; OS VTOC—old data set was not found on the volume.
X'04'	X'0F'	RACRET	RACREAS	Invoke RACF to verify that the caller is authorized to the volume for renaming the VTOC index data set. RACROUTE = AUTH.
X'04'	X'10'	RACRET	RACREAS	Invoke RACF to check caller's authorization and rename the data set name in the profile. RACROUTE = DEFINE.
X'04'	X'11'	RACRET	RACREAS	Invoke RACF to rename the data set name in the profile. RACROUTE = DEFINE.
X'04'	X'12'	CVRET	CVSTAT	Invoke CVAFDSM; add new data set name to the index and check for a duplicate name.
X'04'	X'13'	CVRET	CVSTAT	Invoke CVAFDIR; read FMT1 DSCB of the old data set name.
X'08'	X'03'	X'00'	X'05'	Logical error on SSI call; bad SSOB format, length, and so forth.
X'08'	X'08'	VDSS1	VDSS2	VDSS detected an error.
X'08'	X'09'	SYSCDE1	SYSCDE2	ESTAE routine invoked by RTM—redundant.
X'08'	X'0D'	X'00'	X'2D'	Invoke IGG0300F to demount/mount/verify volume; I/O error occurred while reading the volume label.
X'08'	X'0E'	X'00'	X'39'	Check caller's authority and rename FMT1 DSCB; I/O error during VRF recovery routine execution.
X'08'	X'0E'	X'00'	X'3A'	Check caller's authority and rename DSCB; indexed VTOC—data set was found but its FMT1 DSCB ID is invalid.
X'08'	X'0E'	X'00'	X'3B'	Check caller's authority and rename DSCB; OS VTOC–data set was found but the FMT1 DSCB ID is invalid.
X'08'	X'0E'	PRERET	X'3C'	Check caller's authority and rename the DSCB; the IGGPRE00 preprocessing exit rejected the request; the subfunction return code field contains the IGGPRE00 return code.
X'08'	X'12'	CVRET	CVSTAT	Invoke CVAFDSM; add the new data set name to the index and check for a duplicate name.
X'08'	X'13'	CVRET	CVSTAT	Invoke CVAFDIR; read the FMT1 DSCB of the old data set name.
X'08'	X'14'	CVRET	CVSTAT	Invoke CVAFDIR; read the space maps and the first high level VIER.
X'08'	X'15'	CVRET	CVSTAT	Invoke CVAFVRF; check for VRF data in the VIXM.

Table 43. DADSM RENAME Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'08'	X'16'	CVRET	CVSTAT	Invoke CVAFDIR; read the FMT1 DSCB of the new data set name to check for a duplicate data set name.
X'08'	X'17'	ECBCC, SNS0	CHSTAT, SNS1	Read the FMT1 DSCB of the new data set name and check for a duplicate data set name.
X'08'	X'18'	ECBCC, SNS0	CHSTAT, SNS1	Read the FMT1 DSCB of the old data set name.
X'08'	X'19'	CVRET	CVSTAT	Invoke CVAFVRF; write VRF data to the index map.
X'08'	X'1A'	CVRET	CVSTAT	Invoke CVAFDIR; write VRF data to the volume.
X'08'	X'1B'	CVRET	CVSTAT	Invoke CVAFDIR; write the updated data set name to the volume.
X'08'	X'1C'	CVRET	CVSTAT	Invoke CVAFDIR; write the updated FMT1 DSCB to the volume.
X'08'	X'1D'	CVRET	CVSTAT	Invoke CVAFDSM; delete the old data set name from the index.
X'08'	X'1E'	CVRET	CVSTAT	Invoke CVAFVRF; zero out the VRF data.
X'08'	X'1F'	CVRET	CVSTAT	Invoke CVAFDIR; write zero VRF data to the volume.
X'08'	X'20'	ECBCC, SNS0	CHSTAT, SNS1	Write the FMT1 DSCB with the new name.
X'0C'	X'01'	X'00'	X'03'	Validation of DADSM RENAME parameter list; the volume list address range is invalid.
X'0C'	X'01'	X'00'	X'04'	Validation of DADSM RENAME parameter list; the count of volumes in the volume list is less than or equal to 0.
X'0C'	X'02'	GETMRET	X'00'	GETMAIN for DAVLL failed.
X'0C'	X'04'	X'00'	X'09'	Validation of branch entry parameter list failed; DARSKRSV must be zero.
X'0C'	X'04'	X'00'	X'0A'	Validation of branch entry parameter list failed; DARANDSN is zero.
X'0C'	X'04'	X'00'	X'0B'	Validation of branch entry parameter list failed; DARAVOLL is zero.
X'0C'	X'04'	X'00'	X'0C'	Validation of branch entry parameter list failed; DARADSN is zero.
X'0C'	X'04'	X'00'	X'0D'	Validation of branch entry parameter list failed; DARRSRV1 must be zero.
X'0C'	X'04'	X'00'	X'0E'	Validation of branch entry parameter list failed; DARRSRV2 must be zero.
X'0C'	X'04'	X'00'	X'0F'	Validation of branch entry parameter list failed; DARRSRV3 must be zero.
X'0C'	X'04'	X'00'	X'10'	Validation of branch entry parameter list failed; DARRSRV4 must be zero.
X'0C'	X'04'	X'00'	X'11'	Validation of branch entry parameter list failed; DARRSRV5 must be zero.

Table 43. DADSM RENAME Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'0C'	X'04'	X'00'	X'12'	Validation of branch entry parameter list failed; DARRPKEY field is not valid for updating fields in caller's DAREN.
X'0C'	X'04'	X'00'	X'3D'	Validation of branch entry parameter list failed; DARRSRV6 must be zero.
X'0C'	X'04'	X'00'	X'3F'	Validation of branch entry parameter list failed; UCBID of the passed UCB is not valid.
X'0C'	X'04'	X'00'	X'40'	Validation of branch entry parameter list failed.
X'0C'	X'05'	X'00'	X'04'	Verification of volume list (DAVLL) failed; count of volumes in the volume list is zero.
X'0C'	X'05'	X'00'	X'13'	Validation of volume list (DAVLL) failed; invalid DAVLLID field.
X'0C'	X'05'	X'00'	X'14'	Validation of volume list (DAVLL) failed; invalid DAVLVER field.
X'0C'	X'05'	X'00'	X'15'	Validation of volume list (DAVLL) failed; invalid DAVLLEN field.
X'0C'	X'05'	X'00'	X'16'	Validation of volume list (DAVLL) failed; DAVSKRSV must be zero.
X'0C'	X'05'	X'00'	X'17'	Validation of volume list (DAVLL) failed; DAVRSRV1 must be zero.
X'0C'	X'05'	X'00'	X'18'	Validation of volume list (DAVLL) failed; DAVRSRV2 must be zero.
X'0C'	X'05'	X'00'	X'19'	Validation of volume list (DAVLL) failed; DAVLKEY value is invalid for updating fields in caller's DAVLL.
X'0C'	X'05'	X'00'	X'40'	Validation of volume list (DAVLL) failed; device type in the volume list entry or in passed UCB indicates that the device is not DASD.
X'0C'	X'21'	UCBCAPTRC	UCBCAPTRS	IOSCAPU failed when attempting to capture the UCB passed by the caller
X'10'	X'01'	X'00'	X'01'	Validation of DADSM RENAME parameter list failed; address of the parameter list is zero.
X'10'	X'01'	X'00'	X'02'	Validation of DADSM RENAME parameter list failed; address of the old or new data set names is zero or the address of the volume list is zero.
X'10'	X'04'	X'00'	X'04'	Validation of branch entry parameter list failed; count of volumes in the volume list is zero.
X'10'	X'04'	X'00'	X'06'	Validation of branch entry parameter list failed; invalid DARPLID field.
X'10'	X'04'	X'00'	X'07'	Validation of branch entry parameter list failed; invalid DARPVER field.
X'10'	X'04'	X'00'	X'08'	Validation of branch entry parameter list failed; invalid DARPLEN field.

Table 44 on page 89 describes abbreviations for DADSM RENAME diagnostic information.

Table 44. Abbreviation Descriptions for DADSM RENAME Diagnostic Information (Register 0)

Abbreviation	Description
BPRERET	Return code from the IGGDARU2 RENAME preprocessing exit module. For a description of the return code, see *z/OS DFSMS Installation Exits*.
CHSTAT	Channel status from I/O. For a description of the status, see *z/OS DFSMSdfp Advanced Services*.
CVRET	Return code from CVAF function. For a description of the return code, see *z/OS DFSMSdfp Advanced Services*.
CVSTAT	CVSTAT field of CVAF parameter list. For a description of the field, see “CVSTAT Field Codes” on page 40.
ECBCC	ECB completion code. For a description of the completion code, see *z/OS DFSMSdfp Advanced Services*.
ENQRET	Return code from ENQ. For a description of the return code, see *z/OS MVS Programming: Assembler Services Reference ABE-HSP*.
GETMRET	Return code from GETMAIN.
PRERET	Return code from the IGGPRE00 preprocessing exit module. For a description of the return code, see *z/OS DFSMS Installation Exits*.
RACREAS	RACF reason code. See Diagnostic Code for request type.
RACRET	RACF return code. See Diagnostic Code for request type.
SNS0	First sense byte from the IOB. For a description of the sense byte, see *z/OS DFSMSdfp Advanced Services*.
SNS1	Second sense byte from the IOB. For a description of the sense byte, see the appropriate hardware manual.
SYSCDE1,2	System completion code (2 bytes). For a description of the completion code, see *z/OS DFSMS Macro Instructions for Data Sets*.
UCBCAPTRC	Return code from IOSCAPU service.
UCBCAPTRS	Reason code from IOSCAPU service.
VDSS1,2	VDSS error reason code (2 bytes). For a description of the VDSS1 and VDSS2 field values, convert the reason code to decimal and see “VTOC/Data Set Services (VDSS) Reason Codes” on page 455.

Status Codes from RENAME

After the RENAME macro instruction is executed, the last byte of each 12-byte entry in the volume list indicates one of the that are described in Table 45.

Table 45. Status Codes from RENAME Macro Instruction

Rename Status Code	Description
0 (X'00')	The format-1 DSCB for the data set has been renamed in the VTOC on this volume.
1 (X'01')	The VTOC of this volume does not contain the format-1 DSCB of the data set to be renamed.
2 (X'02')	One of the following conditions occurred: • The data set could not be renamed because the data set was password protected and the password was not supplied in the two attempts allowed. • An attempt was made to rename a VSAM data space or a catalogued VSAM data set. • An attempt was made to rename a VTOC index data set. • An SMS validation failure occurred.
3 (X'03')	A format-1 DSCB containing the new data set name already exists in the VTOC of this volume, or an attempt was made to rename a data set to a name starting with 'SYS1.VTOCIX'
4 (X'04')	One of the following conditions occurred: • A permanent I/O error occurred while trying to rename the data set on this volume. • An incorrect format-1 DSCB was encountered while processing this volume. • No space is available in the index VIER for the new name, and no additional VIERs are available.
5 (X'05')	It could not be verified that this volume was mounted nor was a unit available for mounting the volume.
6 (X'06')	The operator was unable to mount this volume.
7 (X'07')	The data set was not renamed, because it was currently open for processing.
8 (X'08')	The data set is defined to RACF, but either you are not authorized to access the data set or the data set is defined to RACF on multiple volumes.

DADSM SCRATCH Return and Diagnostic Codes

Control returns to the instruction following the SVC 29 generated by the SCRATCH macro. Register 15 contains the applicable DADSM SVC return code as shown in Table 46. Register 0 contains 4 bytes of diagnostic information consisting of the DADSM error code, subfunction identifier, subfunction return code, and subfunction reason code as shown in Table 47 on page 92. Each volume list entry contains the scratch status code as shown in Table 49 on page 100.

If an error occurs, DADSM might issue message **IEC614I**, which would contain the failure-related return and reason codes.

To determine whether the data set has been successfully deleted from each volume on which it resides, you must examine the scratch status code, that is, the last byte of each entry in the volume list.

If an error results from a CVAF function, the subfunction return code field contains the CVAF return code, and the subfunction reason code field contains the CVAF status code (CVSTAT).

If an error results from the execution of an EXCP channel program (DADSM sometimes uses EXCP channel programs to access DSCBs), the subfunction return code and reason code fields have either of the following contents:

- The ECB completion code and the CSW channel status (if the ECB completion code is not X'41' and the channel status is not zero)
- Two sense bytes from the IOB (if the ECB completion code is X'41' and there is no channel status)

If an error results from a RACF invocation, the subfunction return code and reason code fields contain the RACF return code and reason code.

Table 46 describes the conditions that are indicated by the DADSM SVC return code.

Table 46. DADSM SCRATCH Return Codes (R15)

DADSM SVC Return Code	Description
00 (X'00')	Function successful.
04 (X'04')	No volume containing any part of the data set was mounted, nor was there a unit available for mounting. The data set might be a VIO data set that was not allocated to your job step.
08 (X'08')	An unusual condition was encountered on one or more volumes.
12 (X'0C')	One of the following conditions occurred: • The DADSM SCRATCH parameter list is invalid. • The volume list is invalid. • At entry to SCRATCH, register 0 was not zero or did not point to a valid UCB. The address must be that of a UCB, not a UCB copy. The SCRATCH status code will not have been set.

DADSM/CVAF Diagnostic Aids

Table 47 describes the conditions that are indicated by the diagnostic information that is returned in register 0. The abbreviations that are used in Table 47 are defined in Table 48 on page 99.

Table 47. DADSM SCRATCH Diagnostic Information (Register 0)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'00'	X'3B'	X'00'	X'01'	Verify DADSM SCRATCH parameter list; address of the parameter list is zero.
X'00'	X'3B'	X'00'	X'02'	Verify DADSM SCRATCH parameter list; address of the data set name or address of the volume list is zero.
X'00'	X'3B'	X'00'	X'03'	Verify DADSM SCRATCH parameter list; address of the volume list is invalid.
X'00'	X'3B'	X'00'	X'04'	Verify DADSM SCRATCH parameter list; count of volumes in the volume list is less than or equal to zero.
X'00'	X'3B'	X'00'	X'45'	Verification of DADSM SCRATCH parameter list; storage protection validation failed for parameter list.
X'00'	X'3B'	X'00'	X'46'	Verification of DADSM SCRATCH parameter list; storage protection validation failed for data set name.
X'04'	X'06'	X'00'	X'1C'	Verification of SMS flags failed; FMT1 DSCB indicates data set is SMS-managed, but FMT4 DSCB indicates that volume is not SMS-managed.
X'04'	X'06'	X'00'	X'1D'	Verification of SMS flags failed; caller indicates data set is SMS-managed, but FMT1 DSCB indicates non-managed.
X'04'	X'06'	X'00'	X'1E'	Verification of SMS flags failed; caller indicates the data set is not cataloged, but the FMT1 DSCB indicates the data set is cataloged.
X'04'	X'06'	X'00'	X'1F'	Verification of SMS flags failed; caller indicates data set is cataloged, but FMT1 DSCB indicates the data set is not cataloged.
X'04'	X'06'	X'00'	X'20'	Verification of SMS flags failed; caller indicates data set is not SMS-managed, but FMT4 DSCB indicates that it is an SMS volume. This error might be caused by a damaged catalog entry. Issuing the recatalog command might clear this error.
X'04'	X'06'	X'00'	X'23'	Verification of SMS flags failed; caller indicated that the data set is not SMS managed, but FMT1 DSCB indicates that the data set is SMS managed.
X'04'	X'07'	BPRERET	X'00'	Invoke the IGGDASU2 preprocessing exit module; if the module rejects the request, the subfunction return code field contains the IGGDASU2 preprocessing exit return code.
X'04'	X'08'	X'00'	X'22'	Verification of *last referenced date* failed; date passed by caller does not match date in FMT1 DSCB.
X'04'	X'09'	VDSS1	VDSS2	VDSS detected an error.

Table 47. DADSM SCRATCH Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'04'	X'0B'	X'00'	X'27'	Verify DADSM SCRATCH request; DSAB open count (DSABOPCT) for VIO data set is not zero.
X'04'	X'0B'	X'00'	X'28'	Verify DADSM SCRATCH request; UCB data management count (UCBDMCT) for VIO data set is not zero.
X'04'	X'0B'	X'00'	X'29'	Verify DADSM SCRATCH request; volume list contains more than one volume for VIO data set.
X'04'	X'0B'	X'00'	X'2A'	Verify DADSM SCRATCH request; input UCB address is zero.
X'04'	X'0B'	X'00'	X'4B'	Verify DADSM SCRATCH request; data set is unexpired.
X'04'	X'0B'	ENQRET	X'24'	Verify DADSM SCRATCH request; enqueue on TIOT failed.
X'04'	X'0B'	ENQRET	X'25'	Verify DADSM SCRATCH request; enqueue on SYSDSN failed.
X'04'	X'12'	RACRET	RACREAS	Invoke RACF to verify that the caller is authorized to the volume for scratching the VTOC index data set.
X'04'	X'13'	RACRET	RACREAS	Invoke RACF to verify that the caller is authorized to the volume, after RACHECK of the data set with generic profile failed.
X'04'	X'18'	X'00'	X'33'	Read DSCBs and check caller's authority; indexed VTOC—data set in index, but no FMT1 DSCB found.
X'04'	X'18'	X'00'	X'34'	Read DSCBs and check caller's authority. Module SECLOADA failed the request. Possible reasons are that the data set is password protected and a password was not provided, or for a VSAM data set, the catalog might have structure problems that prevented authorization verification. An IDCAMS DIAGNOSE job might provide information about the catalog.
X'04'	X'18'	X'00'	X'35'	Read DSCBs and check caller's authority; caller not functionally authorized to scratch VTOC index data set.
X'04'	X'18'	X'00'	X'36'	Read DSCBs and check caller's authority; cannot scratch VTOC index data set on an indexed VTOC volume.
X'04'	X'18'	X'00'	X'40'	Read DSCBs and check caller's authority; an open DEB was found for the data set being scratched.
X'04'	X'18'	X'00'	X'44'	Read DSCBs and check caller's authority; OS VTOC—FMT1 DSCB not found on volume.
X'04'	X'18'	X'00'	X'4C'	The data set is a VSAM data set.
X'04'	X'19'	CVRET	CVSTAT	Invoke CVAFDSM; delete data set name from index.
X'04'	X'27'	X'00'	X'41'	Invoke IGG0290F to demount/mount/verify volume; volume could not be demounted.

Table 47. DADSM SCRATCH Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'04'	X'27'	X'00'	X'42'	Invoke IGG0290F to demount/mount/verify volume; volume could not be mounted.
X'04'	X'28'	X'00'	X'38'	Locate volume to be processed; volume serial number from volume list is zero.
X'04'	X'28'	X'00'	X'39'	Locate volume to be processed; volume not mounted on any online, shareable DASD unit.
X'04'	X'28'	X'00'	X'3A'	Locate volume to be processed; device type of volume from volume list does not match UCB device type of unit on which volume resides.
X'04'	X'28'	X'00'	X'3B'	Locate volume to be processed; device on which current volume is mounted is not ready.
X'04'	X'29'	X'00'	X'3C'	Remove DSCBs; FMT1 indicates data set has more than 3 extents, but FMT3 CCHHR address in FMT1 is zero.
X'04'	X'29'	PRERET	X'3D'	Remove DSCBs; the IGGPRE00 preprocessing exit rejected the request.
X'04'	X'36'	RACRET	RACREAS	Invoke RACF to verify that the caller is authorized to the data set. RACROUTE Request = AUTH.
X'04'	X'37'	RACRET	RACREAS	Invoke RACF to verify that the caller is authorized to the data set and to return erase status. RACROUTE Request = AUTH.
X'04'	X'38'	RACRET	RACREAS	Invoke RACF to verify that the caller is authorized to the volume after RACHECK of data set with discrete profile failed. RACROUTE Request = AUTH.
X'04'	X'3D'	CDMRSN1	CDMRSN2	Invoke CDM to delete a PDSE directory; directory could not be deleted. For explanation see “CDM Reason Codes” on page 408.
X'04'	X'3E'	COMMRC	COMMRS	System error. Unable to convert Extent Descriptor Table prior to calling IGGPRE00.
X'04'	X'3F'	COMMRC	COMMRS	System error. Error converting CCHH to relative track address format.
X'04'	X'40'	COMMRC	COMMRS	System error. Error sorting the Extent Descriptor Table entries.
X'04'	X'41'	COMMRC	COMMRS	System error. Error converting Extent Descriptor Table entries to Format 5 DSCB format entries.
X'04'	X'42'	COMMRC	COMMRS	System error. Error converting Extent Descriptor Table entries to Format 7 DSCB format entries.
X'04'	X'43'	COMMRC	COMMRS	System error. Error converting Format 5 DSCB entries to Extent Descriptor Table format entries.
X'04'	X'44'	COMMRC	COMMRS	System error. Error converting Format 7 DSCB entries to Extent Descriptor Table format entries.
X'08'	X'03'	X'00'	X'05'	Logical error on SSI call; bad SSOB format, length, and so forth.
X'08'	X'09'	X'00'	X'00'	VDSS detected an error.

Table 47. DADSM SCRATCH Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'08'	X'0A'	SYSCDE1	SYSCDE2	ESTAE routine invoked by RTM; system completion code is passed back in register 0.
X'08'	X'0B'	X'00'	X'2B'	Validate DADSM SCRATCH request; VSCRATCH failed for VIO data set.
X'08'	X'0B'	PRERET	X'2C'	Verify DADSM SCRATCH request; the IGGPRE00 preprocessing exit rejected a request for a VIO data set.
X'08'	X'0C'	X'00'	X'2D'	Update FMT5 DSCBs for data set with more than 16 extents; CCHH of record to be read/written is less than the beginning CCHH of VTOC.
X'08'	X'0C'	X'00'	X'2E'	Update FMT5 DSCBs for data set with more than 16 extents; CCHH of record to be read/written is greater than the ending CCHH of VTOC.
X'08'	X'0C'	X'00'	X'2F'	Update FMT5 DSCBs for data set with more than 16 extents; count of number of extents indicates new FMT5 is already full.
X'08'	X'0C'	X'00'	X'3F'	Update FMT5 DSCBs for data set with more than 16 extents; number of FMT0 DSCBs is less than the number needed to describe extents being freed.
X'08'	X'0D'	CVRET	CVSTAT	Invoke CVAFDIR; read FMT5 DSCB.
X'08'	X'0E'	CVRET	CVSTAT	Invoke CVAFDIR; write FMT5 DSCB.
X'08'	X'0F'	CVRET	CVSTAT	Invoke CVAFDIR; write FMT0 DSCB.
X'08'	X'10'	CVRET	CVSTAT	Invoke CVAFDIR; read FMT0 DSCB.
X'08'	X'11'	X'00'	X'30'	Erase tracks occupied by data set; PGFIX of I/O work area failed.
X'08'	X'14'	ECBCC, SNS0	CHSTAT, SNS1	Read FMT4 and FMT1 DSCBs.
X'08'	X'15'	CVRET	CVSTAT	Invoke CVAFDIR; read index maps, high level VIER, and FMT4 DSCB.
X'08'	X'16'	CVRET	CVSTAT	Invoke CVAFVRF; check for VRF data in the VIXM.
X'08'	X'17'	ECBCC, SNS0	CHSTAT, SNS1	Channel program issued to read FMT4 and FMT1 DSCBs after IGGVRF00 disabled the VTOC index.
X'08'	X'18'	X'00'	X'31'	Read DSCBs and check callers authority; I/O error during VRF recovery routine execution.
X'08'	X'18'	X'00'	X'32'	Read DSCBs and check caller's authority; OS VTOC—data set found, but FMT1 DSCB ID is invalid.
X'08'	X'19'	CVRET	CVSTAT	Invoke CVAFDSM; delete data set name from index.
X'08'	X'1A'	CVRET	CVSTAT	Invoke CVAFDIR; read FMT1 DSCB.
X'08'	X'1B'	CVRET	CVSTAT	Invoke CVAFDIR; read FMT4 DSCB.
X'08'	X'1C'	CVRET	CVSTAT	Invoke CVAFDIR; write FMT4 DSCB.
X'08'	X'1D'	CVRET	CVSTAT	Invoke CVAFDIR; release buffers.
X'08'	X'1E'	ECBCC, SNS0	CHSTAT, SNS1	Read FMT6 DSCB.
X'08'	X'1F'	ECBCC, SNS0	CHSTAT, SNS1	Write FMT6 DSCB.

Table 47. DADSM SCRATCH Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'08'	X'20'	ECBCC, SNS0	CHSTAT, SNS1	Write FMT0 DSCB.
X'08'	X'21'	ECBCC, SNS0	CHSTAT, SNS1	Read first FMT5 DSCB.
X'08'	X'22'	X'00'	X'37'	Update FMT6 DSCB; first split cylinder extent belonging to data set is less than the lowest CCHH in first FMT6 DSCB.
X'08'	X'23'	ECBCC, SNS0	CHSTAT, SNS1	Read FMT5 DSCB.
X'08'	X'24'	ECBCC, SNS0	CHSTAT, SNS1	Write FMT5 DSCB.
X'08'	X'25'	ECBCC, SNS0	CHSTAT, SNS1	Read FMT0 DSCB.
X'08'	X'26'	ECBCC, SNS0	CHSTAT, SNS1	Write FMT4 DSCB.
X'08'	X'27'	X'00'	X'43'	Invoke IGG0290F to demount/mount/verify volume; I/O error occurred while reading the volume label.
X'08'	X'29'	X'00'	X'3E'	Remove DSCBs; DSCB pointed to by the FMT1 is neither a FMT2 nor FMT3 DSCB.
X'08'	X'2A'	CVRET	CVSTAT	Invoke CVAFDIR; read FMT2/FMT3 DSCBs.
X'08'	X'2B'	CVRET	CVSTAT	Invoke CVAFVRF; write VRF data to the index map.
X'08'	X'2C'	CVRET	CVSTAT	Invoke CVAFDIR; write buffer lists to the volume.
X'08'	X'2D'	CVRET	CVSTAT	Invoke CVAFDIR; zero out all FMT1, FMT2, and FMT3 DSCBs.
X'08'	X'2E'	CVRET	CVSTAT	Invoke CVAFDSM; release volume space to the VPSM.
X'08'	X'2F'	CVRET	CVSTAT	Invoke CVAFVRF; zero out VRF data.
X'08'	X'30'	CVRET	CVSTAT	Invoke CVAFDIR; write index maps and VIXM to the volume.
X'08'	X'31'	CVRET	CVSTAT	Invoke CVAFDIR; read FMT5 and FMT6 DSCBs.
X'08'	X'32'	CVRET	CVSTAT	Invoke CVAFDIR; write FMT4 DSCB to disable indexed VTOC.
X'08'	X'33'	CVRET	CVSTAT	Invoke CVAFDIR; write FMT4 DSCB with DS4DOSBT bit 'on' for OS VTOC.
X'08'	X'34'	CVRET	CVSTAT	Invoke CVAFDIR; write FMT4 DSCB for OS VTOC, after the request was rejected by the IGGDASU2 preprocessing exit.
X'08'	X'35'	CVRET	CVSTAT	Invoke CVAFDIR; write FMT4 DSCB if more than five tracks are being erased.
X'08'	X'39'	ECBCC, SNS0	CHSTAT, SNS1	Channel program issued to read FMT4 and FMT1 DSCBs after VTOC index was disabled.
X'08'	X'3A'	CVRET	CVSTAT	Invoke CVAFDIR; write FMT4 DSCB for OS VTOC, after erase failure.
X'08'	X'3C'	CVRET	CVSTAT	Invoke CVAFDIR; rewrite FMT4 DSCB with DS4DOSBT bit 'on'; data set with incorrect number of extents was found.
X'08'	X'47'	ECBCC	CHSTAT	Erase failed.
X'0C'	X'01'	X'00'	X'03'	Verify DADSM SCRATCH parameter list; address of volume list is invalid.

Table 47. DADSM SCRATCH Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'0C'	X'01'	X'00'	X'04'	Verify DADSM SCRATCH parameter list; count of volumes in the volume list is less than or equal to zero.
X'0C'	X'02'	GETMRET	X'00'	GETMAIN for DAVLL failed.
X'0C'	X'04'	X'00'	X'09'	Verification of branch entry parameter list failed; DASSKRSV must be zero.
X'0C'	X'04'	X'00'	X'0A'	Verification of branch entry parameter list failed; flag DASVRFRD is 'on', but DASREFDT is zero.
X'0C'	X'04'	X'00'	X'0B'	Verification of branch entry parameter list failed; DASAVOLL is zero.
X'0C'	X'04'	X'00'	X'0C'	Verification of branch entry parameter list failed; DASADSN is zero.
X'0C'	X'04'	X'00'	X'0D'	Verification of branch entry parameter list failed; DASRSRV1 must be zero.
X'0C'	X'04'	X'00'	X'0E'	Verification of branch entry parameter list failed; DASRSRV2 must be zero.
X'0C'	X'04'	X'00'	X'0F'	Verification of branch entry parameter list failed; DASRSRV3 must be zero.
X'0C'	X'04'	X'00'	X'10'	Verification of branch entry parameter list failed; DASRSRV4 must be zero.
X'0C'	X'04'	X'00'	X'11'	Verification of branch entry parameter list failed; DASRSRV5 must be zero.
X'0C'	X'04'	X'00'	X'12'	Verification of branch entry parameter list failed; DASRSRV6 must be zero.
X'0C'	X'04'	X'00'	X'13'	Verification of branch entry parameter list failed; DASRSRV7 must be zero.
X'0C'	X'04'	X'00'	X'14'	Verification of branch entry parameter list failed; DASPKEY value invalid for updating fields in caller's DASCR.
X'0C'	X'04'	X'00'	X'47'	Verification of branch entry parameter list failed; DASRSRV8 must be zero.
X'0C'	X'04'	X'00'	X'49'	Verification of branch entry parameter list failed; UCBID of passed UCB is invalid.
X'0C'	X'04'	X'00'	X'4A'	Verification of branch entry parameter list failed; device type in volume list entry or in passed UCB indicates that the device is not DASD.
X'0C'	X'05'	X'00'	X'15'	Verification of volume list (DAVLL) failed; invalid DAVLLID field.
X'0C'	X'05'	X'00'	X'16'	Verification of volume list (DAVLL) failed; invalid DAVLVER field.
X'0C'	X'05'	X'00'	X'17'	Verification of volume list (DAVLL) failed; invalid DAVLLEN field.
X'0C'	X'05'	X'00'	X'18'	Verification of volume list (DAVLL) failed; DAVSKRSV must be zero.

Table 47. DADSM SCRATCH Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'0C'	X'05'	X'00'	X'19'	Verification of volume list (DAVLL) failed; DAVRSRV1 must be zero.
X'0C'	X'05'	X'00'	X'1A'	Verification of volume list (DAVLL) failed; DAVRSRV2 must be zero.
X'0C'	X'05'	X'00'	X'1B'	Verification of volume list (DAVLL) failed; DAVLKEY value invalid for updating fields in caller's DAVLL.
X'0C'	X'05'	X'00'	X'4A'	Verification of volume list (DAVLL) failed; device type in volume list entry or in passed UCB indicates that the device is not DASD.
X'0C'	X'46'	UCBCAPTRC	UCBCAPTRS	IOSCAPU failed when attempting to capture the UCB passed by the caller.
X'10'	X'01'	X'00'	X'01'	Verification of DADSM SCRATCH parameter list failed; address of parameter list is zero.
X'10'	X'01'	X'00'	X'02'	Verification of branch entry parameter list failed; address of the data set name or address of the volume list is zero.
X'10'	X'01'	X'00'	X'48'	Verification of UCB passed in register. Register 0 was not zero or did not point to valid UCB. The address must be that of a UCB, not a UCB copy.
X'10'	X'04'	X'00'	X'04'	Verification of branch entry parameter list failed; count of volumes in the volume list is zero.
X'10'	X'04'	X'00'	X'06'	Verification of branch entry parameter list failed; invalid DASPLID field.
X'10'	X'04'	X'00'	X'07'	Verification of branch entry parameter list failed; invalid DASPVER field.
X'10'	X'04'	X'00'	X'08'	Verification of branch entry parameter list failed; invalid DASPLEN field.

Table 48 on page 99 describes the abbreviations for DADSM SCRATCH diagnostic information.

Table 48. Abbreviation Descriptions for DADSM SCRATCH Diagnostic Information (Register 0)

Abbreviation	Description
BPRERET	Return code from the IGGDASU2 SCRATCH preprocessing exit module. For a description of the return code, see *z/OS DFSMS Installation Exits*.
CDMRSN1,2	CDM reason codes (low order 2 bytes). For explanation see "CDM Reason Codes" on page 408.
CHSTAT	Channel status from I/O. For a description of the status, see *z/OS DFSMSdfp Advanced Services*.
COMMRC	DADSM common return code. For a description of that return code see Table 53 on page 104.
COMMRS	DADSM common reason code. For a description of that reason code, see Table 53 on page 104.
CVRET	Return code from the CVAF function. For a description of the return code, see *z/OS DFSMSdfp Advanced Services*.
CVSTAT	CVSTAT field of the CVAF parameter list. For a description of the fields, see "CVSTAT Field Codes" on page 40.
ECBCC	ECB completion code. For a description of the code, see *z/OS DFSMS: Using Data Sets*.
ENQRET	Return code from ENQ. For a description of the return code, see *z/OS MVS Programming: Assembler Services Reference ABE-HSP*.
GETMRET	Return code from GETMAIN.
PRERET	Return code from the IGGPRE00 preprocessing exit module. For a description of the return code, see *z/OS DFSMS Installation Exits*.
RACREAS	RACF reason code. See Diagnostic Code for request type.
RACRET	RACF return code. See Diagnostic Code for request type.
SNS0	First sense byte from the IOB. For a description of the sense byte, see *z/OS DFSMSdfp Advanced Services*.
SNS1	Second sense byte from the IOB. For a description of the sense byte, see *z/OS DFSMSdfp Advanced Services*.
SYSCDE1,2	System completion code (low order 2 bytes). For a description of the completion code, see *z/OS DFSMS Macro Instructions for Data Sets*.
UCBCAPTRC	Return code from IOSCAPU service.
UCBCAPTRS	Reason code from IOSCAPU service.
VDSS1,2	VDSS error reason code (low order 2 bytes). For a description of the VDSS1 and VDSS2 field values, convert the reason code to decimal and see "VTOC/Data Set Services (VDSS) Reason Codes" on page 455.

Status Codes from SCRATCH

After the SCRATCH macro instruction is executed, the last byte of each 12-byte entry in the volume list indicates one of the conditions that are explained in Table 49 on page 100.

Table 49. Status Codes from SCRATCH Macro Instruction

Scratch Status Code	Meaning
0 (X'00')	The format-1 DSCB for the data set has been deleted from the VTOC on the volume indicated.
1 (X'01')	The VTOC of this volume does not contain the format-1 DSCB to be deleted.
2 (X'02')	One of the following conditions occurred: • The data set could not be scratched because the console operator or TSO/E terminal operator did not specify the correct password in the two attempts allowed. • The user tried to scratch a VSAM data space or a catalogued VSAM data set. • The user tried to scratch the VTOC index data set. • An SMS validation failure occurred. • Verification of the *last referenced date* failed.
3 (X'03')	The DSCB was not deleted because either the OVRD option was not specified or the retention cycle had not expired.
4 (X'04')	One of the following conditions occurred: • A permanent I/O error or an invalid F1 DSCB was encountered when processing this volume. • An unexpected CVAF error return code was encountered. • An installation exit rejected the request. • An I/O error occurred while the DASD tracks occupied by the data set were being erased. Either the ERASE option had been specified in the scratch parameter list or the ERASE attribute had been specified for a RACF-defined data set. • Invalid F5 or F7 DSCB encountered, an invalid update to F5 or F7 DSCB was encountered, or an invalid extent in an F1, F2, or F3 was encountered when processing this data set on this volume.
5 (X'05')	It could not be verified that this volume was mounted, nor was there a unit available for mounting the volume.
6 (X'06')	The operator was unable to mount this volume.
7 (X'07')	The format-1 DSCB was not deleted because the data set was open.
8 (X'08')	The format-1 DSCB indicates the data set is defined to RACF, but either you are not authorized to the data set or to the volume, or the data set is a VSAM data space.

After the SCRATCH macro instruction is executed, the next to last byte of the *first* entry in the volume list indicates one of the conditions that are described in Table 50.

Table 50. Secondary Status Codes

Secondary Status Code	Description
0 (X'00')	No secondary status for this volume.
128 (X'80')	The data set was RACF-protected and the user was authorized via the RACF DATASET class to scratch the data set. This means that at least one volume entry was protected.

DADSM UPDATE Return and Diagnostic Codes

Table 51 describes the conditions indicated by the diagnostic information returned in register 0. The abbreviations used in Table 51 are defined in Table 52 on page 103.

Table 51. DADSM UPDATE Diagnostic Information (Register 0)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'04'	X'01'	X'00'	X'01'	Caller is neither APF authorized nor in key 0 nor in supervisor state.
X'04'	X'04'	X'00'	X'15'	Error validating SMS flag. SMS flag in parameter list does not match UCB SMS flag.
X'04'	X'06'	X'ENQRET'	X'00'	Enqueue on TIOT failed.
X'04'	X'07'	X'ENQRET'	X'00'	Enqueue on data set failed. RET = USE.
X'04'	X'08'	X'ENQRET'	X'00'	Enqueue on data set failed. RET = CHNG.
X'04'	X'09'	X'ENQRET'	X'00'	ENQ on VTOC failed.
X'04'	X'0A'	X'ENQRET'	X'00'	Enqueue on demount resource for virtual volume failed.
X'04'	X'0B'	X'CVRET'	X'CVSTAT'	CVAFDIR: Error reading DSCB by data set name.
X'04'	X'0C'	X'CVRET'	X'CVSTAT'	CVAFDIR: Error reading DSCB by CCHHR.
X'04'	X'0D'	X'CVRET'	X'CVSTAT'	CVAFDIR: Error reading DSCB by TTR.
X'04'	X'0E'	X'00'	X'1B'	DAUSMSMV is neither 0 nor 1.
X'04'	X'0E'	X'00'	X'1C'	DAUSMSUV is neither 0 nor 1.
X'04'	X'0E'	X'00'	X'1D'	DAUSMSRV is neither 0 nor 1.
X'04'	X'0E'	X'00'	X'1E'	DAULSTVN is neither 0 nor 1.
X'04'	X'0E'	X'00'	X'1F'	DAUCHGIV is neither 0 nor 1.
X'04'	X'0E'	X'FUNC-ID'	X'20'	Invalid function ID found for format-1 DSCB update request. The return code contains the invalid function ID.
X'04'	X'0E'	X'00'	X'21'	DAUSMSIV is neither 0 nor 1.
X'04'	X'0E'	X'00'	X'22'	DAUSMSCV is neither 0 nor 1.
X'04'	X'0E'	X'FUNC-ID'	X'23'	Invalid function ID found for format-4 DSCB update request. The return code contains the invalid function ID.
X'08'	X'0B'	X'CVRET'	X'CVSTAT'	CVAFDIR: Error reading DSCB by data set name.
X'08'	X'0C'	X'CVRET'	X'CVSTAT'	CVAFDIR: Error reading DSCB by CCHHR.
X'08'	X'0D'	X'00'	X'19'	Request to update format-1 DSCB, but DSCB read is not a format-1 DSCB.
X'08'	X'0D'	X'00'	X'1A'	Request to update format-4 DSCB, but DSCB read is not a format-4 DSCB.
X'08'	X'0F'	X'CVSTAT'	X'CVSTAT'	Error writing DSCB.
X'0C'	X'02'	X'00'	X'05'	Invalid parameter or request. Function area address (DAUFNPTR) is zero.
X'0C'	X'02'	X'00'	X'06'	Invalid parameter or request. Both DAUF1UPD and DAUF4UPD are off.

Table 51. DADSM UPDATE Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'0C'	X'02'	X'00'	X'07'	Invalid parameter or request. Both DAUF1UPD and DAUF4UPD are on.
X'0C'	X'02'	X'00'	X'08'	Invalid parameter or request. Both CCHHR (DAUCCHHP) and TTR (DAUTTRP) are present.
X'0C'	X'02'	X'00'	X'09'	Invalid parameter or request. CCHHR (DAUCCHHP) is on but CCHHR (DAUCCHHR) is zero.
X'0C'	X'02'	X'00'	X'0A'	Invalid parameter or request. TTR (DAUTTRP) is on but TTR (DAUTTR) is zero.
X'0C'	X'02'	X'00'	X'0B'	Invalid parameter or request. UCB address (DAUADUCB) is zero.
X'0C'	X'02'	X'00'	X'0C'	Invalid parameter or request. The UCB address (DAUADUCB) equals the dummy SMS UCB (DACDUCBP).
X'0C'	X'02'	X'00'	X'0D'	Invalid parameter or request. A format-1 DSCB update was requested, but DS name is either blank or zero.
X'0C'	X'02'	X'00'	X'0E'	Invalid parameter or request. The data set name is VTOC index name.
X'0C'	X'02'	X'00'	X'0F'	Invalid parameter or request. The data set name is VSAM VVDS name.
X'0C'	X'02'	X'00'	X'10'	Invalid parameter or request. The reserved field, DAUSKRSV, is not zero.
X'0C'	X'02'	X'00'	X'11'	Invalid parameter or request. The reserved field, DAURSRV1, is not zero.
X'0C'	X'02'	X'00'	X'12'	Invalid parameter or request. The reserved field, DAURSRV2, is not zero.
X'0C'	X'02'	X'00'	X'13'	Invalid parameter or request. The reserved field, DAURSRV3, is not zero.
X'0C'	X'02'	X'00'	X'24'	Invalid parameter or request. Number of functions requested is zero.
X'0C'	X'02'	X'00'	X'25'	Invalid parameter or request. The number of functions requested is greater than the maximum which can be requested.
X'0C'	X'03'	X'00'	X'14'	Error validating UCB passed by caller. ID of UCB is not X'FF'.
X'0C'	X'03'	X'00'	X'17'	Error validating UCB passed by caller. UCB indicates device is not ready.
X'0C'	X'03'	X'00'	X'18'	Error validating UCB passed by caller. VOLSER is blank in UCB passed by caller.
X'0C'	X'05'	X'FUNC-ID'	X'16'	Function ID is greater than maximum or is zero. The subfunction return code will contain the invalid function ID.
X'0C'	X'13'	UCBCAPTRC	UCBCAPTRS	IOSCAPU failed when attempting to capture the UCB passed by the caller.

Table 51. DADSM UPDATE Diagnostic Information (Register 0) (continued)

Byte 0	Byte 1	Byte 2	Byte 3	
DADSM Error Code	**Subfunction Identifier**	**Subfunction Return Code**	**Subfunction Reason Code**	**Description**
X'10'	X'02'	X'00'	X'02'	Invalid parameter list. The DAUPLID field is invalid.
X'10'	X'02'	X'00'	X'03'	Invalid parameter list. The DAUPVER field (version) is invalid.
X'10'	X'02'	X'00'	X'04'	Invalid parameter list. The DAUPLEN field (length) is invalid.

Table 52. Abbreviation Descriptions for DADSM UPDATE Diagnostic Information (Register 0)

Abbreviation	Description
CVRET	Return code from the CVAF function.
CVSTAT	CVSTAT field of the CVAF parameter list.
ENQRET	Return code from ENQ.
FUNC-ID	Function identifier.
UCBCAPTRC	Return code from IOSCAPU service.
UCBCAPTRS	Reason code from IOSCAPU service.

DADSM Common Functions

Create, Extend, and Scratch might have COMMRC returned in the diagnostic code. Table 53 on page 104 describes the diagnostic codes for the DADSM common functions.

DADSM/CVAF Diagnostic Aids

Table 53. DADSM Common Diagnostic Codes

Byte 2	Byte 3	
COMMRC (Function Number)	**COMMRS (Function Description)**	**Description**
X'01'	X'04'	RTA_XXYYZ: Zero value input.
X'01'	X'08'	RTA_XXYYZ: Invalid start RTA.
X'01'	X'0C'	RTA_XXYYZ: Invalid end RTA.
X'01'	X'10'	RTA_XXYYZ: End RTA is not higher than start RTA.
X'01'	X'14'	RTA_XXYYZ: The input RTA value cannot be converted into the 2 byte field (RTA is too large).
X'01'	X'2C'	RTA_XXYYZ: 'YY' value is not less than MAX CYL.
X'02'	X'04'	XXYYZ_RTA: Zero value input.
X'02'	X'2C'	XXYYZ_RTA: 'YY' value is not less than MAX CYL.
X'03'	X'04'	CCHH_RTA: Zero value input.
X'03'	X'18'	CCHH_RTA: End CCHH is less than start CCHH.
X'03'	X'1C'	CCHH_RTA: Start 'CC' value is not less than MAX CYL.
X'03'	X'20'	CCHH_RTA: End 'CC' value is not less than MAX CYL.
X'03'	X'24'	CCHH_RTA: Start 'HH' value is not less than MAX TRKS.
X'03'	X'28'	CCHH_RTA: End 'HH' value is not less than MAX TRKS.
X'04'	X'04'	RTA_CCHH: Zero value input.
X'04'	X'10'	RTA_CCHH: End RTA is not higher than start RTA.
X'04'	X'14'	RTA_CCHH: The input RTA value cannot be converted into the 2-byte field (RTA is too large).
X'04'	X'1C'	RTA_CCHH: Start 'CC' value is not less than MAX CYL.
X'04'	X'20'	RTA_CCHH: End 'CC' value is not less than MAX CYL.
X'05'	X'01'	OLDTONEW: NEWEDT too small.
X'06'	X'01'	NEWTOOLD: NEWEDT eye-catcher invalid.
X'06'	X'02'	NEWTOOLD: NEWEDT level invalid.
X'06'	X'03'	NEWTOOLD: OLDEDT too small.
X'06'	X'04'	NEWTOOLD: An RTA in NEWEDT is bigger than X'FFFF'.
X'07'	X'01'	SORTEDT: NEWEDT eye-catcher invalid.
X'07'	X'02'	SORTEDT: NEWEDT level invalid.
X'08'	X'01'	F5TOEDT: EDT too small.
X'09'	X'01'	EDTTOF5: EDT eye-catcher invalid.
X'09'	X'02'	EDTTOF5: EDT level valid.
X'09'	X'05'	EDTTOF5: EDT entry count is greater than 26.
X'0A'	X'01'	F7TOEDT: EDT too small.
X'0B'	X'01'	EDTTOF7: EDT eye-catcher invalid.
X'0B'	X'02'	EDTTOF7: EDT level invalid.
X'0B'	X'05'	EDTTOF7: EDT entry count is greater than 16.
X'0C'	X'01'	CYL_ALGN: EDT eye-catcher invalid.
X'0C'	X'02'	CYL_ALGN: EDT level invalid.

Table 53. DADSM Common Diagnostic Codes (continued)

Byte 2	Byte 3	
COMMRC (Function Number)	**COMMRS (Function Description)**	**Description**
X'0D'	X'01'	RR_RL: EDT eye-catcher invalid.
X'0D'	X'02'	RR_RL: EDT level invalid.
X'0D'	X'03'	RR_RL: Invalid RTA,RTA+1 pair.
X'0E'	X'01'	RL_RR: EDT eye-catcher invalid.
X'0E'	X'02'	RL_RR: EDT level invalid.
X'0E'	X'03'	RL_RR: Invalid length in RTA, length pair.
X'10'	X'01'	Validate_EDT: VTOC error detected; overlapping extents or out-of-order extents detected; the current extent's xxxRTAST value is less than the prior extent's xxxRTAED value.
X'10'	X'02'	Validate_EDT: VTOC error detected; out-of-order extents or a cycle within the free-space DSCB chain detected; the first extent's xxxRTAST is not greater than the saved xxxRTAED from the previous conversion operation against this extent descriptor table.
X'10'	X'03'	Validate_EDT: VTOC error detected; an invalid extent was detected in the VTOC; an extent contains a xxxRTAST value that is greater than or equal to the extent's xxxRTAED value.
X'10'	X'04'	Validate_EDT: VTOC error detected; an invalid empty Format 5 or Format 7 was found in the VTOC; the input extent descriptor table or free-space DSCB contains no extents, and it is not the first free-space DSCB processed.

DADSM Dumps

DADSM calls CVAF to perform services on its behalf. If DADSM receives a nonzero return code from CVAF, DADSM module IGG032DB is called to produce a dump.

The dump title line is in the following format. See Figure 2 for a sample title line.

```
JOBNAME   DADSM  XXXXXX  ERROR    TCB=yyyyyyyy  W/A=zzzzzzzz
```

Descriptions of the variables in this format follow:

XXXXXX is the failure-related DADSM function.
yyyyyyyy is TCB address.
zzzzzzzz is the DADSM work-area address.

```
KTMJOB     DADSM  EXTEND  ERROR    TCB=00DC6400  W/A=00DFAB20
```

Figure 2. Sample DADSM Dump Title Line

Examine the work area pointed to by *zzzzzzzz* in the preceding example to locate the eye-catcher **CVPL**. The CVFCTN field (CVPL +X'06') and the CVSTAT field (CVPL + X'07') in the failure-related CVPL identify the failure-related CVAF function and CVAF status, respectively. For a description of function code values, see the table of CVFCTN values in the DADSM/CVAF modifier keywords section of *z/OS DFSMSdfp Diagnosis Guide*. For a description of the CVSTAT codes, see "CVSTAT Field Codes" on page 40.

Chapter 6. DASD Common Services Diagnostic Aids

The following major diagnostic aids are provided for Direct Access Storage Device (DASD) common services:

- A description of the contents of the problem determination area (PDA)
- A description of return, reason, status, and response codes from:
  - Common filter services
  - DASD calculation services
  - Device information services

DASD Common Services Error Recovery and Detection

The DASD common services problem determination area (PDA) contains the following failure-related information:

- Module name
- Return code
- Reason code
- Most recent calling function

PDA information is contained in the failure-related ISPF log entry's FEEDBACK field. See the first entry of the sample ISPF log in Figure 10 on page 149 for an example. The in-storage format of the PDA follows.

DASD Common Services—Problem Determination Area (PDA)

The DASD common services PDA provides the keyword-oriented diagnostic information as noted above. Table 54 provides a detailed description of all PDA information.

Table 54. PDA Information

Offset Dec(Hex)	Bytes and Bit Pattern	Field Name	Description
0 (X'00')	2	DISPDRTN	Return code
2 (X'02')	1	DISPDCM1	A comma
3 (X'03')	3	DISPDRSN	Reason code
6 (X'06')	1	DISPDCM2	A comma
7 (X'07')	8	DISPDNAM	Name of the module detecting the error
15 (X'0F')	1	DISPDCM3	A comma
16 (X'10')	4	DISPDISP	Hexadecimal displacement from module entry point to the instruction detecting the error
20 (X'14')	1	DISPDCM4	A comma
21 (X'15')	3	DISPDID1	ID of the most recent calling function
24 (X'18')	1	DISPDCM5	A comma
25 (X'19')	3	DISPDID2	ID of second most recent calling function
28 (X'1C')	1	DISPDCM6	A comma
29 (X'1D')	3	DISPDID3	ID of third most recent calling function

© Copyright IBM Corp. 1979, 2002

DASD Common Services Return and Reason Codes

The following sections describe these DASD diagnostic aids:
- Common filter services
- DASD calculation services
- Device information services

Common Filter Services Return and Reason Codes

Common filter services passes a return code and a reason code to the caller in registers 15 and 0 respectively. Return code 0 in register 15 indicates that filtering has been performed successfully; in this case register 0 contains reason code 0. A nonzero return code in register 15 indicates that filtering has terminated with errors.

Table 55 describes nonzero return codes and their reason and response codes.

Table 55. Nonzero Return Codes and Applicable Reason Codes

Return Code	Description
8 (X'08')	Filtering has terminated with an error.
	Reason Code: The error is identified by the following hexadecimal values in register 0:
	X'01' Key area field is zero
	X'02' Key count field is zero.
	X'03' Key pointer field is zero.
	X'04' Key length field is zero.
	X'05' Key length is larger than maximum size.
	X'06' Internal key pointer is zero.
	X'07' Internal key length is less than minimum.
	X'08' Internal key hash code is invalid.
	X'09' Data area pointer is zero.
	X'0A' Data count field is zero.
	X'0B' Data pointer is zero.
	X'0C' Data length is zero.
	X'0D' Parameter list length is incorrect.
	X'0E' Macro level is incorrect.
	X'0F' Match count pointer is zero.
	X'10' Status pointer is zero.
	X'11' Specification of separator is invalid.
	X'12' Specification of substituent is invalid.
	X'13' Filtering option is invalid.
	X'14' Keywords INTERNAL and PARSONLY are mutually exclusive.
	X'1E' Filter key is invalid.
	X'1F' Internal key is invalid.
	X'20' Data format is invalid.

DASD Calculation Services Return, Reason, and Status Codes

DASD calculation services passes a return code and a reason code to the caller in registers 15 and 0 respectively. Return code 0 in register 15 indicates that DASD calculation services has completed successfully; in this case register 0 contains reason code 0. A nonzero return code indicates an error condition.

As cited in the following table, a *status code* may be associated with a nonzero return code. The *status code* is passed back in an information return area pointed to by the DASD calculation services parameter list. This field is not normally accessible to you, because the call to DASD calculation services is normally via an internal system call.

Table 56 describes nonzero return codes and their reason and *status codes* if applicable.

Table 56. Nonzero Return Codes and Applicable Reason and Status Codes

Return Code	Description
4 (X'04')	At least one data set information request had a nonzero *status code*. **Reason Code:** Register 0 contains the number of the first data set information request whose *status code* was nonzero. The possible values of the *status code* are listed in hexadecimal below: **X'00'** Completed processing of request without error. **X'04'** Invalid return code or value received from precalculation exit (IGBDCSX1); processing continues with DCS defaults. **X'08'** Invalid return code or value received from postcalculation exit (IGBDCSX2); processing continues with DCS defaults. **X'0C'** Both installation exits passed invalid data to DCS; processing continues with DCS defaults. **X'10'** Called routine TRKCALC returned with error for this data set request. **X'14'** Insufficient information supplied for a data set information request. No calculations performed. **X'18'** DSCB list is invalid. **X'1C'** Data set is a storage index. No calculations performed. **X'28'** No DSCB list is supplied for a data set space request.
16 (X'10')	Return area provided is too small. DCS checks the size of the return area before processing and determines if it is large enough to contain the requested output. No processing is performed. **Reason Code:** Register 0 contains the size of the return area required.

Table 56. Nonzero Return Codes and Applicable Reason and Status Codes (continued)

Return Code	Description
20 (X'14')	Caller provided an invalid value in the parameter list. No processing is performed. **Reason Code:** The user error is identified by the following hexadecimal values in register 0: **X'00'** Size of parameter list is invalid. **X'04'** Parameter list is not owned by the caller. **X'08'** Device specification is invalid. **X'0C'** RTNAREA is not specified (address is zero). **X'10'** RTNAREA is not owned by caller. **X'14'** PDAREA is not specified (address is zero). **X'18'** PDAREA is not owned by caller. **X'1C'** PDAREA is too small. **X'20'** DCSDS table is not owned by caller. **X'24'** No requests are specified. **X'28'** Parameter list level is invalid.
28 (X'1C')	Abend is detected with RETRY parameter specified. See the PDA for more error information.
32 (X'20')	Storage is unavailable. **Reason Code:** Register 0 contains the return code from the GETMAIN routine.
40 (X'28')	Error is encountered in call to DIS. **Reason Code:** Register 0 contains the DIS return code.
44 (X'2C')	Error is encountered in call to TRKCALC for a globally used value (for example, not data set specific). **Reason Code:** Register 0 contains the TRKCALC return code.

Device Information Services Return, Reason, and Response Codes

Device information services passes a return code and a reason code to the caller in registers 15 and 0 respectively. Return code 0 in register 15 indicates that device information services has completed successfully; in this case register 0 contains reason code 0. A nonzero return code indicates an error condition that caused termination before processing was complete.

As cited in the following table, a *response code* may be associated with a nonzero return code. The *response code* is passed back in an information return area pointed to by the device information services parameter list. This field is not normally accessible to you, because the call to device information services is normally via an internal system call.

Table 57 on page 111 describes nonzero return codes and their reason codes and *response codes* if applicable.

Table 57. Nonzero Return Codes and Applicable Reason and Response Codes

Return Code	Description
4 (X'04')	Request was completed, but one or more entries in the return area contain a nonzero *response code*. **Reason Code:** Register 0 contains the address of the first entry containing a nonzero *response code*. The following values are the possible values of the *response code*: **X'01'** Volume serials on device are requested, but device is not tape or DASD. **X'02'** Volume attributes of device are requested, but device is not tape or DASD. **X'04'** Device name is requested, but IEFEB4UV cannot find the corresponding look-up value in the EDT. **X'08'** Tracks or cylinders are requested, but device is not DASD. **Note:** These *response codes* are bit-oriented. A combination of bits indicate multiple errors.
8 (X'08')	Request could not be completed with the return area provided. A RESUME may be used to continue from the last completed UCB. **Reason Code:** Register 0 contains a resume value.
12 (X'0C')	Request could not be completed with the return area provided and one or more entries in the return area contain a nonzero *response code*. A RESUME may be used to continue from the last completed UCB. **Reason Code:** Register 0 contains a resume value.
16 (X'10')	Caller provided an invalid parameter in the parameter list. **Reason Code:** The user error is identified by the following hexadecimal **X'04'** Storage for the parameter list is not owned by the caller. **X'08'** Size of the parameter list is invalid. **X'0C'** Storage for the return area is not provided or is not owned by the caller. **X'10'** Storage for the volume serial list is not owned by the caller. **X'14'** Storage for the device number list is not owned by the caller. **X'18'** Storage for the PDA is not provided, is not owned by the caller, or is too small. **X'1C'** Storage for the return area is too small to contain at least a header and one entry. **X'20'** A bad resume code was passed. **X'24'** No matches: search arguments given are too restrictive. **X'28'** No requests are specified. **X'2C'** Parameter list level is invalid. **X'30'** Conflicting feature request. **X'34'** Device number passed is invalid. **X'38'** Device number range passed is invalid. **X'3C'** A release call was made, but there was no storage to free.

Table 57. Nonzero Return Codes and Applicable Reason and Response Codes (continued)

Return Code	Description
20 (X'14')	An internal or system error is encountered. Register 0 contains more specific information about the error. **Reason Code:** The user error is identified by the following hexadecimal values in register 0: **X'04'** The JESCT pointer to the EDT is zero. **X'08'** The service level of device information services cannot coexist with that of the EDT.
24 (X'18')	An error return code was received from common filter service. **Reason Code:** Register 0 contains the return code from common filter service.
28 (X'1C')	An abend has occurred. **Reason Code:** Register 0 contains the abend code provided by RTM.
32 (X'20')	A nonzero return code is returned from IEFEB4UV. **Reason Code:** Register 0 contains the IEFEB4UV return code.
36 (X'24')	An error is found in call to DEVTYPE. **Reason Code:** Register 0 contains the return code from DEVTYPE.

Chapter 7. DESERV Macro Diagnostic Aids

The section lists the reason codes for the DESERV macro system error return code DESRC_SEVE, X'24'. All other DESERV macro return and reason codes can be found in *z/OS DFSMS Macro Instructions for Data Sets*.

Return Code DESRC_SEVE

A return code of DESRC_SEVE should be considered to be a terminating error. The reason codes are provided for Diagnosis, Modification, and Tuning Information (DMTI).

Reason Codes returned by the DESERV macro

DESERV reason codes returned from the macro invocation are fullword values. The values listed here are the low order halfword values. The high order byte of the fullword reason code will be X'27', and the next byte contains a code that identifies the module that detected the error.

The reason codes that follow are listed by the DESERV function invoked.

DESERV ALL Function Reason Codes

Reason Code	Meaning
1031(X'407')	Bad return code from SETLOCK. DESRS_SETLOCK_ERR
1079(X'437')	Nonzero return code from an IGWFESTK request. DESRS_ADD_STACK_FAILED
1095(X'447')	Issued by the DESERV recovery routine when entered for an unknown reason (for example, program check). DESRS_UNKNOWN

© Copyright IBM Corp. 1979, 2002

DESERV GET Function Reason Codes

Reason Code	Meaning
1026(X'402')	PMARA to PMAR merge failed. DESRS_PMAR_MERGE_ERR
1027(X'403')	No space to store connections. DESRS_ELEMENT_CREATE_FAILED
1028(X'404')	Number of CONNIDs exceeded. DESRS_ANCHOR_CREATE_FAILED
1029(X'405')	IGWFTOKM SET failed. DESRS_SET_ERROR
1030(X'406')	IGWFTOKM EXTRACT failed. DESRS_EXTRACT_ERROR
1031(X'407')	Bad return code from SETLOCK. DESRS_SETLOCK_ERR
1032(X'408')	Error converting PDSDE to PMAR. DESRS_PDSDE_CONVRT_ERR
1033(X'409')	JCDM returned an unexpected return and reason code combination. DESRS_UNEXPECTED_JCDM_ERROR
1035(X'40B')	FREEMAIN failure. DESRS_FREEMAIN_ERROR
1088(X'440')	PDSDE merge into PMAR failed. DESRS_PDSDE_MERGE_ERR
1092(X'444')	A call to an internal routine failed. DESRS_IGWCDFNC_ERROR
1093(X'445')	A call to IGWFECHG failed. DESRS_CANT_CHANGE_RECOVERY
1094(X'446')	A call to GETMAIN failed. DESRS_GETMAIN_ERROR
1095(X'447')	BLDL returned an unknown return and reason code. DESRS_UNKNOWN
1098(X'44A')	DESERV exit returned an unexpected return code. DESRS_EXIT_ERROR

DESERV PUT Function Reason Codes

Reason Code	Meaning
1029(X'405')	IGWFTOKM SET failed. DESRS_SET_ERROR
1030(X'406')	IGWFTOKM EXTRACT failed. DESRS_EXTRACT_ERROR
1031(X'407')	Bad return code from SETLOCK. DESRS_SETLOCK_ERR
1033(X'409')	JCDM returned an unexpected return and reason code combination. DESRS_UNEXPECTED_JCDM_ERROR
1035(X'40B')	FREEMAIN failure. DESRS_FREEMAIN_ERROR
1093(X'445')	A call to IGWFECHG failed. DESRS_CANT_CHANGE_RECOVERY
1094(X'446')	A call to GETMAIN failed. DESRS_GETMAIN_ERROR
1096(X'448')	A call to the pacb_locate routine failed. DESRS_PACBLOC_ERROR
1098(X'44A')	DESERV exit returned an unexpected return code. DESRS_EXit_ERROR

DESERV GET_ALL Function Reason Codes

Reason Code	Meaning
1031(X'407')	Bad return code from SETLOCK. DESRS_SETLOCK_ERR
1033(X'409')	JCDM returned an unexpected return and reason code combination. DESRS_UNEXPECTED_JCDM_ERROR
1087(X'43F')	A call to an internal routine failed. DESRS_IEWBXILO_ERROR
1094(X'446')	A call to GETMAIN failed. DESRS_GETMAIN_ERROR

DESERV GET_NAMES Function Reason Codes

Reason Code	Meaning
1025(X'401')	IEWLCNVT macro failed. DESRS_IEWLCNVT_ERROR
1031(X'407')	Bad return code from SETLOCK. DESRS_SETLOCK_ERR
1033(X'409')	JCDM returned an unexpected return and reason code combination DESRS_UNEXPECTED_JCDM_ERROR
1035(X'40B')	FREEMAIN failure, DESRS_FREEMAIN_ERROR
1056(X'420')	Internal format of name is inconsistent. DESRS_COMP_NAME_BAD
1087(X'43F')	A call to an internal routine failed. DESRS_IEWBXILO_ERROR
1090(X'442')	An unexpected Standard Attribute Record (SAR) code was found when reading attribute records. DESRS_UNEXPECTED_SAR_CODE
1094(X'446')	A call to GETMAIN failed. DESRS_GETMAIN_ERROR

DESERV RELEASE Function Reason Codes

Reason Code	Meaning
1030(X'406')	IGWFTOKM EXTRACT failed. DESRS_EXTRACT_ERROR
1031(X'407')	Bad return code from SETLOCK. DESRS_SETLOCK_ERR

Chapter 8. Distributed FileManager/MVS Diagnostic Aids

The following diagnostic aids are provided for Distributed FileManager/MVS:

- A description of errors:
  - Basic error processing
  - Resource problems
  - System logic errors
  - DDM data stream errors
  - Unexpected source termination
  - Communication errors
- A description of messages:
  - Distributed FileManager/MVS system messages
  - DDM reply messages
- A description of Distributed FileManager/MVS problem determination tools:
  - Trace facility for Distributed FileManager/MVS
  - TRACE command
  - DFM trace options
  - Formatting the DFM trace table
  - IPCS dump formatting with SMSXDATA for DFM
  - IPCS Messages
  - Formatting LOGREC records
  - Slip trapping for a specific reply message
  - GTF tracing of input/output buffers
- A description of Distributed FileManager/MVS server-diagnostic-information field (SRVDGN) data items
- A description of implementation specific error codes:
  - Distributed FileManager/MVS general error codes
  - Distributed FileManager/MVS communications error codes
  - Distributed FileManager/MVS agent error codes
  - Distributed FileManager/MVS file/directory error codes
  - Distributed FileManager/MVS locking error codes
  - Distributed FileManager/MVS access method error codes

Distributed FileManager/MVS Errors

The following section documents these types of errors:

- Basic error processing
- Resource problems
- System logic errors
- DDM data stream errors
- Unexpected source termination
- Communication errors

Basic Error Processing

When Distributed FileManager/MVS detects an error while processing a DDM command, the appropriate architected DDM error reply message is sent back to the source system that generated the DDM command stream. The reply message contains a server-diagnostic-information field (SRVDGN) and an error code. The server-diagnostic-information field of the error reply message contains up to 255 bytes of diagnostic data. This data can be used to correlate the error with a dump, LOGREC record, or other diagnostic information.

© Copyright IBM Corp. 1979, 2002

Distributed FileManager/MVS Diagnostic Aids

For a description of Distributed FileManager/MVS specific error codes included in the SRVDGN, see the following list:

- “Distributed FileManager/MVS General Error Codes” on page 137
- “Distributed FileManager/MVS Communications Error Codes” on page 139
- “Distributed FileManager/MVS Agent Error Codes” on page 140
- “Distributed FileManager/MVS File/Directory Error Codes” on page 141
- “Distributed FileManager/MVS Locking Error Codes” on page 141
- “Distributed FileManager/MVS Access Method Error Codes” on page 141

By using the error codes, source system users or MVS system programmers will be able to determine actions necessary to resolve the error. For example, the returned reply message FILNFNRM means ″file not found″. This means that the command processor failed to find the file because it does not exist on the MVS system. The end user issuing the request may or may not anticipate such a response as normal depending on the application logic. Another possibility could be that the specified file was not found because the file name input in the DDM command was misspelled.

Resource Problems

When Distributed FileManager/MVS detects a problem with a resource (for example, data set, storage, file lock), an attempt is made to free the resource or wait for the resource to become available (waiting does not occur with all resources). A ″command check″ (CMDCHKRM) reply message is returned to the source system after Distributed FileManager/MVS detects a resource problem. Command check usually indicates a problem with the software or hardware. This problem requires diagnosis. Command check is not limited to an inability to free a resource. In most cases, the resource could not be obtained because of a logic error— not because it is temporarily unavailable. If the resource is unavailable after a period of time, the ″resource limits reached″ (RSCLMTRM) reply message is returned to the source system. See Table 58 on page 125, Table 59 on page 125, and Table 60 on page 125 for more information about RSCLMTRM reply messages.

System Logic Errors

The Distributed FileManager/MVS support runs under an EUT FRR. For special situations, the recovery environment is temporarily changed to an ESTAE environment. When a system logic or hardware error is detected by Distributed FileManager/MVS, first failure data capture (FFDC) is performed. This results in a dump that includes all of the pertinent data necessary for debugging the error. The error is also recorded in the system LOGREC file. After the error is recorded, retry and/or end-of-memory processing is performed. This process consists of flushing the buffers and cache, closing all open data sets, and freeing all outstanding locks, latches, and other resources.

If the detected error impacts the entire conversation, a “permanent agent error” (AGNPRMRM) reply message is returned to the source system, and the conversation is terminated. If the detected error impacts only the agent, AGNPRMRM is returned to the source, and the agent is terminated. If the detected error only impacts the command being processed, a “command check” (CMDCHKRM) reply message is returned to the source system, and the command process is terminated. For details about AGNPRMRM and CMDCHKRM reply messages, see “DDM Reply Messages Returned by Distributed FileManager/MVS” on page 125.

DDM Data Stream Errors

When Distributed FileManager/MVS detects a problem with the input DDM data stream, a “syntax error” (SYNTAXRM), “protocol error” (PRCCNVRM), or “error communications message” (ERRCMNMSG) reply message is built by the reply message builder. Additionally, for SYNTAXRM and PRCCNVRM, the reply message builder writes a record to the system LOGREC file. The LOGREC record includes return and reason codes that identify the subcomponent/module that detected the error. The entire reply message and its SRVDGN is included in the LOGREC record.

A DDM data stream error is categorized as a “source system error.” For a discussion of Distributed FileManager/MVS diagnostic techniques that examine DDM data stream errors on the MVS system, see “Slip Trapping for a Specific Reply Message” on page 132, “GTF Tracing of Input/Output Buffers,” and “Formatting LOGREC Records” on page 132. However, because the DDM data stream is not necessarily generated by the end user on the source system, some intervening software or hardware product could have generated the error. (Note: If this type of error is discovered, diagnostic techniques outside of Distributed FileManager/MVS might be required.)

Unexpected Source Termination

APPC/MVS notifies Distributed FileManager/MVS when the source system has terminated. Distributed FileManager/MVS closes all data sets opened for this source system’s conversation. Once the data sets are closed, the conversation is terminated.

Communication Errors

Distributed FileManager/MVS detects four types of communication errors:

- A SEND ERROR was issued by the source. Distributed FileManager/MVS stops all processing in the conversation, but processing does not terminate.
- APPC/MVS returns an unexpected error return code. Distributed FileManager/MVS logs the error into the system LOGREC file and terminates the conversation.
- APPC/MVS returns an error code that indicates a system logic error in Distributed FileManager/MVS. See “System Logic Errors” on page 118 for a description of processing in this case.
- APPC/MVS indicates a DEALLOCATE ABEND of the source system. Distributed FileManager/MVS performs no error recording; it simply terminates the conversation.

If a source system attempts to communicate with an MVS system that has the Distributed FileManager/MVS installed but not active, an AGNPRMRM reply message with error code X'0021' (see “Distributed FileManager/MVS General Error Codes” on page 137) is returned to the source system.

Distributed FileManager/MVS Messages

For Distributed FileManager/MVS to process input DDM commands, the Central Address Space (DFM), which is a started system procedure, the APPC conversation, and the APPC Transaction Program address space (GDEDFM) must be successfully initialized. Failures in this initialization process are reported to the operator console. See “Distributed FileManager/MVS System Messages” for more information. Additionally, an AGNPRMRM is returned. For more information, see Table 58 on page 125.

After the Central Address Space, APPC conversation, and APPC Transaction Program are successfully initialized, the DDM data stream input from the source system is screened for syntax errors (SYNTAXRM), protocol errors (PRCCNVRM), and error communication messages (ERRCMNMSG). See Table 58 on page 125, Table 59 on page 125, and Table 60 on page 125 for more information.

If the DDM data stream is correct syntactically and conforms to the DDM protocol, the DDM commands are screened for the following errors before the command processors attempt to process the request:

INVRQSRM “invalid request”. The request is not valid, according to the DDM architecture.

TRGNSPRM “target not supported”. The DDM command is not supported by Distributed FileManager/MVS.

CMDNSPRM “command not supported”. The DDM command is supported in a level that exceeds the agreed-upon manager level.

PRMNSPRM “parameter not supported”. The DDM command parameter is supported in a level that exceeds the agreed-upon manager level.

The DDM commands that successfully pass the preceding filters are next passed to the command processors. The command processors screen for errors detectable at that level, for example, command parameter error, environmental errors, execution errors. See Table 61 on page 126 for more information.

Distributed FileManager/MVS System Messages

Distributed FileManager/MVS issues the following messages:

GDE001E **DISTRIBUTED FILEMANAGER TARGET SYSTEM INITIALIZATION FAILED** *additional message text*.

Explanation: A problem occurred during initialization of the Distributed FileManager/MVS target function. The initialization was requested by a START command. The additional message text explains the problem:

null message text

A system error occurred. Refer to the preceding error messages to determine the specific error. If necessary, call IBM Service for programming support.

ECSA STORAGE COULD NOT BE OBTAINED. *n***HEX BYTES REQUIRED**.

A request for ECSA storage could not be fulfilled. *n* indicates the number of ECSA (hex) bytes that are required to initialize Distributed FileManager/MVS. Increase the ECSA size by at least *n* and reinitialize Distributed FileManager/MVS

GDEIGBS COULD NOT BE LOADED. LOAD ABEND CODE =*cccccccc* **REASON CODE** =*rrrrrrrr* **ALIAS NAME** =*nnnnnnnn*

The DFM load module GDEIGBS could not be loaded because of an error detected by the LOAD function during an attempt to obtain the address of the *nnnnnnnn* CSECT. Check the LOAD ABEND *(cccccccc)* and reason *(rrrrrrrr)* codes issued and take corrective action. If no corrective action solves the problem, call IBM Service for programming support.

Note: An ABEND code of 99 and a reason code of 99 are set by Distributed FileManager/MVS to indicate that the loaded ALIAS NAME was link-edited above the 16M line. ALIAS NAME must be link-edited below the 16M line.

Operator Response: Notify the system programmer.

System Programmer Response: See **Explanation** for specific answers and problems.

Problem Determination: Refer to the ABEND or reason codes. These codes are contained in the message text or any related console messages.

System Action: The Distributed FileManager/MVS target function is inactive.

Descriptor Codes: 5 (immediate command response); 12 (important information)

Routing Information: This information is sent to the console that issued the START DFM command.

GDE002I **DISTRIBUTED FILEMANAGER TARGET CONVERSATION INITIALIZATION FAILED FOR ASID**=*aaaaaaaa* **USERID** =*uuuuuuuu* **PRTNRLU**=*nnnnnnnn* **CONVID**=*cccccccc additional message text*

Explanation: Distributed FileManager/MVS was attempting to initialize the Distributed FileManager/MVS target transaction program (TP) in address space *aaaaaaaa* in response to an APPC ALLOCATE request. A Distributed FileManager/MVS source system issues this request. The Distributed FileManager/MVS target TP initialization abnormally terminated. The following message text describes the specific problem:

null message text

A system error occurred. Refer to the preceding error messages to determine the specific error. If necessary, call IBM Service for programming support.

PRIVATE STORAGE COULD NOT BE OBTAINED. *n* **HEX BYTES REQUIRED**.

A request for private storage could not be fulfilled. *n* indicates the number of private storage bytes (in hex) that is required to initiate the conversation. Increase the conversation address space's region space by at least *n*. Retry the Distributed FileManager/MVS source LU. The region size is specified in two places: The JCL TP profile, and the TPDEFAULT REGION (xx) parameter in the ASCHPMxx member. The ASCHPMxx member is in the system PARMLIB (for example, SYS1.PARMLIB). For a description of ASCHPMxx, see the MVS Initialization and Tuning Guide.

ECSA STORAGE COULD NOT BE OBTAINED..*n* **HEX BYTES REQUIRED**

A request for ECSA storage could not be fulfilled. *n* indicates the number of ECSA bytes (in hex) that is required to initialize the system. Increase the ECSA size by at least *n* hex. Retry the Distributed FileManager/MVS source LU.

ATTACH ERROR. ATTACH RTNCD1=*rrrrrrrr*

An error was detected by the ATTACH function. Check the ATTACH return code *(rrrrrrrr)*, and correct the problem. Retry the DFM source LU.

APPC ERROR. VERB=*vvvvvvvv* **RTNCD** =*rrrrrrrr*

An APPC routine detected an error. The APPC verb routine supplies the name of the processed APPC verb *(vvvvvvvv)*. The APPC verb routine also supplies the return code *(rrrrrrrr)*. Correct the problem. Retry the Distributed FileManager/MVS source LU.

The following additional information is provided:

USERID *uuuuuuuu* is the MVS user ID specified on the conversation allocation request.

PRTNRLU *nnnnnnnnnnnnnnnn* is the name of the logical unit (LU). The Distributed FileManager/MVS source server that requests conversation allocation is located in the LU.

CONVID *cccccccc* is the conversation ID. APPC/MVS assigns the conversation ID to this conversation.

Operator Response: Notify the system programmer.

System Programmer Response: See **Explanation** for specific answers and problems.

Problem Determination: If a system error occurred, an ABEND is issued, and the error is logged in the system LOGREC file (for example, SYS1.LOGREC). A dump is also produced and placed in a system dump data set (for example, SYS1.DUMP00). In all other cases, an ABEND is issued. The error is not recorded, and no dump is taken.

System Action: The conversation is terminated.

Descriptor Code: 6 (informational job status)

Routing Information: 10 (system/error maintenance)

GDE003I DISTRIBUTED FILEMANAGER TARGET SYSTEM INITIALIZATION COMPLETE

Explanation: Initialization of the Distributed FileManager/MVS target function was successful.

System Action: Distributed FileManager/MVS is available for use.

Descriptor Code: 5 (immediate command response)

Routing Information: The information is sent to the console that issued the START DFM command.

GDE004E DISTRIBUTED FILEMANAGER TARGET MAIN ADDRESS SPACE FAILED IN ASID=*aaaaaaaa*

Explanation: The main Distributed FileManager/MVS address space (named DFM) abnormally terminated in ASID *aaaaaaaa*

Operator Response: Notify the system programmer.

System Programmer Response: Determine the reason for failure, and take corrective action. This error message precedes other error messages. Correct the preceding error messages. To restore full Distributed FileManager/MVS target function, restart Distributed FileManager/MVS.

Problem Determination: The error is logged in the system LOGREC file (for example, SYS1.LOGREC). A dump is also produced and placed in a system dump data set (for example, SYS1.DUMP00).

System Action: The Distributed FileManager/MVS DFM DSAS is terminated. The Distributed FileManager/MVS conversation address spaces runs continuously.

Descriptor Codes: 4 (informational system status); 12 (important informational)

Routing Information: 10 (system/error maintenance)

GDE004E DISTRIBUTED FILEMANAGER TARGET CENTRAL ADDRESS SPACE FAILED IN ASID=*aaaaaaaa* **REASON**=*bbbbbbbb*

Explanation: The central Distributed FileManager/MVS address space abnormally terminated in ASID:*aaaaaaaa***REASON**:*bbbbbbbb*

Operator Response: Notify the system programmer.

System Programmer Response: Determine the reason for failure, and take corrective action. This error message precedes other error messages. Correct the preceding error messages. To restore full Distributed FileManager/MVS target function, restart Distributed FileManager/MVS.

Problem Determination: The error is logged in the system LOGREC file (for example, SYS1.LOGREC). A dump is also produced and placed in a system dump data set (for example, SYS1.DUMP00).

System Action: The Distributed FileManager/MVS DSAS is terminated. The Distributed FileManager/MVS conversation address spaces runs continuously.

Descriptor Codes: 2 (informational system status); 12 (important informational)

Routing Information: 10 (system/error maintenance)

GDE005I DISTRIBUTED FILEMANAGER TARGET CENTRAL SYSTEM ALREADY ACTIVE

Explanation: The Distributed FileManager/MVS target function is already active.

System Action: The START DFM command is ignored.

Descriptor Code: 4 (informational system status)

Routing Information: 10 (system/error maintenance)

GDE005I DISTRIBUTED FILEMANAGER TARGET SYSTEM NOT ACTIVE, START DFM

Explanation: The Distributed FileManager/MVS target function is inactive.

System Action: Jobs that require the Distributed FileManager/MVS target function are terminated. These jobs terminate with nonzero return codes.

Descriptor Code: 4 (informational system status)

Routing Information: 10 (system/error maintenance)

GDE006E INVALID KEYWORD kkkkkkkk DISTRIBUTED FILEMANAGER TARGET SYSTEM NOT STARTED

Explanation: The Distributed FileManager/MVS start parameter member DFM00 in the system PARMLIB has a keyword which is invalid.

System Action: The Central Address Space (system procedure DFM) is not started. Any subsequent attempt to start an APPC conversation with Distributed FileManager/MVS fails because the Central Address Space is not active.

Descriptor Code: 4 (informational system status)

Routing Information: 10 (system/error maintenance)

DDM Reply Messages Returned by Distributed FileManager/MVS

Tables 58, 59, 60, and 61 list DDM reply messages returned by Distributed FileManager/MVS.

Table 58. Message Returned during Conversation Initialization

DDM Reply Message	Possible Distributed FileManager/MVS Hex Error Code
AGNPRMRM	0021

Table 59. Messages Returned by Non-Control DSSs by Communications Manager/Parser

DDM Reply Message	Possible Distributed FileManager/MVS Hex Error Codes
AGNPRMRM	0001
CMDCHKRM	0001
CMDNSPRM	3005
PRCCNVRM	3002 (PRCCNVCDs: X’01’-X’06’)
PRMNSPRM	0007
RSCLMTRM	0006, 0013
SYNTAXRM	0025 (SYNERRCDs: X’01’-X’05’, X’07’-X’0C’, X’0E’, X’10’, X’12’, X’16’-X’1A’, X’1D’)
VALNSPRM	0007, 0026

Table 60. Reply Messages Returned for Control DSSs by Communications Manager

General Condition	DDM Reply Message	Possible DDM Error Codes
DSS header < 6	SYNTAXRM	ENUVAL: X’01’
DSS header len not = #bytes	SYNTAXRM	ENUVAL: X’02’
DSS header byte not D0	SYNTAXRM	ENUVAL: X’03’
Not CMNMLT, but CMNDSS input	SYNTAXRM	ENUVAL: X’04’
Insufficient storage to support new agent	RSCLMTRM	ENUVAL: X’0006’
Agent limit exceeded (SWTAGN)	RSCLMTRM	Error Code X’0023’
Attach error (SWTAGN)	CMDCHKRM	Error Code X’0024’
AGNNAM missing (SWTAGN AGNABN TRMRCVAGN)	ERRCMNMSG	DSSERRCD: X’07’
Not valid CMNDSS	ERRCMNMSG	DSSERRCD: X’08’
SWTAGN does not follow EXCSAT	ERRCMNMSG	DSSERRCD: X’0A’

Distributed FileManager/MVS Diagnostic Aids

Table 61. DDM Reply Messages Returned by Distributed FileManager/MVS

DDM Reply Message	Codepoint	Description
ACCINTRM	X'1266'	Access intent not valid for access method
ACCMTHRM	X'1231'	Invalid access method
AGNPRMRM	X'1232'	Permanent agent error
CHGFATRM	X'1261'	Change file attributes rejected
CLSDMGRM	X'125E'	File closed with damage
CMDCHKRM	X'1254'	Command check
CMDCMPRM	X'124B'	Command processing complete
CMDNSPRM	X'1250'	Command not supported
CSRNSARM	X'1205'	Cursor not selecting a record position
DCLCNFRM	X'1220'	Declare conflict
DCLNAMRM	X'1256'	Invalid declared name
DCLNFNRM	X'1257'	Declared name not found
DFTRECRM	X'1204'	Default record error
DTARECRM	X'1206'	Invalid data record
DUPDCLRM	X'1255'	Duplicate declared name
DUPFILRM	X'1207'	Duplicate file name
DUPKSIRM	X'1209'	Duplicate key same index
DUPRNBRM	X'120A'	Duplicate record number
ENDFILRM	X'120B'	End of file condition
EXSCNDRM	X'123A'	Existing condition
FILATHRM	X'123B'	Not authorized to file
FILDMGRM	X'125A'	File damaged
FILFULRM	X'120C'	File is full
FILIUSRM	X'120D'	File in use
FILNAMRM	X'1212'	Invalid file name
FILNFNRM	X'120E'	File not found
FILNOPRM	X'1211'	File not open
FILOLORM	X'121D'	File open lock option changed
FILSNARM	X'120F'	File space not available
FILTNARM	X'121E'	File temporarily not available
INTATHRM	X'125C'	Not authorized for open intent for named file
INVRQSRM	X'123C'	Invalid request
KEYDEFRM	X'123D'	Invalid key definition
KEYLENRM	X'122D'	Invalid key length
KEYMODRM	X'1260'	Key value modified after cursor was last set
KEYUSIRM	X'123F'	Key update not allowed by same index
KEYVALRM	X'1240'	Invalid key value
MGRLVLRM	X'1210'	Manager level conflict
NEWNAMRM	X'124F'	Invalid new file name
OBJNSPRM	X'1253'	Object not supported

Table 61. DDM Reply Messages Returned by Distributed FileManager/MVS (continued)

DDM Reply Message	Codepoint	Description
OPNCNFRM	X'1242'	Open conflict error
PRCCNVRM	X'1245'	Conversational protocol error
PRMNSPRM	X'1251'	Parameter not supported
RECCNTRM	X'125B'	Record count mismatch
RECDMGRM	X'1249'	Record damaged
RECINARM	X'1259'	Record inactive
RECIUSRM	X'124A'	Record in use
RECLENRM	X'1215'	Record length mismatch
RECNAVRM	X'126F'	Record not available
RECNBRRM	X'1224'	Record number out of bounds
RECNFNRM	X'1225'	Record not found
RSCLMTRM	X'1233'	Resource limits reached
STRDMGRM	X'1268'	Stream damaged
SUBSTRRM	X'1265'	Invalid substream
SYNTAXRM	X'124C'	Data stream syntax error
TRGNSPRM	X'125F'	Target not supported
UPDCSRRM	X'124D'	Update cursor error
UPDINTRM	X'124E'	No update intent on record
VALNSPRM	X'1252'	Parameter value not supported

Distributed FileManager/MVS Problem Determination Tools

The following section documents these types of problem determination tools:

- Trace facility for Distributed FileManager/MVS
- TRACE command
- DFM trace options
- Formatting the DFM trace table
- IPCS dump formatting with SMSXDATA for DFM
- Distributed FileManager/MVS IPCS messages
- Formatting LOGREC records
- Slip trapping for a specific reply message
- GTF tracing of input/output buffers

Trace Facility for Distributed FileManager/MVS

Distributed FileManager/MVS uses the SMSX component trace support to trace events (for example, module entry, exit, post/wait, control block changes). The SMSX component trace support is controlled by the MVS/ESA TRACE operator command.

Complete syntax and usage information for TRACE can be found in *z/OS MVS System Commands*.

TRACE Command for DFM

The trace facility can be activated by the operator TRACE command, or you can use the SYS1.PARMLIB member CTncccxx to specify trace options for the TRACE CT® command and to turn off tracing.

CTncccxx must exist at IPL for tracing to be available on the system. The default is CTISMS00. For more information on using CTncccxx to specify tracing options, see ″CTncccxx″ in *z/OS MVS Initialization and Tuning Reference*.

The syntax of the TRACE command for tracing the PDSE component is:

```
TRACE      ,WTRSTART=membername [WRAP|NOWRAP ] ] ]
           CT [,WTRSTOP=jobname ] ]
           [CT [,ON ] { {COMP=SYSSMS } [,SUB=(sub[sub],...)]
           [,PARM=CTISMSxx] } ]
           [CT [,nnnK ] {                                    } ]
           [CT [,nnnnM] {                                    } ]
           [CT [,OFF] {                                      } ]
```

The default **SYSSMS** trace table size is 72 KB.

PARM=CTISMS*xx* identifies the SYS1.PARMLIB member that contains the tracing options.

The *nnn***K** and *nnnn***M** keywords can be used to specify trace table sizes of 16–999 KB or 1–2047 MB, respectively.

You are then prompted to specify trace options. You must respond with the REPLY command, using the following syntax:

```
REPLY      id
           [,JOBNAME=(jobnamelist)]
           ASID[,=(asidlist.)]
           [,OPTIONS=(namename...)]
           [,WTR=procname|DISCONNECT]
           [,END]
```

Tips:

- The *id* value is the identification number, 0–99, specified on the prompting message.
- You can list up to 16 jobs or ASIDs to be used as filters in tracing.
- The REPLY can be continued on multiple lines. END must be the final parameter, to identify the end of the REPLY.

The TRACE and REPLY commands are fully described in *z/OS MVS System Commands*.

DFM Trace Options

The valid options for use with this trace facility are:

Table 62. Valid Options for this Trace Facility

Trace Event	Description
ENTRY	All module entries.
EXIT	Module exits with nonzero return codes only.
EXITA	All module exits.
CALL	Equivalent to specifying both ENTRY and EXIT.
UEXIT	Entries and exits of user exits.
RRTN	Entries and exits of recovery routines.
CB	Control block changes.
POST	Usage of certain MVS services.
COMP=(DFM)	Specifies Distributed FileManager/MVS.
SUBCOMP=(*component- list*)	Specifies minor functional areas within the specified major areas.

The following terms are used to designate the functional areas of Distributed FileManager/MVS management:

COMP	SUBCOMP
DFM	ALL GAM GCA GCS GFL

Tips:

- You can also specify **ALL** rather than a list for both the **COMP** and **SUBCOMP** keywords.
- You can specify the **SUBCOMP** keyword without specifying the **COMP** keyword.

The following is an example of an OPTIONS reply to cause tracing of all module entries and nonzero exits for the GCS subcomponent of Distributed FileManager/MVS.

OPTIONS=(CALL,COMP(DFM),SUBCOMP(GCS))

The following command will turn off the trace facility:

Formatting the DFM Trace Table

The IPCS CTRACE command is used to format and view the information in the trace table.

The MVS DUMP command can be used to produce an SVCDUMP-type dump containing the trace buffer data. See *z/OS MVS IPCS Commands* for a complete description of this service.

IPCS Dump Formatting with SMSXDATA for DFM

The IPCS verb exit SMSXDATA can be used to format Distributed FileManager/MVS control blocks.

VERBEXIT	**SMSXDATA**	'[**COMP**(DFM) | **ALL**)] [**GLOBAL** | **NOGLOBAL**] [**MAP** | **NOMAP**] [**DFMSPACE** | **NODFMSPACE**] [**DETAIL** | **EXCEPTION**] [**CURRENT** | **ASID**(*asid*) | **JOBNAME**(*jobname*)]'

Descriptions of the keywords follow:

COMP(DFM | ALL)

DFM Specifies the Distributed FileManager/MVS component

ALL The default value. ALL selects all of the SMSX components.

For more information about SMSX component trace, see ″SMS Diagnostic Aids.″

GLOBAL | NOGLOBAL

This keyword is used to include these global control structures for Distributed FileManager/MVS :

Component	**Structures**
	BSCB, LDSB, MSM, TIAB, TITL

MAP | NOMAP

This keyword requests the formatting of a module map of the load modules.

DFMSPACE | NODFMSPACE

DFMDSP

This keyword requests formatting of all DFM data space pages. It does not format individual control blocks.

DETAIL | EXCEPTION

Definitions of these keywords follow:

DET Requests formatting of Distributed FileManager/MVS control blocks. For a figure of control blocks, see ″Distributed FileManager/MVS Control Blocks Formatted by SMSXDATA″ below.

EXC Validates all control blocks for the selected component(s) and formats *only* those control blocks that are detected to be invalid for the selected component(s). Validation consists of control block header verification.

CURRENT | ASID(*asid*) | JOBNAME(*jobname*)
These options are used as follows:

ASID(*asid-number*)
asid-number is a decimal number that represents the ASID to select for dump access of control blocks for the selected component(s).

CUR Selects each active address space (for example, dispatched on a CPU when the dump was taken) for dump access of control blocks for the selected component(s).

JOB *jobname* represents the jobname and associated ASID to select for dump access of control blocks for the selected component(s).

Table 63. Distributed FileManager/MVS Control Blocks Formatted by SMSXDATA

Control Block	Macro	Description
BACB	GDEBACB	Base Address Space Control Block
BSCB	GDEBSCB	Base System Control Block
BTCB	GDEBTCB	Base Task Control Block
EXC	GDEEXC	Exchange Server Attributes Control Block
GCCB	GDEGCCB	Communications Manager Conversation Control Block
HACB	GDEHACB	Agent Conversation Control Block
HTCB	GDEHTCB	Agent Task Control Block
LFLT	GDELFLT	Lock File Table Holder
LDSB	GDELDSB	Lock Manager Data Space Book
MOPQ	GDEMOPQ	Open PDSE Queue
MSM	GDEMSM	Stream Master Control Block
TIAB	GDETIAB	Task Invoke Anchor Block
TITL	GDETITL	Task Invoke Task List

The following is an example of an IPCS command to generate the SMSX module map for the Distributed FileManager/MVS component. This command also formats Distributed FileManager/MVS global control blocks (BSCB, LDSB, MSM, TIAB, TITL) for each active address space during time of the dump.

VERBEXIT SMSXDATA 'COMP(DFM) GLOBAL MAP DETAIL'

The following is an example of an IPCS command to format Distributed FileManager/MVS control blocks (BACB, LFLT, MSM, TIAB) for the specified address space. This command also formats the data from the Distributed FileManager/MVS data spaces.

VERBEXIT SMSXDATA 'COMP(DFM) DFMSPACE DETAIL ASID(asid)'

Formatting LOGREC Records

For system logic errors, DSS data stream errors, or unexpected APPC/MVS errors, Distributed FileManager/MVS writes records to the system LOGREC file. You can format and print the LOGREC software error records by using EREP (PGM=IFCEREP1) with the following OPTIONS parameters:

"TYPE=S,PRINT=PS'

You will get a listing of ALL the software records sorted by severity; abends are listed first. To filter the listing to a specific day, a numeric representation of the date is used with the previous option. In the following example, the day is 252 in the year 1993:

"TYPE=S,PRINT=PS,DATE=(93252)'

Slip Trapping for a Specific Reply Message

DDM reply messages are returned to the source system and include a server diagnostic area (SRVDGN). Normally, source errors do not cause a dump. SYNTAXRM and PRCCNVRM reply messages indicate DDM data stream errors; these errors might be a by-product of hardware or software (other than the end user). Accordingly, this error record is placed in the system LOGREC file. However, if it is deemed necessary to dump Distributed FileManager/MVS at the time of building a specific reply message, it is possible to set a SLIP trap. The SLIP trap forces a SVCD dump.

Because only one PER SLIP trap can be set at a time, you need to determine which Reply Message Builder is invoked (GDEHRBRM or GDEGBRMB). In the case of SYNTAXRM and PRCCNVRM, the system LOGREC record tells you the name of the invoked Reply Message Builder. Otherwise, examine the original reply message SRVDGN to determine the subcomponent/module calling for a reply message.

For a description of the mapping for the server-diagnostic-information fields, see “Distributed FileManager/MVS SRVDGN Data Items” on page 135 The first two bytes of the Distributed FileManager/MVS reason code identify the subcomponent/module that is requesting the reply message. The Communication Manager modules (GDEGxxxx) use GDEGBRMB; all other modules use GDEHRBRM. (The Communication Manager Modules are identified by subcomponent/module identification X'5516'- X'5529'.) If you are unable to examine the original SRVDGN, and it is not a SYNTAXRM or PRCCNVRM, most likely the invoked message builder is GDEHRBRM.

In order to set the trap, the indirect address of the Distributed FileManager/MVS Reply Message Builder modules must be known. The following AMBLIST utility job determines the offset of these modules that are in the GDEIGCA LPA load module:

```
//A EXEC PGM=AMBLIST
//SYSPRINT DD SYSOUT=*
//SYSLIB DD DISP=SHR,DSN=SYS1.LPALIB
//SYSIN DD *
 LISTLOAD OUTPUT=XREF,MEMBER=GDEIGCA
```

To find the correct offset module, examine the AMBLIST output listing in Figure 3 . The following excerpt from a sample AMBLIST listing shows that the offset of GDEGBRMB is X'9978' for the load module GDEIGCA. The offset of GDEHRBRM isX'434C8'.

```
LIST LOAD OUTPUT=XREF, MEMBER=GDEIGCA
                                        *****   MODULE SUMMARY    *****
   MEMBER NAME:     GDEIGCA
   LIBRARY:         SYSLIB
       NO ALIASES   **

                                 ****          ATTRIBUTES OF MODULE          ****
                        **   BIT   STATUS        BIT   STATUS        BIT   STATUS
                              0    RENT           1    REUS           1    NOT_OVLY
                              4    NOT_OL         5    BLOCK          6    EXEC
                              8    NOT_DC         9    ZERO_ORG       10   EP_ZERO
                              12   EDIT           13   NO_SYMS        14   F_LEVEL

                                        MODULE SSI:    NONE
                                        APFCODE:       00000001
                                        RMODE:         ANY
                        *****LOAD MODULE PROCESSED EITHER BY VS LINKAGE EDITOR OR BIN
                                NUMERICAL MAP AND CROSS_REFERENCE LIST OF LOAD MODU
                 CONTROL SECTION                                                ENTRY
                   LMOD LOC      NAME          LENGTH   TYPE                     LMOD  LO
                       00      GDEI2MAT          528     SD
                      528      GDEHRRTR         2D00     SD
                     3228      GDEHPBPM         6750     SD
                     9978      GDEGBRMB         1FA0     SD
                     B918      GDEGBDRM         1E38     SD                          15190
                                                                                     15190
                  discontinuous section of listing
                                                                                     15190
                                                                                     300D0
                    42368     GDEHRFND         1160     SD
                    434C8     GDEHRBRM         2780     SD
                    45C78     GDEHRDSV         1628     SD
```

Figure 3. Sample AMBLIST listing.

The next thing you need to know for the SLIP trap is the field offset; the offset indicates the type of reply message.

Note: Consult Level 2 for offsets and dependent values.

For GDEGBRMB, the macro to consult is GDEGRMB; the field is GRMB_MSG_CP. The GRMB_MSG_CP is at offset X'18' in the GDEGRMB. This field contains the DDM codepoint value: For example, SYNTAXRM is X'124C'.

The following example shows the SLIP command to cause a SVCD dump when the command enters GDEGBRMB (as in GDEIGCA,9978). The matching condition is that the input request in GDEGRMB at offset X'18', as in (1R?+4??+18), is to build a SYNTAXRM (EQ,0000124C).

SLIP SET,IF,LPAMOD=(GDEIGCA,9978),DATA=(1R?+4??+18,EQ, 0000124C),

JOBNAME=GDEDFM,ACTION=SVCD,ID=DFM1,END

For GDEHRBRM, the macro to consult is GDEHMPL. The field is HMPL_MSG_INDEX. This field does not contain the DDM codepoint, but the message index (defined in GDEHMPL, for example, HYMPL_SYNX) is constant (46, X'2E').

The following example tells the SLIP command to cause a SVCD dump when the command enters GDEHRBRM (as in GDEIGCA,434C8). The matching condition is that the input request in GDEHMPL at offset X'10', as in (1R?+4??+10), is to build a SYNTAXRM (EQ,002E)

SLIP SET,IF,LPAMOD=(GDEIGCA,434C8),DATA=(1R?+4??+10,EQ,002E),

JOBNAME=GDEDFM,ACTION=SVCD,ID=DFM2,END

GTF Tracing of Input/Output Buffers

If MVS/APPC is running and the LU is not defined as LU=OWN, VTAM® traces the input/ output buffers to see what is being exchanged between the source and target servers. GTF trace, an MVS facility, is used by VTAM to collect trace records. The VTAM trace can show buffer contents. VTAM trace can also show SNA request and response information. The following information shows VTAM commands and functions:

- Turn on the GTF trace facility:
- *S GTF.IT*
- *R xx, TRACE=RNIO,USR*
- *R xx U*

When the GTF initialization is complete, GTF tells you the name of the trace data set in message AHL906I.

- Find the VTAM LU to trace:
- *D APPC,LU,ALL*

The LU in use is shown (for example, LLUN=MF1AP001)

- Find the name of the VTAM procedure:
- *D A,L*

In the next example, the VTAM procedure is called ″VTAM2″.

- Start the VTAM procedure by tracing the APPC LU:
- *F VTAM2,TRACE,ID=MF1AP001,TYPE=BUF*

In order to view the trace records, you must stop the GTF trace.

- To stop the trace:
- *P IT*
- Format the trace under IPCS by using the following command with the GTF trace data set named as target.
- *GTFTRACE RNIO SLIP SVC USR JOBNAME(GDEDFM APPC)*

Distributed FileManager/MVS SRVDGN Data Items

The server-diagnostic-information (SRVDGN) field provides diagnostic information for DDM reply messages that are returned to the source system. The SRVDGN can be used with the other architected fields. These fields are for DDM reply messages. The other fields are used for problem determination and resolution.

For a list DDM reply messages returned by Distributed FileManager/MVS, see “DDM Reply Messages Returned by Distributed FileManager/MVS” on page 125.

Figure 4 on page 136 shows the SRVDGN format that is returned by Distributed FileManager/MVS. The SRVDGN consists of two sections: a constant section (always supplied) and a variable section (contents depend on the reported error). The constant section contains the Distributed FileManager/MVS return and reason codes. The last two bytes of the Distributed FileManager/MVS reason code are an error code. “Distributed FileManager/MVS Implementation Specific Codes” on page 136 lists the Distributed FileManager/MVS error codes and their descriptions.

The variable section of the SRVDGN is in the form of keyed data items. Data items apply to a reported error. Each data item in the SRVDGN consists of a two-byte field. This field is followed by a key field that is also two bytes. The key field is followed by the data item value. The length field only specifies (in bytes) the length of the date item value. Most data items are used only by IBM Service; however, some data items might be used by the MVS systems programmer for problem determination. These data items are documented in Table 64 on page 136.

All SRVDGN information should be supplied to IBM Service when the source programmer, MVS system programmer, or application programmer cannot solve the problem.

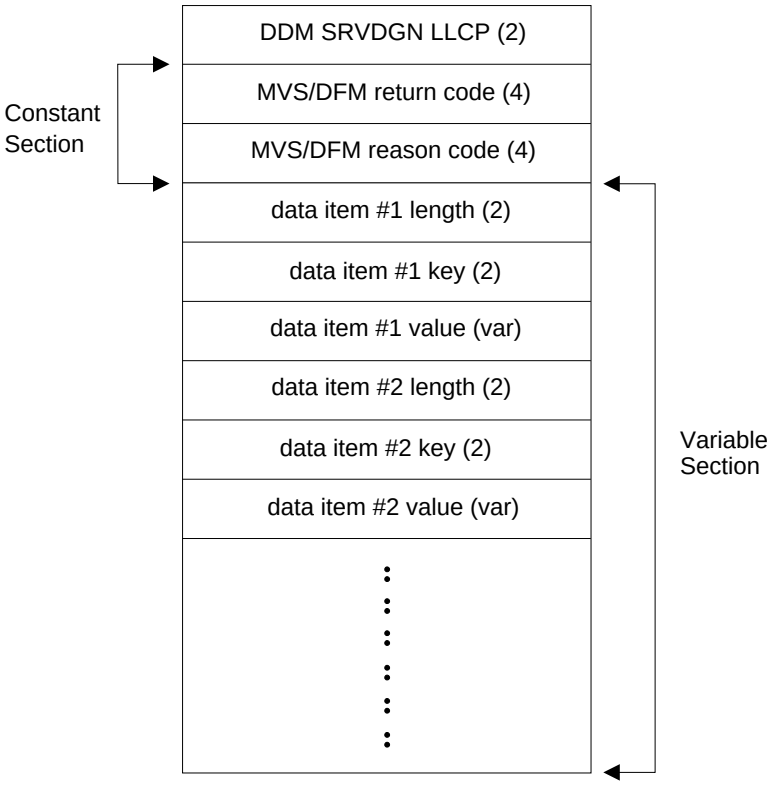


Figure 4. Server Diagnostic Area (SRVDGN) Format. Length of field is in parenthesis following field description.

Table 64. Distributed FileManager/MVS SRVDGN Data Items

Data Item Key	Data Item Length	Data Item Description
X'8428'	4	SVC 99 return code
X'8429'	4	SVC 99 reason code
X'842A'	4	SVC 99 information code
X'842F'	4	SVC 99 error reason code
X'8505'	4	APPC return code
X'8667'	4	ALESERV return code
X'8430'	4	ATTACH return code
X'842B'	4	FAMS return code
X'842C'	4	FAMS reason code

Distributed FileManager/MVS Implementation Specific Codes

The following section documents these types of codes:

- Distributed FileManager/MVS general error codes
- Distributed FileManager/MVS communications error codes
- Distributed FileManager/MVS agent error codes
- Distributed FileManager/MVS file/directory error codes
- Distributed FileManager/MVS locking error codes
- Distributed FileManager/MVS access method error codes

Distributed FileManager/MVS General Error Codes

Tables 65, 66, 67, 68, 69, and 70 list Distributed FileManager error codes.

Table 65. Distributed FileManager/MVS General Error Codes

Error Code	Description
0000	DFM_OK: Processing completed without error.
0001	DFM_SYS_LOGIC_ERR_CONV: System Logic Error at the conversation level. The conversation is terminated.
0002	DFM_SYS_LOGIC_ERR_AGNT: System Logic Error at the agent level. The agent is terminated.
0003	DFM_SYS_LOGIC_ERR_CMND: System Logic Error at the command level. The command is terminated.
0004	DFM_INVALID_FILNAM: Invalid file name specified.
0005	DFM_INVALID_DCLNAM: Invalid declared name specified.
0006	DFM_OUT_OF_PRIV_STOR: Private or extended private storage could not be obtained.
0007	DFM_PARM_NOTSUPP: A parameter supplied on the DDM command is not supported. See the CODPNT field in the associated PRMNSPRM reply message for the codepoint of the parameter that is not supported.
0008	DFM_ATTR_NOTSUPP: An attribute codepoint supplied in a file attribute list is not supported.
0009	DFM_OBJECT_NOTSUPP: A DDM command object supplied with the DDM command is not supported. See the CODPNT field in the associated OBJNSPRM reply message for the codepoint of the object that is not supported.
000A	DFM_FILE_NOT_FOUND: A file with the supplied name cannot be found in the catalog. If a member-qualified data set name was supplied, the PDSE cannot be found.
000B	DFM_MEM_NOT_FOUND: A member-qualified data set name was supplied, but the member does not exist in the PDSE. (DFM_FILE_NOT_FOUND is returned if the PDSE does not exist.)
000C	DFM_FILE_IN_USE: Access to the data set/member requested is denied because the data set/member is already in use.
000D	DFM_DUP_FILNAM: A data set/member with the supplied name already exists.
000E	DFM_NOT_AUTH_TO_FILE: The user is not authorized to access the requested data set/member.
000F	DFM_FILE_FULL: The data set/member cannot be extended.
0010	DFM_FILE_OPEN: The data set/member is already open.
0011	DFM_IO_ERROR: The file/record cannot be accessed due to an I/O error or media failure.
0012	DFM_DSTYPE_NOTSUPP: Requested function not supported for this data set/member type.
0013	DFM_MAX_INBUF: Maximum allowed number of input buffers exist; no others can be accepted at this time.
0014	DFM_MEM_ONLY: Requested function is only supported for PDSE members but was requested against a data set.
0016	DFM_OUT_OF_ECSA_STOR: Extended CSA storage could not be obtained.
0017	DFM_DCLNAM_NOT_FOUND: Declare name not found.
0018	DFM_INVALID_MEMNAM: Invalid member name specified.
0019	DFM_INVALID_FILCLS: Invalid FILCLS specified.
001A	DFM_INVALID_ACCINT: Invalid access intent specified.
001B	DFM_INVALID_ACCMTHCL: Invalid access method class specified.
001C	DFM_FILE_EMPTY: The data set/member is empty and cannot be processed by this command.

Table 65. Distributed FileManager/MVS General Error Codes (continued)

Error Code	Description
001D	DFM_INVALID_RECLEN: 1) The supplied record length is not valid for the data set/member to be created; or 2) it does not match the record length of the existing data set/member; or 3) the record length of the data set/member to be opened cannot be supported; or 4) The length of the record read exceeds the defined record length class for the data set.
001E	DFM_INVALID_RECLENCL: The supplied record length class is not valid for the data set/member to be created or it does not match the length class of the existing data set/member. Or the record length class of the data set/member to be opened is not supported.
001F	DFM_OUT_OF_DSPAC_STOR: Data space storage could not be obtained.
0020	DFM_GDEIGBS_NOT_FOUND: An entry point in load module GDEIGBS could not be found.
0021	DFM_SYS_NOT_INIT: DFM system is not initialized yet.
0022	DFM_CMD_NOTSUPP: A valid DDM command but it is not supported by MVS/DFM.
0023	DFM_AGENT_LIMIT: The maximum number of agent tasks supported has been reached; no others can be started at this time.
0024	DFM_ATTACH_ERR: An error was received from the MVS ATTACH (or ATTACHX) service. See data item with key X’8430’ in the SRVDGN for return code received from ATTACH.
0025	DFM_DSS_SYNTAX_ERR: See syntax error code (SYNERRCD) in the associated SYNTAXRM reply message.
0026	DFM_PARMVAL_NOTSUPP: A parameter value supplied in the DDM command is not supported. See the CODPNT field in the associated VALNSPRM reply message for the codepoint of the parameter whose value is not supported.
0027	DFM_FILE_TEMP_NOT_AVA: File is temporarily unavailable. Either the file is damaged and must be repaired before further use, or a target system process such as disk compression prevents immediate use.
0028	DFM_SVC99_ERR: An error was received from the MVS dynamic allocation service (SVC 99). See the following data items in the SRVDGN for more information: X’8428’ (SVC 99 return code), X’8429’ (SVC 99 reason code), X’842A’ (SVC 99 information code), X’842F’ (SVC 99 error reason code)
0029	DFM_DCLNAM_CONFLICT: The declared name is associated with a data set/member but the request is for a directory OR the declared name is associated with a directory but the request is for a data set/member.
0030	DFM_OUTBUFF_TOO_SMALL: The output buffer is too small to contain the reply information that needs to be sent.
0032	DFM_EXPIRE_DATE: An unexpired expiration date exists for the PDSE or SAM data set being deleted. This operation is not allowed.
0033	DFM_INVALID_GDGNAM: Valid or invalid GDG name was specified. GDGs are not supported.
0035	DFM_INVALID_FILNAM_LEN: Invalid file name length was specified
0036	DFM_FILE_PROTECTED: File is protected. This operation is not allowed.
0037	DFM_NON_REUSABLE_VSAM: The VSAM file is non-reusable. This operation is not allowed.
0FFE	DFM_DETACH_ERR: An error was received from the MVS DETACH service. See data item with key X'xxxx' in the SRVDGN for return code received from DETACH.
0FFF	DFM_RSN_TRANSLATE_FAIL: The translation of reason code failed.

Distributed FileManager/MVS Communications Error Codes

Table 66. Distributed FileManager/MVS Communications Error Codes

Error Code	Description
2000	APPC Send routine error. See data item with key X'5507' in SRVDGN for ATBSEND return code.
2001	APPC Send_Error routine error. See data item with key X'5507' in SRVDGN for ATBSERR return code.
2002	APPC Request_To_Send routine error. See data item with key X'5507' in SRVDGN for ATBRTS return code.
2003	APPC Prepare_To_Receive routine error. See data item with key X'5507' in SRVDGN for ATBPTR return code.
2004	APPC Receive_Immediate routine error. See data item with key X'5507' in SRVDGN for ATBRCVI return code.
2005	APPC Receive_and_Wait routine error. See data item with key X'5507' in SRVDGN for ATBRCVW return code.
2006	APPC Deallocate routine error. See data item with key X'5507' in SRVDGN for ATBDEAL return code.

Distributed FileManager/MVS Agent Error Codes

Table 67. Distributed FileManager/MVS Agent Error Codes

Error Code	Description
3000	Duplicate DCLNAM. An entry already exists in the declared name table with the specified declared name.
3001	DCLNAM in use. The declared name cannot be deleted from the declared name table because the associated data set/member is open.
3002	DDM protocol error. See conversational protocol error code (PRCCNVCD) in the associated PRCCNVRM reply message.
3003	The number of managers specified by the source in an EXCSAT command has exceeded the table size of the manager level table supported by Distributed FileManager/MVS. This could mean that the MGRLVLLS parameter value in the EXCSAT command has been corrupted.
3004	The manager level specified attempts to re-specify a manager level specified by a previous EXCSAT command.
3005	The command is not supported at the manager level specified by the previous EXCSAT command.
3006	The parameter is not supported at the manager level specified by the previous EXCSAT command.

Distributed FileManager/MVS File/Directory Error Codes

Table 68. Distributed FileManager/MVS File/Directory Error Codes

Error Code	Description
4000	DFMK_INVALID_NEWFILNM: File name supplied as the new file name in a RNMFIL request is invalid.
4001	DFMK_DIR_FUNC_NOTSUPP: Function not supported for a DDM directory file.
4002	An attempt was made to change a parameter that is not supported by MVS/DFM. Examples of non-supported parameters are FILEXNSZ, FILINISZ, and FILMAXEX.
4004	An unexpected error was returned from FAMS. See the following data items in the SRVDGN for more information: X'842B' (FAMS return code), X'842C' (FAMS reason code).
4006	The length of the data, management, or storage class name specified is greater than 8 characters.
4007	The data, management, or storage class name specified could not be found or is invalid.

Distributed FileManager/MVS Locking Error Codes

Table 69. Distributed FileManager/MVS Locking Error Codes

Error Code	Description
5000	Maximum allowed number of locks on a data set/member per agent task currently exists; no others can be established.
5001	Requester attempts to release lock(s) which he does not own.
5002	Requester attempts to release lock(s) when there is no locks to release.
5004	There is a conflict between record accessors and stream accessors.
5005	A stream accessor requested a stronger lock than they already hold.

Distributed FileManager/MVS Access Method Error Codes

Table 70. Distributed FileManager/MVS Access Method Error Codes

Error Code	Description
6001	Maximum record length supported by the MVS access method has exceeded.
6002	Cursor position already at Beginning of File (BOF).
6003	Cursor position already at End of File (EOF).
6004	Update intent has not been established for the record to be modified.
6005	File is not open OR open is lost during the processing.
6006	Duplicate record number/key. If the return code is 8, the inserted record caused a duplicate key condition which is *not* allowed. If the return code is 4, the inserted record or read contains a duplicate key in either the same or different index, where the duplicate key *is* allowed.
6007	Changing the key field of an existing record is not allowed or an attempt was made to change the length of a record when updating an ESDS.
6009	End of file has been reached in case of SETNXT/SETPLS commands. For SETPRV and SETMNS the file does not contain any record before the current cursor position. For all other commands: EOF/BOF encountered unexpectedly.
600A	An inactive record cannot be inserted into a nondelete-capable file.
600B	Requested record could not be found or is inactive.
600C	There is a conflict between the requested command and a capability of the file, such as a request to insert data into file with INSCP(FALSE) attribute.

Table 70. Distributed FileManager/MVS Access Method Error Codes (continued)

Error Code	Description
600D	1. The requested function would cause the cursor to be outside the bounds of the file; **or** 2. The cursor is invalid due to a previous FRCBFF command.
600F	Random load not supported for a DDM sequential file (FILCLS(SEQFIL)).
6010	The read completed successfully but errors were detected.
6011	A read was requested for a substream that is beyond the end of file.
6012	Full (vs. partial) key required.
6013	The length of the supplied record key is greater than the defined key length of the file.
6015	RECCNT could not be specified in RECAL prior to DDM Level 2.
6016	Maximum allowed number of stream data spaces currently exists; no others can be created.
6017	The value of KEYLLM is after the value of KEYHLM on a SETKEYLM command.
6018	No next key in sequence
6020	Number of records/inactive records supplied is less than what was specified on the RECCNT parameter.
6021	BYPINA(FALSE) is specified for a direct file.
6024	More data objects sent than were specified in the RECCNT parameter.
6025	DFM_REC_NUM_OUT_OF_BOUNDS. Records number exceeds set limits
6026	Record not available. For example, the record is on a volume that is not mounted, or the alternate index points to a nonexistent record in the base cluster.
6027	Record in use. The record is locked by another user.

Chapter 9. ISMF Diagnostic Aids

The following major diagnostic aids are provided for the Interactive Storage Management Facility (ISMF):

- A description of ISMF execution options
- An explanation of how ISMF processes abends and generates dumps
- Explanations of how to use the logging/abend control entry panel and the ERTB display panel
- A description of ISMF's use of the ISPF log
- A description of ISMF return and reason codes

ISMF Execution Options

When diagnosing failures in ISMF, the following options may make problem diagnosis easier:

1. Run with **WTP message** turned on in your TSO/E profile. This gives you the message identifier and an online symptom dump for error messages issued by components other than ISMF. *z/OS TSO/E Command Reference* explains how to set up your TSO/E profile.
2. Allocate a SYSUDUMP, SYSABEND, or SYSMDUMP so the system will provide a full dump if an abend occurs. For information about allocating a dump data set, see *z/OS MVS JCL User's Guide* and *z/OS MVS JCL Reference*.
3. Run with the ISPF log turned on.
4. Look at the abend recovery and logging options in the ISMF profile to determine the options that best suit your needs.

"Determining How ISMF Processes Abends and Generates Dumps" discusses ISMF abend recovery options and how to make the best use of the ISPF logging facility.

Determining How ISMF Processes Abends and Generates Dumps

ISMF initially supplies the values in your profile that determine the way ISMF runs during each session. The information stored in your profile controls the logging of trace and error data, the way ISMF recovers from abends, and the way background jobs are processed.

You can invoke the profile by choosing option 0, **ISMF profile**, on the ISMF primary option menu or by entering the **PROFILE** command on the command line of any ISMF panel (except the ISMF profile option menu, an abend panel, or the ISMF ERTB display panel). If you select the profile option on the primary option menu, or enter the **PROFILE** command, ISMF displays the ISMF profile option menu, as shown in Figure 5 on page 144. From the profile option menu you can view your profile parameters for logging and error recovery. You can also display the ISMF job statements or the execute statements that ISMF uses to submit background jobs.

© Copyright IBM Corp. 1979, 2002

```
  Panel  Help
-----------------------------------------------------------
DGTSPPF1                    ISMF PROFILE OPTION MENU
Enter Selection or Command ===>

Select one of the following options and Press Enter:

  0  User Mode Selection
  1  Logging and Abend Control
  2  ISMF Job Statement
  3  DFSMSdss Execute Statement
  4  ICKDSF Execute Statement
  5  Data Set Print Execute Statement
  6  IDCAMS Execute Statement
  X  Exit









Use HELP command for Help; Use END command or X to Exit.
```

Figure 5. ISMF—Profile Option Menu

Using the Logging/Abend Control Entry Panel

Selecting option 1, **LOGGING and ABEND CONTROL**, on the ISMF profile option menu, displays the panel shown in Figure 6 on page 145, which allows you to:

- Specify the amount of information ISMF records (the first three options shown on the panel in Figure 6 on page 145). This is discussed in “ISMF’s Use of the ISPF Log” on page 148.
- View or change the way ISMF handles abends (the fourth option, “RECOVER FROM ABENDS,” shown on the panel in Figure 6 on page 145). This is discussed in “Recovering from Abends” on page 147.
- Cause ISMF to automatically present the ERTB display panel (see Figure 7 on page 146) if an ISMF error record is created whose contents *exactly match* the field values entered by the user. (Blank fields *are not* compared.)

  This option is used to trap an intermittent failure. It assumes you have some failure-related data (such as the module name, return code, reason code, or feedback information). You must enter this data on the logging/abend control entry panel. For example, if the **COPY** line operator was failing intermittently with a reason code 103, you would enter the line operator name in the feedback field and the reason code in the reason code field. (See Figure 6 on page 145.) If the failure occurred again, the trap would spring and ERTB panel would display the failure-related ERTB entry.

  Note that after selecting trap conditions, you can enable or disable the trap simply by changing the **DISPLAY ERTB WHEN THE RECORD BELOW IS LOGGED** field.

```
  Panel  Utilities  Help
-------------------------------------------------------------------------------
DGTDPPF2              LOGGING/ABEND CONTROL ENTRY PANEL
Command ===>

Optionally Specify One or More:

  Log Detailed Error Data . . Y  (Y or N)
  Log Inter-module Trace  . . N  (Y or N)
  Log Trace-point Trace . . . N  (Y or N)

  Recover from Abends . . . . Y  (Y or N)

  Display Ertb when the Record Below is Logged . . N     (Y or N)
    Module Name . .                                      (8 Characters)
    Proc Name . . .                                      (1 to 8 Characters)
    Return Code . .                                      (1 to 4 Digits)
    Reason Code . .                                      (1 to 4 Digits)
    Short Msg . . .                                      (8 Characters)
    Long Msg  . . .                                      (8 Characters)
    Service . . . .                                      (1 to 8 Characters)
    Feedback  . . .                                      (1 to 35 Characters)
Use ENTER to Perform Verification;
Use HELP Command for Help; Use END Command to Save and Exit.
```

Figure 6. ISMF—Entry Panel for Logging and Abend Parameters

Note: The contents of the ERTB field are defined following Figure 7 on page 146.

Using the ERTB Display Panel

You display the ISMF error table (ERTB) panel (shown in Figure 7 on page 146) information by:

- Getting a match on the values you entered on the logging/abend control entry panel. (See “Using the Logging/Abend Control Entry Panel” on page 144.)
- Entering the **ERTB** command from any panel except:
  - The ERTB panel itself
  - The ISMF ABEND panel
  - The ISMF primary option panel
  - The profile primary option panel
  - When in PDF BROWSE/EDIT
  - RACF panels from the security line operator
  - When ISMF temporarily transfers you out of ISMF.

  This option is used to capture error information if a message or other symptom indicates that an error has occurred. Once the system displays the ERTB panel, you may preserve the information by either recording it on the keyword worksheet in *z/OS DFSMSdfp Diagnosis Guide*, or by issuing the ISPF PRINT command to save the screen image in the ISPF list data set. You can browse or print this data set for later use.

The information displayed on the ERTB panel provides failure symptoms that you can specify as search keywords. Figure 7 on page 146 shows a sample panel. The panel displays three ERTB entries at a time. The entry initially on the right is the current (error) entry. You may use UP/DOWN commands to view other ERTB entries. A maximum of fifteen entries can be viewed before the display wraps around. Note that the right-most entry displayed in Figure 7 is the one that matches the values selected in Figure 6.

```
  Panel  Help
-------------------------------------------------------------------------------
DGTIERTB                   ISMF ERROR TABLE DISPLAY
Command ===>

ISMF Ertb Address . . : 046C51B4                            User : USER5
Error Entry Address . : 046C5310                            Date : 95/08/03
Last Panel Displayed  : DGTLGP11                            Time : 17:51
Entry : 046C5260
----------------------    Entry : 046C52B8
Module Name: DGTFARP2     ----------------------     Entry : 046C5310
Proc Name  : DGTFARP2     Module Name : DGTFARP2     ----------------------
Return Code: 0008         Proc Name . : DGTFARP2     Module Name: DGTFARF3
Reason Code: 0696         Return Code : 0008         Proc Name  : SCNROWOR
Short Msg  :              Reason Code : 0696         Return Code: 0008
Long Msg . :              Short Msg . :              Reason Code: 0701
Service  . : TBSKIP       Long Msg  . :              Short Msg  :
Feedback . :              Service . . : TBSKIP       Long Msg . :
                          Feedback  . :              Service  . : TBSCAN
                                                     Feedback . :


Use ENTER to Review Displayed Entries; Use UP/DOWN Command for other Entries;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 7. ISMF—ERTB Display Panel

The fields contain the following information:

Field Name	Description
ISMF ERTB ADDRESS	The address of the ERTB itself. UP/DOWN commands do not affect the contents of this field.
ERROR ENTRY ADDRESS	The address of the ERTB entry which represents the current error. Because of variations in environmental data, you must exercise judgment before considering this entry to be the desired entry. UP/DOWN commands do not affect the contents of this field.
LAST PANEL DISPLAYED	The panel ID of the panel last displayed for the dialog being executed. In ISPF split screen mode, each ISMF session/panel has its own ERTB. Therefore, an ISMF session in the upper screen runs with a different ERTB than the session in the lower screen.
USER	The TSO/E user ID.
DATE	The current date.
TIME	The time (updated when ENTER is pressed).
ENTRY	The address of the ERTB entry shown below it. As you use UP/DOWN commands, the entries move right or left one entry at a time.
MODULE NAME	The failure-related module name.
PROC NAME	The failure-related procedure name.
RETURN CODE	The return code.
REASON CODE	The reason code.

SHORT MSG	The message ID for the short message.
LONG MSG	The message ID for the long message.
SERVICE	The name of the failing external service.
FEEDBACK	The Problem Determination Area (PDA) or feedback information from any service.

Recovering from Abends

ISMF responds to abend conditions based on both the value (**Y** or **N**) you specify in the **RECOVER FROM ABENDS** field in your profile, and the way you invoke ISPF (in normal or test/trace mode). The value in the **RECOVER FROM ABENDS** field on the Logging/Abend Control Entry Panel determines whether ISMF retains control after an abend occurs, and the type of error related information that is recorded. The ISPF mode determines where control will be passed (to ISPF or TSO/E) when ISMF does not recover from abends.

If you specify

```
RECOVER FROM ABENDS     ===> Y
```

ISMF attempts to recover from the abend and retain control. When an abend occurs, an ISMF symptom dump at the time of the failure is placed in the ISPF log. In many cases the symptom dump is sufficient to diagnose the problem. See Figure 8 for an example of the format. The ISMF symptom dump contains the following information:

- The application involved in the failure
- The CDS name (if applicable)
- The function involved in the failure
- The current release level and function modification identifier (FMID) for ISMF
- The abend code
- The name of the module involved in the failure, if it is available
- The name of the CSECT involved in the failure
- The entry point address (EPA) of the failing module
- The program status word (PSW)
- The 16 bytes of data around the failing instruction
- The contents of the registers when the abend occurred

Exceptions: ISMF cannot recover from user abends 111, 222, 989, 997, or 999. ISPF uses these abend codes for certain unrecoverable errors. If you add functions to ISMF, you must avoid using these abend codes.

```
14:32   *****  ISMF ERROR  ***** - APPLICATION(DGTA - OPTLIB); FUNCTION(DEFINE)
14:32                            - WORK.SCDS
14:32                            - ISMF 3.3.1   FMID JDP3331   SYSTEM ABEND CODE 0C1
14:32                            - CSECT NAME DGTFLCDE   EPA 8260D5F8   PSW 078D2000 8000002A
14:32                            - DATA AT PSW 00000022 - 1000822E 3488078D 20008262 631E0000
14:32                            - R0  00000000   R1  0270A554   R2  026E6618   R3  000535E0
14:32                            - R4  00041698   R5  00000000   R6  000535E0   R7  00056800
14:32                            - R8  0270A56C   R9  026F98E7   R10 026E813C   R11 0270A4CC
14:32                            - R12 8260D5F8   R13 0270A4CC   R14 8260D750   R15 00000000
```

Figure 8. ISMF—Symptom Dump in the ISPF Log

Passing Control to ISPF or TSO/E

If you specify

```
RECOVER FROM ABENDS     ===> N
```

ISMF does not attempt to retain control when an abend occurs. The type of dump that is taken depends on the ISPF mode.

If You Invoked ISPF in Normal Mode

ISMF passes control to ISPF. The ISPF primary option menu or the ISPF master application menu is displayed and an ISPF symptom dump is placed in the log. See Figure 9 for a sample ISPF symptom dump. The ISPF symptom dump consists of:

- The release level of ISPF
- The abend code
- The program status word (PSW)
- The name of the module involved in the failure, if it is available
- The entry point address (EPA) of the module involved in the failure
- The entry point address (EPA) of the ISPF ISPSUBS and ISPTBLS modules
- The contents of the registers when the abend occurred

```
14:45      ISP SUBTASK ABEND     - VS 03.8 ISP 2.3.0000  ABEND CODE = 00007A PSW  FF85000D 00000000
14:45                            - NAME  EPA 01C8DB00  ISPSUBS EPA 81C01F88 ISPTBLS EPA 000236B0
14:45      REGISTERS AT          - R0  80000000  R1  8000007A  R2  01CE2668  R3  00000001
14:45      ENTRY TO              - R4  00000001  R5  01CC0D28  R6  00027CB8  R7  00030B10
14:45      ABEND                 - R8  00000000  R9  0002E7A0  R10 01C6A304  R11 01CE3B24
14:45                            - R12 80036534  R13 0002E7A0  R14 8003675C  R15 00000000
```

Figure 9. ISPF—Symptom Dump in the ISPF Log

If You Invoked ISPF in Test/Trace Mode

ISPF passes control to TSO/E. A TSO/E error message is displayed and, if a SYSUDUMP, SYSABEND, or SYSMDUMP data set was preallocated, a dump is taken. To obtain more detailed failure-related information, see the section that describes TSO/E messages for ISMF abends in *z/OS DFSMSdfp Diagnosis Guide*. For more information on allocating a dump data set and the dump format, see *z/OS MVS JCL User's Guide* and *z/OS MVS JCL Reference*.

Note: Test mode is set automatically any time you invoke the ISPF/PDF dialog test facility. It remains set until you exit from ISPF.

ISMF's Use of the ISPF Log

ISMF uses the ISPF log to record information about the execution of line operators, commands, and error conditions that it detects during processing. Used with the information recorded for TSO/E and ISPF tasks, the entries made by ISMF enable you to keep track of the tasks you perform in a given ISPF or ISMF session. The entries can also assist in diagnosing errors. To ensure that logging is done, you must specify a primary allocation greater than zero in the ISPF log parameters. If you specify zero as the primary allocation, no information about your session appears in the ISPF log. For more information about specifying ISPF log parameters, see the *z/OS ISPF User's Guide*. ISMF records a standard set of information during each session. Entering **Y** in any of the fields that control logging on the Logging/Abend Control Entry Panel determines what additional information is recorded in the ISPF log for each ISMF session. You can choose to log more detailed data about error conditions. You can also log inter-module and trace-point trace records.

Standard Log Information

The standard information that appears in the log consists of the following:

For line operators performed in the foreground:

- The name of the line operator
- The list entry for which the line operator was used (for example **USER19.FUNLIB.TEXT**)
- The ISMF application used (for example, data set application)
- A completion message.

For line operators or list commands that have been submitted successfully:

- The name of the line operator or list command
- The list entry for which the line operator was used
- The number of entries for which the list command was used
- The ISMF application
- The name of the job
- The job number or name of the job data set
- A status message.

For error conditions:

- The ISMF application that was being used when the error occurred
- The function that was being performed (for example, **FILTER**)
- The list entry associated with a line operator in error, if applicable
- The number of list entries associated with a list command in error
- The return code and reason code for the error.

Logging of Detailed Error Data

Entering **Y** for the first option, **LOG DETAILED ERROR DATA**, on the Logging/Abend Control Entry Panel gives you the following additional information for error conditions:

- The names of the ISMF module and procedure that detected the error
- The ID of the error message associated with the failure and the ID of the last panel displayed before the error
- The name of the external service that failed and any error information that the service returned.

Figure 10 is an example of the ISPF log with detailed logging for errors.

```
15:01   *****  ISMF ERROR  ***** - APPLICATION(DGT2 - VOLUME); FUNCTION(SELECT)
15:01                            - RETURN CODE(0016); REASON CODE(1023)
15:01                            - MODULE(DGTFDA41); PROCEDURE(DEVINFO)
15:01                            - MESSAGE ID( - ); LAST PANEL(DGTDVVA1)
15:01                            - SERVICE(DEVINFO); FEEDBACK(10,038,IGBDIS01,1674,)
15:03   *****  ISMF ERROR  ***** - APPLICATION(DGT2 - VOLUME); FUNCTION(SELECT)
15:03                            - RETURN CODE(0008); REASON CODE(0006)
15:03                            - MODULE(DGTFVA03); PROCEDURE(GENAARAY)
15:03                            - MESSAGE ID(DGTDS011 - DGTDS011); LAST PANEL(DGTDVVA1)
15:03                            - SERVICE(DGTFVA04); FEEDBACK(NONE)
15:03   *****  ISMF ERROR  ***** - APPLICATION(DGT1 - DATA SET); FUNCTION(DISPLIST)
15:03                            - RETURN CODE(0008); REASON CODE(0102)
15:03                            - MODULE(DGTFCTCK); PROCEDURE(DCGFCTCK)
15:03                            - MESSAGE ID(DGTCT001 - DGTCT001); LAST PANEL(DGTLDDS1)
15:03                            - SERVICE(NONE); FEEDBACK(EDIT)
```

Figure 10. ISMF—Detailed Error Entries in the ISPF Log

Inter-Module Traces

Entering **Y** for the second option, **LOG INTER-MODULE TRACE**, gives you a log record for every module that executes during an ISMF session. This information can help you to diagnose loops, or determine module flow when there is more than one possible path to an error. The log record for an inter-module trace consists of:

- The name of the module
- The contents of register 1 and register 11 when the trace entry was made
- Any additional trace data that was recorded.

Figure 11 is an example of the ISPF log with ISMF inter-module trace entries.

```
14:58   *****  ISMF ERROR  ***** - APPLICATION(DGT1 - DATA SET); FUNCTION(DISPLIST)
14:58                            - RETURN CODE(0008); REASON CODE(0102)
14:58                            - MODULE(DGTFCTCK); PROCEDURE(DGTFCTCK)
14:58                            - MESSAGE ID(DGTCT001 - DGTCT001); LAST PANEL(DGTLDDS1)
14:58                            - SERVICE(NONE); FEEDBACK(PROFIEL)
14:58      ISMF    - TRACE       - MODULE ID(DGTFDS21); REG 1(01C9D02E);  REG 11(01CE1374)
14:58                            - TRACE DATA()
14:58      ISMF    - TRACE       - MODULE ID(DGTFDS18); REG 1(01C9C317);  REG 11(01CE1334)
14:58                            - TRACE DATA()
14:58      ISMF    - TRACE       - MODULE ID(DGTFFOP1); REG 1(01CD01F6);  REG 11(01CE1324)
14:58                            - TRACE DATA()
14:58      ISMF    - TRACE       - MODULE ID(DGTFCTSE); REG 1(01CC1216);  REG 11(01CE137C)
14:58                            - TRACE DATA()
14:58      ISMF    - TRACE       - MODULE ID(DGTFFOX1); REG 1(01CDB64E);  REG 11(01CE1254)
14:58                            - TRACE DATA()
14:58      ISMF    - TRACE       - MODULE ID(DGTFDS16); REG 1(01C9A8AB);  REG 11(01CE2264)
14:58                            - TRACE DATA()
14:58      ISMF    - TRACE       - MODULE ID(DGTFFOQ1); REG 1(01CD6626);  REG 11(01CE1354)
14:58                            - TRACE DATA()
14:58      ISMF    - TRACE       - MODULE ID(DGTFLPCK); REG 1(01CA27E6);  REG 11(01CE1354)
14:58                            - TRACE DATA()
14:58      ISMF    - TRACE       - MODULE ID(DGTFFOG1); REG 1(01CC2136);  REG 11(01CE1234)
14:58                            - TRACE DATA()
14:58      ISMF    - TRACE       - MODULE ID(DGTFCTCK); REG 1(01CC2606);  REG 11(01CE12B4)
14:58                            - TRACE DATA()
14:58      ISMF    - TRACE       - MODULE ID(DGTFLPPR); REG 1(01CA395F);  REG 11(01CE121C)
14:58                            - TRACE DATA()
```

Figure 11. ISMF—ISPF Log with ISMF Inter-Module Trace Entries

Trace-Point Traces

Entering **Y** for the third option, **LOG TRACE-POINT TRACE**, gives you additional trace data from pre-established points in ISMF modules or functions. Used with the inter-module trace, this data provides detailed information about the flow of ISMF modules and functions. The log record consists of:

- The ID of the trace point
- The contents of register 1 and register 11 when the trace entry was made
- Any additional trace data that was recorded.

Figure 12 is an example of the ISPF log with ISMF trace-point entries.

```
ISMF    - TRACE      - POINT ID(DGTFCTFT); REG 1(01C8F720); REG 11(01CD5084)
                     - TRACE DATA(VGET SERVICE......................)
```

Figure 12. ISMF—ISPF Log with ISMF Trace-Point Entries

Logging of OAM and DB2® Errors

If OAM or DB2 issues a nonzero return code, ISMF will report it in the feedback area of the log entry. The entry will contain a return code and a reason code.

For OAM errors, see “Chapter 11. OAM Diagnostic Aids” on page 195.

For DB2 errors, see *DB2 Messages and Codes*.

ISMF Return Codes

When one ISMF function calls another, the return code from the called function is placed in register 15. Table 71 lists the standard return codes that are used by ISMF. Some of the individual interfaces may define return codes differently.

Table 71. ISMF—Standard Return Codes

Return Code	Description
0 (X'00')	Successful.
4 (X'04')	Function performed with minor error.
8 (X'08')	Function not performed; no termination.
12 (X'0C')	Function not performed; terminating error.
16 (X'10')	Function not performed; terminating error.
20 (X'14')	Function not performed; terminating error.

ISMF Reason Codes

Reason Code	Description
0001	DEVINFO: Unable to complete request with the return area provided. Return code: 4.
0002	DEVINFO: Some of the returned entries have nonzero response codes. Return code: 8.
0003	DEVINFO: Caller provided a bad parameter in the parameter list. Return code: 12.
0004	DEVINFO: Internal error was encountered. Return code: 16.
0005	Unexpected return code (DGTCDSAP) or (DGTCVAAP) came from routines called by the data set or volume application component.
0006	Error(s) occurred during previous processing. Check the previous log entries for detailed error information.
0007	List source is invalid. It should be either VTOC or CATALOG.
0008	TRKCALC macro failed. Return code: 4.
0009	Automatic data area is in use. An ISMF common service routine was entered and the frozen storage for its automatic data area was already in use.
0010	Selection criteria is invalid and will produce an empty list.
0011	Filter criteria is invalid and will produce an empty list.
0012	Prolog failure: Storage for the invoking module’s automatic data area was not obtained.

0013 Epilog failure: Invoking module's automatic data area could not be freed.

0014 Panel number is invalid. A valid panel number is 1, 2, or 3.

0015 Invalid command is not detected by the command check routine.

0016 Convert Utility: Target data length must be 4 bytes or less when converting a character string to its fixed binary representation.

0017 Convert Utility: Invalid character was found in the data to be converted. Data conversion could not take place. In some cases, imbedded blanks are invalid.

0018 Desired DAB was not found in the DAB list.

0019 Data to be converted contained all blanks and could not be converted.

0020 TBADD: Severe table error. Return code: 20.

0021 TBADD: Table is not open. Return code: 12.

0022 TBADD: Numeric convert error. Return code: 16.

0023 TBCREATE: Severe table error. Return code: 20.

0024 TBCREATE: Table is in use and ENQ failed. Return code: 12.

0025 TBCREATE: Table already exists. REPLACE was not specified. Return code: 8.

0026 TBDELETE: Severe table error. Return code: 20.

0027 TBDELETE: Table is not open. Return code: 12.

0028 TBDELETE: CRP was at top (0) and remains at top. Return code: 8.

0029 TBEND: Severe table error. Return code: 20.

0030 TBEND: Table is not open. Return code: 12.

0031 TBGET: Severe table error. Return code: 20.

0032 TBGET: Table is not open. Return code: 12.

0033 TBGET: Variable value was truncated, or insufficient space was provided to return all extension variable names. Return code:16.

0034 TBGET: CRP was at top (0) and remains at top. Return code: 8.

0035 Convert utility: Source data length must be 2 bytes when converting data set organization from the DSCB format to the list format.

0036 TBSCAN: Severe table error. Return code: 20.

0037 TBSCAN: Table is not open. Return code: 12.

0038 TBSCAN: Row does not exist. No match was found and CRP is set to top (0). Return code: 8.

0039 TBSCAN: Variable value was truncated, or insufficient space was provided to return all extension variable names. Return code: 16.

0040 Convert utility: Target length must be at least 3 bytes to hold the converted DSORG value.

0041 Convert utility: Source data length must be 1 byte when converting allocation unit from the DSCB format to the ISMF list format.

0042 Convert utility: Target data length must be at least 3 bytes to hold the converted allocation unit.

0043 Convert utility: Allocation unit could not be determined from the input.

0044 Convert utility: Source data length for converting record format is invalid. It must be 1 byte.

0045 Convert utility: Target data length must be at least 5 bytes to hold the converted record format.

0046 Convert utility: Record format could not be determined from the input.

0047 Convert utility: Source data length must be 1 byte when converting the allocation unit from the catalog format to the ISMF list format.

0048 Convert utility: Source data length must be at least 3 bytes to convert the allocation unit into the corresponding bit mapping.

0049 Convert utility: Target data length must be at least 1 byte to convert the allocation unit into the corresponding bit mapping.

0050 Convert utility: Source data length must be 4 bytes when converting device type from the UCB to the ISMF list format.

0051 RETRY routine is in control.

0052 No storage was allocated. Return code: 4.

0053 Real storage is not available to back request. Return code: 8.

0054 Unexpected error occurred while processing ESTAE. Return code: 16.

0055 ESTAE macro was unable to obtain storage for an SCB. Return code: 20.

0056 Level of DFDSS was not returned by DGTFMDO3.

0057 Level of DFHSM was not returned by DGTFMDO4.

0058 Not all storage is freed. Return code: 4.

0059 Delete request was not issued. Return code: 4.

0060 Level of DFDSS could not be determined by the DFDSS ADRMCLVL macro. X'04' was passed back in the high order byte in register 1 by the ADRMCLVL macro, indicating that it was unable to determine the current level of DFDSS.

0061 LOAD macro failed. Control was given to the error routine specified in the LOAD macro invocation. Return code: 4.

0062 ISPF select failed. Specified panel could not be found. Return code: 12.

0063 Unable to build the error table (ERTB).

0064 No exits exist for this TCB. Return code: 12.

0065 ISPF SETMSG failed. Panel message not found or syntax is invalid. Return code: 12.

0066 ISPF SETMSG service failed. Severe error occurred in processing ISPF SETMSG service. Return code: 20.

0067 ISPF log failed. Log message invalid or not found. Return code: 12.

0068 ISPF log failed. Severe error in ISPF log service. Return code: 20.

0069 Unable to build the inter-module trace table (IMTT).

0070 Unable to build the load list block (LLBL).

0071 Unable to build the storage control table (SCT).

0072 Unable to build the service routine vector table (SRVT).

0073 Unable to build the trace point trace table (TPTT).

0074 ISPF select failed. Truncation error in storing the ZCMD or ZSEL variable. Return code: 16.

0075 ISPF select failed. Severe error. Return code: 20.

0076 Invalid return code received from ISPF select.

0077 Invalid logging type was passed to the ISMF log utility.

0078 Convert utility: Target data length must be at least 7 bytes when converting the device type from the UCB into the ISMF list format.

0079 Convert utility: Device type could not be determined from the data.

0080 Convert utility: Source data length must be 7 bytes when converting the device type from the ISMF list form to the UCB form.

0081 Convert utility: Target data length must be 4 bytes when converting the device type from the ISMF list form to the UCB form.

0082 Convert utility: Source data length must be 1 byte when converting the DSCB change indicator bit into an EBCDIC character string .

0083 Convert utility: Target data length must be at least 3 bytes when converting the DSCB change indicator bit into an EBCDIC character string.

0084 Convert utility: Source data length for converting the option code field of a DSCB (DS1OPTCD) into a character string must be 1 byte.

0085 Convert utility: Target data length for converting the option code field of a DSCB (DS1OPTCD) into a character string must be at least 1 byte.

0086 Convert utility: Source data length must be 4 bytes when converting a fullword fixed binary number representing bytes into the corresponding kilobyte representation.

0087 Convert utility: Source data length must be 3 bytes when converting a fixed binary date into its character string representation.

0088 Convert utility: Target data length must be at least 8 bytes to hold a converted date in its character string format, 'YY/MM/DD'.

0089 Convert utility: Specified source index is invalid.

0090 Convert utility: Specified target index is invalid.

0091 Convert utility: Invalid year was found when performing a date conversion.

0092 Convert utility: Invalid month was found when performing a date conversion.

0093 Convert utility: Invalid day was found when performing a date conversion.

0094 Convert utility: Source data length must be 8 bytes when converting a date in the character string format, 'YY/MM/DD', into a fixed packed decimal format, 'YYDDDS', where S is the sign digit.

0095 Convert utility: Target data length must be 3 bytes when converting a date in the character string format, 'YY/MM/DD', into a fixed packed decimal format, 'YYDDDS', where S is the sign digit.

0096 Convert utility: Source data length can be a maximum of 8 bytes when converting a character string to its fixed binary representation.

0097 Convert utility: Source data length must be 3 bytes when converting a date in packed decimal format to its character string representation of 'YY/MM/DD'.

0098 Link SVC failed.

0099 Convert utility: Undetermined data set organization.

0100 Convert utility: Unable to load the conversion translate table (CTTB).

0101 Invalid scroll amount, the valid format is: P (page); H (half); D (data); M (maximum); C (cursor); NNNN (a number from 1 to 9999).

0102 Command not found in the application command table.

0103 One or more commands not found in the application command table.

0104 Command processor parameter list pointer=0.

0105 Application command table pointer=0.

0106 Command table anchor block pointer=0.

0107 Array name pointer=0.

0108 Format name pointer=0.

0109 Command line pointer=0.

0110 Unexpected error occurred in external routine.

0111 Command line length=0.

0112 Command length is invalid.

0113 Command specified is in disable status in the application command table.

0114 Panel type is invalid. Function only supports a list panel.

0115 Command/line operator control blocks chain is not 0 before storage is freed.

0116 Application ID is invalid.

0117 No application table is being loaded.

0118 When the hide pending mode is on, no list command is allowed.

0119 Termination type is invalid.

0120 Command table initialization failed.

0121 Command table entry count=0.

0122 Error occurred while command processor was trying to process the error routine.

0123 CTABCLCB is not equal to CPPLCLCB.

0124 Nonzero return code came from ISPF control service.

0125 Reserved for command processor.

0126 Invalid storage size passed to the get-memory routine.

0127 Invalid storage ID passed to the get-memory routine.

0128 Unexpected column name found in DAB. DABs may be out of order.

0129 Invalid cursor control block chained from the command processor parameter list (CPPL).

0130 Invoker's APID does not match current CTABAPID.

0131 Invoker's APID was not found in the CTAB chain.

0132 Two CTABs have duplicate APIDs that match the invoker's APID.

0133 Nonzero return code was passed back from the invoked CLIST.

0134 Specified CLIST name was too long. CLIST name must be 1 to 8 characters long (optionally preceded by '%').

0135 ISPF/PDF edit failed. Specified member was not found. Return code: 14.

0136 ISPF/PDF edit failed. Data was zero length (empty sequential data set or zero length member of a partitioned data set). Return code: 12.

0137 Nonzero return code was passed back from the specified CLIST.

0138 Invalid system completion code was passed in the retry work area.

0139 Incorrect number of parameters was specified.

0140 Member name specified is too long.

0141 TBSAVE: Table is not open. Return code: 12.

0142 TBSAVE: Table output library is not open. Return code: 16.

0143 TBSAVE: Severe error. Return code: 20.

0144 TBSAVE: Unexpected return code.

0188 Convert utility: The target data length must be at least 10 bytes to hold a converted date in its character string format YYYY/MM/DD.

0189 Convert utility: An invalid century and year in the format YYYY was found when performing a date conversion.

0190 Convert utility: An invalid century was found when performing a date conversion.

0191 Convert utility: An invalid century was found when performing a date conversion.

0197 Convert utility: The source data length must be 4 bytes when converting a date in packed decimal format to its character string representation of YYYY/MM/DD.

0200 Line operator anchor block pointer=0.

0201 List command and line operator conflict.

0202 Line operator was not found in the application line operator table.

0203 Entry in the line operator field is greater than 10 characters.

0204 Record format is not F or FB.

0205 Program failed because of a failure in the invoked external routine. Refer to the failed external routine problem determination area.

0206 Application line operator table pointer=0.

0207 Line operator processor work area pointer=0.

0208 Invalid return code from ISPF/PDF browse service.

0209 Duplicate table entry found while loading the table entries. Only one of the duplicates is loaded.

0210 Line operator initialization failed.

0211 Line operator processor failed due to multiple errors.

0212 Operator line length is invalid.

0213 Line operator field length=0.

0214 DGTFLPP3 rewrite the saved ERTB entry so that the error message can be displayed on screen.

0215 TOTLINES field is zero.

0216 Line operator does not have 'H' in column one but is asking DGTFLPP2 to check whether it is a hide line operator.

0217 LPP2ALL and LPP2HI flags are off. No checking or verification is done by this program.

0218 Return code from DGTEFOFI is zero, but no value returned to entry number filed. Invalid combination of return code and value returned.

0219 Caller’s ID is invalid. Valid IDs are 'CT', 'LP'.

0220 Invalid value in LOOPCNT.

0224 CSRFLAG value is not 1 and not -1. It is an invalid and unexpected value.

0225 Invalid return code came from ISPF/PDF edit.

0226 ISPF control service failed.

0227 Invalid value specified.

0230 No invocation is set for the hide line operator. LPPLHIDE is off.

0231 Hide amount is not numeric.

0232 LAI address (LPPLLAIA) is 0.

0233 Line operator does not have 'H' at column one and is not a hide line operator.

0234 No entry in the list area image to be processed. LPPLLAIT is less than 1.

0235 Hide range exceeds allowable maximum of 99999.

0236 LPPLENT is less than 1.

0237 Hide amount overlaps another pending line operator.

0238 HIDEAMT=0.

0239 HIDEAMT is not in the allowable range, 1 to 99999.

0240 LPPLENT is greater than LPPLLAIT.

0241 CSRFLAG is off when there is entry in the list area image.

0242 LPPLROWA=0.

0243 LPCBENO=0.

0244 Entry number in LPAB (LPCBENO) is greater than the size of the LAI.

0245 LPCB address is 0.

0246 Invalid value returned from format service. Valid values are 'Y' or 'N'. NAE of the format service can be found in the ISPF log.

0250 Invalid FIND parameter: ISMF user did not enter a column tag.

0251 Invalid command parameter: Only one parameter is accepted for this command.

0254 Invalid cursor position: Cursor is placed between the data columns in the list area. This is invalid for scrolling with the cursor position.

0255 Value of the scroll variable is invalid.

0256 Area name from LPCB is invalid.

0301 Format service anchor block (FSAB) does not exist. Pointer to FSAB in the global data repository block (GDRB) is 0.

0302 Format anchor block (FAB) was not found. No match of format name and array name exists in a FAB.

0303 Control block structure with the same format and array name already exists.

0304 Variables of list area (VLA) are invalid. VLA may be incomplete.

0305 Input parameter number of entries (ENTNUM) is invalid. Value is greater than the total number of entries in the LAI or is a negative number.

0306 Input parameter multiline (MULLINE) is invalid. Value of MULLINE is 2, but this is a single line entry.

0307 Input parameter number of entries in the input array (INPENT) is invalid. Number of entries specified in INPENT is not equal to the number of entries actually in the LAI.

0308 Data column descriptor (DCD) for corresponding data attribute block (DAB) does not exist.

0309 Input parameter area is invalid. Area name is not one of the following: input, object, data, or command.

0310 Input parameter offset is invalid. Offset is out of range.

0311 Data attribute block pointer (DABLPTR) does not point to a valid DABL.

0312 FDBLEFT or FDBRITE in the format data block (FDB) is invalid. Value is less than 0 or greater than the number of displayable data columns.

0314 Data column is too wide to be put in the data area.

0315 Invalid input parameter ADDRFABOR DCDNO in module DGTFFOX1.

0316 ISPF/PDF browse failed. Zero length data (empty sequential data set or 0 length member of a partitioned data set). Return code: 12.

0317 ISPF/PDF browse failed. Specified member not found. Return code: 14.

0318 Input parameter column name list (COLLIST) is invalid. At least one of the column names in the COLLIST was not found.

0319 Column tag specified by the user is invalid.

0320 Column tag specified is not displayed.

0321 Entry identified by the ROWID is not displayed.

0322 ISPF/PDF browse failed. No members were in library when the member list was requested. Return code: 16.

0323 Cursor is not placed in a field of the LIST panel.

0324 Cursor is placed in one of the fields: fixed area (entry statement, data column statement) or warning.

0325 Cursor is placed in a gap between 2 data columns.

0326 Cursor is placed in a field of the LIST panel, but it is not one of the following: command line, entry statement, data column statement, warning, column headings, tags, or LAI.

0327 ISPF/PDF browse failed. Severe error, unable to continue. Return code: 20.

0328 ISPF/PDF edit failed. No members were in the library when member list was requested. Return code: 16.

0329 Value of FDBVSIZE (the vertical size of the list area of the current display with factor of split screen considered) in module DGTFFOF1 is invalid.

0330 Value in the extension variable from the variables of list area (VLA) is invalid.

0331 Value of MAXL (length of variables NAME1 and NAME2) in module DGTFFOA2 is invalid.

0332 Incorrect number in the entry statement of the fixed area.

0333 TAGLEN value in module column DGTFFOI1 is invalid.

0334 No left parenthesis in the tag text.

0335 Length of the output variable is not equal to the length of the data to be retrieved.

0336 Value of COLSIZE in module DGTFFOY1 is invalid.

0337 Length of the static load module text is invalid.

0338 Input parameter area option (AREAOP) is invalid.

0339 One of the control block lengths in the static load module is incorrect.

0340 Invalid return code.

0341 Input parameter ADDRFAB is invalid.

0342 Input parameter CRPLAI or CRPVAL is invalid.

0343 FSAB exists without any FAB. Format services were not cleaned up.

0344 Character scroll amount in DGTFFOM1 is invalid. Amount is neither PAGE nor MAX.

0345 ISPF/PDF edit failed. Severe error, unable to continue. Return code: 20.

0346 CLIST line operator canceled by user via END command.

0347 Object name (second column of list) is blank.

0348 DELETE failed because the list was generated from VTOC.

0349 Invalid column heading length. The column heading length obtained from the message library is less than that stored in the static text, or greater than the length of the data area in the list panel.

0350 DFHSM line operator failed.

0351 DFQFMACC failed to read MCVT, SVC error.

0352 DFQFMACC failed to read MCVT, MWE error.

0353 DFQFMACC failed to read BCDS (B Record), SVC error.

0354 DFQFMACC failed to read BCDS (B Record), MWE error.

0355 DFQFMACC failed to read BCDS (C Record), SVC error.

0356 DFQFMACC failed to read BCDS (C Record), MWE error.

0357 DFQFMACC failed to read MCDS, SVC error.

0358 DFQFMACC failed to read MCDS, MWE error.

0359 DFSMShsm™ command entry/confirmation panel canceled.

0360 Unsupported DFSMShsm level.

0361 DFSMShsm not installed.

0362 DFSMShsm not active.

0363 The user is not a storage administrator.

0366 TSO DELETE: Minor error encountered. Return code: 4.

0367 TSO DELETE: Function completed, but specific details were bypassed. Return code: 8.

0368 TSO DELETE: Entire function could not be performed. Return code: 12.

0369 TSO DELETE: Severe error. Return code: 16.

0370 TSO DELETE: Severe error. Return code: 20.

0373 Data set name is unreadable '?????????'. Unable to alter data set.

0374 ALTER line-op failed: severe error in DGTFAL04, or in DGTQAL01. Unable to alter data set.

0375 ALTER line-op failed: severe error in DGTFAL05. Unable to alter data set.

0400 Sort parameter is invalid.

0401 Filter syntax is invalid. Only one set of filter criteria may be specified.

0402 Application filter-initialization module failed. Application did not initialize appropriate filter control blocks.

0403 FILTER command was canceled by the user with the END command. No filtering took place.

0404 No changes were detected in the filter criteria. List is already filtered according to the specified filter criteria. No new filter took place.

0405 Filter flags not properly initialized. One of the following flags must be on: ADDFLAG, DELFLAG, or MODFLAG.

0406 Invalid filter type specified by the user, detected by the filter module.

0407 Invalid relational operator specified by the user and not detected by the filter panel definition. EQ or NE should have been specified.

0408 Invalid relational operator specified by the user. Valid relational operators are: EQ, NE, GT, GE, LT, and LE.

0409 Column tag must be enclosed in parentheses.

0410 Panel name from the ERTB is invalid.

0411 Column name specified in the DAB is not recognized.

0412 Invalid relational operator. Only valid relational operators for the specified column are EQ and NE.

0413 Invalid relational operator. Only valid relational operator for the specified column is EQ.

0420 Invalid input parameter tags for module DGTFSO04.

0421 Filter value is longer than the allowed maximum length.

0450 Error occurred during DEVINFO invocation. Return code from DEVINFO indicates that an error was encountered. Detailed information has been placed in the problem determination area of the ISMF log.

0451 Respective field's displacement is zero in the return area header of DEVINFO.

0452 Source for the generated list is invalid. Source of a generated list should either be VTOC or CATALOG.

0453 Invalid connector was entered by the user, but was not detected by the panel check routine.

0454 Invalid operator was entered by the user, but was not detected by the panel check routine.

0455 Screen logical number obtained from the profile pool is either invalid (<0) or not in the defined range (1 to 8).

0456 Requested column cannot be found in the data set application static module.

0457 Invalid scroll amount was entered by the user.

0470 Conflict between CPPLPMAD and DGTEWORD. CPPLPMAD indicates that there is a parameter, but the length of the parameter returned by DGTEWORD is not greater than zero.

0471 Invalid pointer (GDRBIACB) to the first ISMF application control block (IACB).

0472 An error exists in the chaining of IACBs.

0473 A conflict exists in the IACB pointers.

0474 The pointer (GDRBIACB) to the first IACB is 0.

0501 TBVCLEAR: Severe table error. Return code: 20.

0502 TBVCLEAR: Table not open. Return code: 12.

0504 TBSARG: Severe table error. Return code: 20.

0505 TBSARG: Table not open. Return code: 12.

0506 TBSARG: All column variables are null, and the namelist parameter was not specified. No argument established. Return code: 8.

0507 TBSKIP: Severe table error. Return code: 20.

0508 TBSARG: Table not open. Return code: 12.

0509 TBSKIP: Variable value has been truncated, or insufficient space was provided to return all extension variable names. Return code: 16.

0510 TBSKIP: Current® row pointer would have gone beyond the number of rows in the table (this includes a table empty condition). ISPF current row pointer is set to 0 (top). Return code: 8.

0512 TBBOTTOM: Variable value has been truncated, or insufficient space was provided to return all extension variable names. Return code: 16.

0513 TBBOTTOM: Severe table error. Return code: 20.

0514 TBBOTTOM: Table not open. Return code: 12.

0515 TBBOTTOM: Table is empty, CRP set to top (0). Return code: 8.

0516 TBBOTTOM: CRP set to top (0). Return code: 8.

0517 TBTOP: Severe table error. Return code: 20.

0518 TBTOP: Table not open. Return code: 12.

0519 TBSORT: Severe table error. Return code: 20.

0520 TBSORT: Table not open. Return code: 12.

0521 TBSORT: Numeric convert error. Return code: 16.

0522 TBPUT: Severe table error. Return code: 20.

0523 TBPUT: Table not open. Return code: 12.

0524 TBPUT: CRP was at top and remains at top. Return code: 8.

0525 TBPUT: Numeric conversion error. Return code: 16.

0526 Improper DAB variable type. It must be C, H, or B.

0527 Call to DELETE was unsuccessful.

0528 Array does not exist.

0529 Invalid sort order: It must be A or D.

0530 SELBFLG1 is not specified or is incorrectly specified. One bit must be on.

0531 Caller's DAB does not match with any VDEFINE DABs.

0532 Scan for the nth row was unsuccessful. The last row found is returned.

0533 Improper search direction. Direction default is to next.

0534 DABL contains no DABs. Array cannot be created.

0535 Get storage for ARVT/RWCB control block failed.

0536 SRVTARB1 and/or SRVTARN1 in SRVT control block is not initialized.

0537 Extension DAB name is not returned in the extension variable list by ISPF.

0538 DABCLEN is invalid.

0539 Array already exists.

0540 Invalid DAB justification flag.

0541 Condition value for an extension DAB is invalid.

0542 Return code is invalid.

0581 Convert utility: Address of the control data attribute block is 0.

0582 Data set name verification: invalid character. Valid characters are: A-Z, 0-9, '#', '$', '@', '.', ')', '(', '*'. In addition, the first character within a set of parentheses may be '+', '-', 0, alphabetic, or national (@, $, #).

0583 Data set name verification: Incomplete data set name specification. Data set name must not end with a '.'.

0584 Data set name verification: Data set name is quoted. Data set name must not be quoted in this situation.

0586 Data set name verification: Invalid member name. Member name must start with a national (@, $, #) or alphabetic character; subsequent characters may also be numeric. Member name must not exceed 8 characters in length.

0587 Data set name verification: Invalid member name. Member name must not exceed 8 characters in length.

0588 Data set name verification: Invalid data set name. Member of a PDS must not be specified.

0589 Data set name verification: Invalid generation data set specification. GDG must be in the form (0), (+n), or (-n), where n is between 1 and 255.

0590 Data set name verification: Invalid data set name. Fully specified data set names are not allowed.

0591 Data set name verification: Data set name is too long. Maximum length, including '.', is 35 characters for a GDG data set name, and 44 characters for a fully specified data set name.

0592 Data set name verification: Invalid data set name. GDG specification is not allowed.

0593 Data set name verification: Invalid partial specification. More than 2 '*' in a single qualifier are not allowed.

0594 Data set name verification: Invalid data set name. Partial specification is not allowed.

0595 Volser verification: Partial specification is not allowed. Partial specification was not requested, but an '*' was found in the input volume serial number.

0596 Volser verification: Full specification is not allowed. Input volser was fully specified, but full specification was not requested.

0597 Log Utility: Request was made to log abend information, but the pointer to the retry work area (RWA) is not valid.

0598 Data set name verification: Qualifier was found that was longer than the allowable maximum of 8 characters.

0599 Data set name verification: Invalid qualifier specification. First character of a qualifier must be alphabetic or national (#, @, $).

0600 MVS/ESA* load SVC was unsuccessful. Requested module could not be loaded.

0601 No matching SCT entry was found.

0602 Unable to build storage control (SCT) entry.

0603 Reserved for CSR, DGTCGMO1. Return code: 8.

0604 Unable to build load list block (LLBL) entry.

0605 No matching LLBL entry found.

0606 Load common service routine failed while trying to link to another module via the ISMF link macro.

0607 Tried to decrement when use count=0.

0608 Convert utility: Truncated data. Target area specified was not large enough to contain the converted data. Data was truncated.

0610 Convert utility: Length of the source data must be from 1 to 4 bytes when converting fixed binary numbers.

0611 Convert utility: Invalid source data length. Length of the data to be converted must be greater than 0.

0612 Convert utility: Target data length is invalid. Length of the target data must be greater than 0.

0613 Convert utility: Source data address=0. Address of the data to be converted is 0.

0614 Convert utility: Target address=0. Address specified for the target address is 0.

0615 Data set name (DSN) verification failed.

0616 Starting address is larger than ending address. Input parameters supplied to the word-parsing routine (DGTEWORD) are invalid.

0617 Log: Message not found. Return code: 12.

0618 Log: Severe error. Return code: 20.

0619 Log: Invalid return code.

0620 Request was made to log status information, but no additional message IDs were specified. At least one message ID must be specified.

0621 Convert utility: Invalid conversion code.

0622 Convert utility: Invalid justification code.

0623 Data set name verification: Unbalanced quotes were found in the input data set name. Data set name may contain no quotes or two quotes (one at each end of the data set name).

0624 Invalid return code from ISPF VDELETE.

0625 Invalid return code from ISPF VPUT.

0626 Invalid return code from ISPF VGET.

0627 Invalid return code from ISPF VDEFINE.

0628 Invalid return code from ISPF SETMSG.

0629 Invalid return code from ISPF GETMSG.

0630 Invalid return code from ISPF DISPLAY.

0631 ISPF DISPLAY: Terminated by END or RETURN. User requested termination via END or RETURN command. Return code: 8.

0632 ISPF DISPLAY: Specified panel, message, or cursor field was not found. Return code: 12.

0633 ISPF DISPLAY: Truncation or translation error in storing defined variables. Return code: 16.

0634 ISPF DISPLAY: Severe error. Return code: 20.

0635 ISPF GETMSG: Specified message was not found. Return code: 12.

0636 ISPF GETMSG: Severe error. Return code: 20.

0637 ISPF SETMSG: Specified message was not found. Return code: 12.

0638 ISPF SETMSG: Severe error. Return code: 20.

0639 ISPF VDEFINE: Variable was not found. Return code: 8.

0640 ISPF VDEFINE: Data truncation occurred. Return code: 16.

0641 ISPF VDEFINE: Severe error. Return code: 20.

0642 ISPF VGET: Variable was not found. Return code: 8.

0643 ISPF VGET: Truncation or translation error occurred during data movement. Return code: 16.

0644 ISPF VGET: Severe error. Return code: 20.

0645 ISPF VPUT: Variable was not found. Return code: 8.

0646 ISPF VPUT: Truncation occurred while copying variables to the application profile pool. Return code: 16.

0647 ISPF VPUT: Severe error. Return code: 20.

0648 ISPF VDELETE: At least one variable was not found. Return code: 8.

0649 ISPF VDELETE: Severe error. Return code: 20.

0650 ISPF TBDELETE: For nonkeyed tables, the table was empty. ISPF current row pointer remains at the top. For keyed tables, the row specified by the key variables does not exist. ISPF current row pointer is set to top. Return code: 8.

0651 ISPF TBDELETE: Table is not open.

0652 ISPF TBDELETE: Severe error.

0653 ISPF TBDELETE: Return code is invalid.

0654 ISPF TBEND: Table is not open.

0655 ISPF TBEND: Severe error.

0656 ISPF TBEND: Return code is invalid.

0657 ISPF TBTOP: Table is not open.

0658 ISPF TBTOP: Severe error.

0659 ISPF TBTOP: Return code is invalid.

0660 ISPF TBVCLEAR: Table is not open.

0661 ISPF TBVCLEAR: Severe error.

0662 ISPF TBVCLEAR: Return code is invalid.

0663 ISPF TBADD: Row with the same key already exists. ISPF current row pointer is set to the top (0). Return code: 8.

0664 ISPF TBADD: Table is not open.

0665 ISPF TBADD: Numeric convert error.

0666 ISPF TBADD: Severe error.

0667 ISPF TBADD: Return code is invalid.

0668 ISPF TBPUT: For nonkeyed tables, the ISPF current-row pointer was at the top and remains at the top. For keyed tables, the key does not match that of the current row. ISPF current row pointer is set to the top. Return code: 8.

0669 ISPF TBPUT: Table is not open.

0670 ISPF TBPUT: Numeric convert error.

0671 ISPF TBPUT: Severe error.

0672 ISPF TBPUT: Return code is invalid.

0673 ISPF TBGET: For nonkeyed tables, the ISPF current-row pointer was at the top and remains at the top. For keyed tables, the row specified by the value in the key variables does not exist. The ISPF current row pointer is set to the top.

0674 ISPF TBGET: Table is not open.

0675 ISPF TBGET: Insufficient space to return all extension variable names.

0676 ISPF TBGET: Severe error.

0677 ISPF TBGET: Return code is invalid.

0678 ISPF TBCREATE: Table already exists.

0679 ISPF TBCREATE: Table in use, ENQ failed.

0680 ISPF VCOPY: Variable not found. Return code: 8.

0681 ISPF TBCREATE: Severe error.

0682 ISPF TBCREATE: Return codes are invalid.

0683 ISPF TBSORT: Table is not open.

0684 ISPF TBSORT: Numeric convert error.

0685 ISPF TBSORT: Severe error.

0686 ISPF TBSORT: Return code is invalid.

0687 ISPF TBBOTTOM: Table is empty.

0688 ISPF TBBOTTOM: Table is not open.

0689 ISPF TBBOTTOM: Insufficient space provided to return all extension variables.

0690 ISPF TBBOTTOM: Severe error.

0691 ISPF TBBOTTOM: Return code is invalid.

0692 ISPF TBSARG: All column variables are null, and the name-list parameter was not specified.

0693 ISPF TBSARG: Table is not open.

0694 ISPF TBSARG: Severe error.

0695 ISPF TBSARG: Return code is invalid.

0696 ISPF TBSKIP: CRP would have gone beyond the number of rows in the table.

0697 ISPF TBSKIP: Table is not open.

0698 ISPF TBSKIP: Insufficient space was provided to return all extension variables.

0699 ISPF TBSKIP: Severe error.

0700 ISPF TBSKIP: Return code is invalid.

0701 ISPF TBSCAN: Row does not exist. No match was found.

0702 ISPF TBSCAN: Table is not open.

0703 ISPF TBSCAN: Insufficient space was provided to return all extension variables.

0704 ISPF TBSCAN: Severe error.

0705 ISPF TBSCAN: Return code is invalid.

0706 TBADD: GETMAIN error. Return code: 20.

0707 Caller is not authorized to invoke this module.

0708 ISPF VCOPY: Truncation occurred. Return code: 16.

0709 ISPF VCOPY: Severe error. Return code: 20.

0710 FTCLOSE: Output file already exists. Return code: 4.

0711 FTCLOSE: File is not open. Return code: 8.

0712 FTCLOSE: Output file is in use. Return code: 12.

0713 FTCLOSE: Output file is not allocated. Return code: 16.

0714 FTCLOSE: Severe error. Return code: 20.

0715 FTCLOSE: Return code from ISPF is invalid.

0716 FTINCL: Skeleton does not exist. Return code: 8.

0717 FTINCL: Skeleton or table in use. Return code: 12.

0718 FTINCL data truncation occurred, or skeleton library or output file esd not allocated. Return code: 16.

0719 FTINCL: Severe error. Return code: 20.

0720 FTINCL: Return code from ISPF is invalid.

0721 FTOPEN: File tailoring already in progress. Return code: 8.

0722 FTOPEN: Output file in use. Return code: 12.

0723 FTOPEN: Skeleton library or output file was not allocated. Return code: 16.

0724 FTOPEN: Severe error. Return code: 20.

0725 Return code from ISPF is invalid.

0726 TBQUERY: Table is not open. Return code: 12.

0727 TBQUERY: Insufficient space provided. Return code: 16.

0728 TBQUERY: Severe error. Return code: 20.

0729 TBQUERY: Unexpected return code.

0730 ISPF VCOPY: Invalid return code.

0731 PDF LMINIT: DATA-ID not created. The error condition is described in the ISPF system dialog variables. Return code: 8.

0732 PDF LMMLIST: Empty member list. Return code: 4.

0733 PDF LMMLIST: LIST OPTION - End of member list. Return code: 8.
PDF LMMLIST: FREE OPTION - Member list does not exist.

0734 PDF LMMLIST: No data set associated with the given DATA-ID. That is, LMINIT has not been completed. Return code: 10.

0735 PDF LMMLIST: One of the following: The data set is not open or is not partitioned. The parameter value is invalid. Return code: 12.

0736 PDF LMMLIST: Truncation or translation error in accessing dialog variables. Return code: 16.

0737 PDF LMMLIST: Severe error, unable to continue. Return code: 20.

0738 PDF LMMLIST: Invalid return code.

0739 Table service failed, list may be incomplete.

0740 Index is not within bounds of the array.

0741 TBSTATS: Variable value has been truncated.

0742 TBSTATS: Severe error.

0743 TBSTATS: Invalid return code.

0806 Load module was not found.

1000 Invalid (unexpected) return code was received from the called function. This could be due to a logic error in the called function or a mismatch between calling and called function. For example, the called function might have added a new return code of which the caller is not aware.

1001 CDTXSTAT field of CDTX contains an unrecognizable value.

1002 Visual ID of the control block is invalid. This could be caused by an invalid address in the pointer to the control block or by incorrect initialization of the visual ID of the control block.

1003 Function requested by the caller is invalid.

1004 Value of the DACDTOT field of DACD does not correspond to the total number of DSN DABs.

1005 While preparing to update the filter status field (DABFS), the DSN DAB was not found.

1006 DYNALLOC Error.

1007 RDJFCB Error.

1008 Open Error.

1009 Duplicate column name in DAB.

1010 Column name in DAB is invalid.

1011 DACL is invalid.

1012 Duplicate DACL.

1013 Column name in DACL is invalid.

1014 More than one DSN was specified for generic DSN.

1015 Percent of ALLOC not used cannot be calculated because either the allocated space or the allocation used is invalid.

1016 GFL error: Nonzero return code.

1017 GFL error: Catalog name entry.

1018 GFL error: Data set name entry.

1019 GFL error: The length was invalid.

1020 Status of the data set name (FCLDSNST) returned by CVAFFILT in FCL is invalid.

1021 TRKCALC error.

1022 DASDCALC error.

1023 DEVINFO error.

1024 CVAFFILT error.

1025 DFSMShsm is not installed or is inactive.

1026 Volume serial number verification: An invalid character was found.

1027 Volume serial number verification: Too many asterisks were found. Only one asterisk may be specified in a partial specification.

1028 Invalid operator was found in DACL during selection criteria processing.

1029 Invalid logical connector was found in DACL during selection criteria processing.

1030 GFL error: No data is returned.

1031 Nonzero return code from catalog management.

1032 Unable to acquire needed data.

1033 Needed data from SWAREQ not acquired.

1034 Percentage of user data reduction cannot be calculated.

1047 Error acquiring data needed to invoke DASDCALC.

1048 Error acquiring keyrange qualifier.

1049 Error acquiring volume flag.

1050 Data set is not partitioned.

1051 Data set does not have SAM/PAM DSORG.

1052 Locate failed. See feedback for catalog reason code.

1053 Unable to locate cluster name for VSAM component.

1054 Data set name is GDG base name.

1055 Data set name is VSAM data or index component name.

1056 Data set's allocated used space is 0.

1057 Data set is migrated.

1058 Submit failed.

1059 RESTORE LIST command invalidly entered on a selection panel.

1060 RESTORE command parameter is neither list nor data set.

1061 More than one RESTORE command parameter entered.

1062 Output data set for background job is neither physical sequential nor a member of a PDS.

1063 Select failed. See feedback field for select return code.

1064 Return code from DYNALLOC is 0, but the SVC99 information reason code is not 0. See feedback for SVC99 request type, error reason code, and information reason code.

1065 Return code from DYNALLOC is 4. See feedback for SVC99 request type, error reason code, and information reason code.

1066 Return code from DYNALLOC is 8. See feedback for SVC99 request type, error reason code, and information reason code.

1067 Return code from DYNALLOC is 12. See feedback for SVC99 request type, error reason code, and information reason code.

1069 VSAM component name found in catalog.

1070 System residence volume serial number (ETSYSRES) in environment table is blank.

1071 Return code from DEVINFO is 0 or 8. UCBTYP is not DASD.

1072 Data set is a user catalog.

1073 Unexpected allocation request type received during background job submission. See feedback for unexpected allocation request type.

1074 Invalid parameter entered on the command line.

1075 Volser verification: is blank.

1076 Volser verification: is a single asterisk.

1077 No column name match found in static column load module. See feedback for column name requested and static column load module name.

1078 No column name match was found in the DAB chain.

1079 Unexpected panel number was encountered. See feedback field for unexpected panel number.

1080 Unexpected panel variable name was encountered. See feedback field for unexpected panel name.

1081 Unexpected catalog entry type was received back from call to DGTEARFN.

1082 Unexpected column name was encountered during verification of data sets in the list. See feedback for unexpected column name.

1083 Unexpected request type was received during array processing. See feedback for the unexpected request type.

1084 No data set names found in the array.

1085 Unexpected file tailoring service was requested during background job submission. See feedback for the unexpected tailoring service name.

1086 Unexpected request type was received during filter data set allocation processing. See feedback for the unexpected request type.

1087 Open error: See feedback for indication of whether data set was input or output, return code from open, and data set name.

1088 Duplicate data set name found in data set list.

1089 Unexpected DCB address was found during background job submission. See feedback for current DCB address and all 3 valid DCB addresses.

1090 More than 255 input volume serial numbers.

1091 Data set name is VSAM cluster name.

1092 Unexpected option flag received during the process of validating the data set name. See feedback field for the unexpected option field.

1093 Unexpected GETMAIN request flag received. See feedback for the unexpected request flag value.

1094 Unexpected job disposition type received during background job submission. See feedback for the unexpected job disposition type.

1095 Close error: See feedback for indication of whether data set was input or output, return code from close, and data set name.

1096 Invalid request flag found during BUILD/DELETE of data set name lists. See feedback for flag byte.

1097 Volser is null or in error.

1098 Output data set for a background job is not a PDS. (See feedback for hexadecimal DSORG value and data set name.)

1099 BLDL failed.

1101 Invalid RESTORE parameter was entered on the data set selection panel.

1102 Invalid job submitted message from CLIST DGTQSU01 submit processing.

1103 Device type is null or in error.

1104 ISPF TBOPEN: Table does not exist.

1105 ISPF TBOPEN: Enqueue failed, table in use.

1106 ISPF TBOPEN: Table input library not allocated.

1107 ISPF TBOPEN: Severe error.

1108 ISPF TBOPEN: Invalid return code.

1109 ISPF TBCLOSE: Table is not open. Return code: 12.

1110 ISPF TBCLOSE: Table output library is not open. Return code: 16.

1111 ISPF TBCLOSE: Severe error. Return code: 20.

1112 ISPF TBCLOSE: Unexpected return code.

1113 Last use mode is undefined for hide.

1114 Unable to build DEVT table.

1115 Nonzero return code from DGTCDT01.

1200 Error acquiring volser.

1201 Error acquiring use attribute.

1202 Error acquiring device number.

1203 Error acquiring device type.

1204 Error acquiring cylinders per volume.

1205 Error acquiring free extents.

1206 Error acquiring free cylinders.

1207 Error acquiring additional free tracks.

1208 Error acquiring cylinders in largest free extent.

1209 Error acquiring additional tracks in largest free extent.

1210 Error acquiring format-0 DSCB count.

1211 Volume capacity cannot be calculated because needed data was not acquired.

1212 Free space cannot be calculated because needed data was not acquired.

1213 Percent free space cannot be calculated because needed data was not acquired.

1214 Percent free space cannot be calculated because needed data has invalid value.

1215 Allocated space cannot be calculated because needed data was not acquired.

1216 Largest extent cannot be calculated because needed data was not acquired.

1217 Cannot determine if DASD is shared because needed data was not acquired.

1218 LSPACE error.

1219 LSPACE cannot be invoked because UCB address was not acquired.

1220 Error acquiring tracks per cylinder.

1221 VTOC size cannot be acquired because needed data cannot be acquired.

1222 Not authorized to access locked catalog.

1223 Catalog temporarily unavailable.

1224 Unauthorized locked catalog was encountered during a search of catalogs for data sets that meet the partially specified data set name.

1225 Catalog temporarily unavailable while searching catalogs for data sets which meet the partially specified data set name.

1226 Specified catalog does not exist.

1230 Invalid criteria name specified in IACL. See feedback for invalid name.

1231 Device type verification: Invalid character found.

1232 Device type verification: Device type is blank.

1233 Device type verification: More than one asterisk was specified in a partial specification.

1234 Device type verification: Partial specification was not requested, but an asterisk was found in the input device name.

1235 Device type verification: Full specification was not requested, but no asterisk was specified.

1236 Device type verification: Device type is a single asterisk.

1237 Device type verification: Device type is not a supported generic name.

1238 TBMOD: Keys did not match, new row added to table. Return code: 12.

1239 TBMOD: Table is not open. Return code: 12.

1240 TBMOD: Numeric conversion error. Return code: 12.

1241 TBMOD: Severe error. Return code: 12.

1242 TBMOD: Invalid return code. Return code: 12.

1243 TBERASE: Table does not exist in the output library. Return code: 12.

1244 TBERASE: Table in use, ENQ failed. Return code: 12.

1245 TBERASE: Output library not allocated. Return code: 12.

1246 TBERASE: Severe error. Return code: 12.

1247 TBERASE: Invalid return code. Return code: 12.

1248 VIEW: View failed. Return code: 12.

1250 Save command error: Invalid list name. List name cannot begin with 'ISR', 'ISP', or 'DGT' and it cannot be quoted.

1252 ALTER error: Data set ALTER against a non-SMS-managed data set is not allowed.

1253 Data set is a VSAM alternate index.

1254 ALTER error: Nonzero return code from SVC 26 ALTER. See feedback for return code and reason code.

1255 Data set has unexpected type of catalog entry.

1256 SUBMIT error: Invalid job submitted message from CLIST DGTQSF01.

1257 Caller requested an invalid execute statement panel.

1258 Storage group name longer than the 8-byte maximum.

1259 Error in CSI GETLIST.

1260 Error in AOMSERV.

1261 DEVINFO resume is not allowed.

1262 Invalid input parameters.

1263 SCRATCH error: No volumes containing any part of the data set were mounted. The data set may be a VIO data set that was not allocated during SCRATCH. Return code: 4.

1264 SCRATCH error: Unusual condition was encountered on one or more volumes. Return code: 8.

1265 SCRATCH error: Invalid volume list. Return code: 12.

1268 OBTAIN error: For return code 04, volume was not mounted; for return code 08, DSCB was not found; for return code 12, I/O error; for return code 16, invalid pointer.

1269 Variable data set record format was specified; must be fixed format.

1270 Repeat line operator used without previous line operator.

1271 Error occurred during processing of user line operator or command.

1272 VREPLACE: Truncation occurred during data movement.

1273 VREPLACE: Severe error.

1274 VREPLACE: Invalid return code.

1275 Error occurred during creation of user command anchor block.

1276 Invalid application ID for this service.

1277 User command failed.

1278 Failure occurred during processing of a list.

1279 Error occurred during initialization of user command.

1280 Invalid return code for user command initialization.

1281 User command control variable could not be retrieved from shared variable pool.

1282 Error occurred in DSERVXIT.

1283 Invalid area specified. Valid values are ALL, DATA, or BOTH.

1284 Invalid value was specified for FROM. Valid values are LAI or SLAI.

1285 Format name in data set anchor block is invalid.

1286 DSERVXIT cannot be invoked because UCB address was not acquired.

1287 Could not position CRP to top of array.

1288 AOMSERV returned nonzero return/reason code.

1289 Invalid subroutine call to DGTFCBSB.

1290 Could not position to top of array.

1291 Error during LASB initialization.

1292 PDF LMINIT: Invalid parameter value. Return code: 12.

1293 PDF LMINIT: Truncation or translation error in accessing dialog variables. Return code: 16.

1294 PDF LMINIT: Severe error. Return code: 20.

1295 PDF LMINIT: Invalid return code.

1296 PDF LMOPEN: Data set could not be opened. Return code: 8.

1297 PDF LMOPEN: No data set associated with the given data ID. Return code: 10.

1298 PDF LMOPEN: Invalid parameter value; data set is already open; cannot open concatenated data sets for output; cannot open a data set allocated SHR for output; cannot open a data set allocated MOD for input. Return code: 12.

1299 PDF LMOPEN: Truncation or translation error in accessing dialog variables. Return code: 16.

1300 PDF LMOPEN: Severe error. Unable to continue. Return code: 20.

1301 PDF LMOPEN: Invalid return code.

1302 PDF LMCLOSE: Data set is not open. Return code: 8.

1303 PDF LMCLOSE: No ISPF library or data set is associated with the given data ID. Return code: 10.

1304 PDF LMCLOSE: Severe error. Unable to continue. Return code: 20.

1305 PDF LMCLOSE: Invalid return code.

1306 PDF LMFREE: Free data set failed. Return code: 8.

1307 PDF LMFREE: No data set is associated with the given data ID. Return code: 10.

1308 PDF LMFREE: Severe error. Unable to continue. Return code: 20.

1309	PDF LMFREE: Invalid return code.
1310	PDF LMGET: No ISPF library or data set is associated with the given data ID. Return code: 10.
1311	PDF LMGET: Data set is not open or is not open for input; LMMFIND not done for a partitioned data set; invalid parameter value. Return code: 12.
1312	PDF LMGET: Truncation or translation error in accessing dialog variables. Return code: 16.
1313	PDF LMGET: Severe error. Return code: 20.
1314	PDF LMGET: Invalid return code.
1315	Cannot process command against empty list.
3000	ESTAE macro invocation failed with return code 4.
3001	ESTAE macro invocation failed with return code 8.
3002	ESTAE macro invocation failed with return code 12.
3003	ESTAE macro invocation failed with return code 16.
3004	ESTAE macro invocation failed with return code 20.
3005	ESTAE macro invocation failed with return code 24.
3006	ESTAE macro invocation failed with an unknown return code.
3007	ISPLINK (CONTROL) service failed with return code 20.
3008	An ABEND occurred. The retry routine completed.
3009	ISPLINK (DISPLAY) provided return code 0; return code 8 was expected.
3010	ISPLINK (DISPLAY) provided return code 12; return code 8 was expected.
3011	ISPLINK (DISPLAY) provided return code 16; return code 8 was expected.
3012	ISPLINK (DISPLAY) provided return code 20; return code 8 was expected.
3013	The current panel was not found in the panel array.
3014	The current panel matches the last panel in the panel array.
3015	The current panel matches the first panel in the panel array.
3016	One or more areas of ISMF storage was not freed.
3017	FREEMAIN provided return code 04; not all virtual storage freed.
3018	FREEMAIN provided return code 08; part of area being fixed is still free.
3019	FREEMAIN provided return code 12; page table is paged out.
3020	Unknown return code from DGTCFM01.
3021	Storage was freed successfully by DGTCFM01.
3022	Unknown return code from FREEMAIN.
3023	DGTCLGCS failed with return code 12.

3024 DGTCLGCS failed with an unexpected return code.

3025 One or more calls to ISPLINK (VDEFINE) failed.

3026 One or more calls to ISPLINK (VDELETE) failed.

3027 Unexpected return code from the volume verification routine.

3028 Invalid data set name.

3032 Termination did not complete.

3033 Initialization failed.

3034 Unexpected return code—logged by caller.

3035 ESTAE was not successful, but processing continued.

3036 VDEFINE failed.

3037 Insufficient storage.

3038 Load command processor modules failed.

3039 Prolog failure in DGTFCTIN.

3040 ISMF command table processor initialization failed.

3041 VPUT failed.

3042 VDELETE failed.

3043 Prolog failure in DGTFCTTE.

3044 Unable to release command processor control blocks.

3045 ESTAE termination abended.

3049 Application selection failed.

3051 Prolog failure in DGTFCTSE.

3052 Load ISMF command table failed.

3055 DEFINE failed.

3056 ALTER failed.

3057 Invalid option was not detected by DGTFSASV.

3061 PREPARE ALTER failed.

3062 DISPLAY ALTER failed.

3063 PREPARE DEFINE failed.

3064 DISPLAY DEFINE failed.

3068 VGET failed.

3070 Prolog failure in DGTFCTCK.

3071 Unable to process command.

3072 Invalid command was not detected by DGTFCTCK.

3073 Prolog failure in DGTFCTPR.

3074 Command processor, DGTFCTPR, failed.

3075 SCDS saved.

3076 Control service failed with a nonzero return code.

3077 No members in library; return code 16 from PDF EDIT.

3078 Severe error from PDF EDIT; return code 20 from PDF EDIT.

3079 Invalid return code from PDF EDIT.

3080 Empty sequential data set or zero length member of a PDS; return code 12 from PDF BROWSE.

3081 Specified member not found; return code 14 from PDF BROWSE.

3082 No members in library; return code 16 from PDF BROWSE.

3083 Severe error encountered; return code 20 from PDF BROWSE.

3084 Invalid return code from PDF BROWSE.

3085 Nonzero return code from ISPF control function.

3086 OPEN provided return code 4.

3087 OPEN provided return code 8.

3088 OPEN provided return code 12.

3089 OPEN provided an unknown return code.

3090 CLOSE provided return code 4.

3091 CLOSE provided return code 8.

3092 CLOSE provided return code 12.

3093 CLOSE provided an unknown return code.

3094 Member name specified, but no data set name specified.

3095 Allocation unit specified, but no primary or secondary amount specified.

3096 Primary space specified, but no allocation unit specified.

3097 Secondary space specified, but no allocation unit specified.

3098 Invalid value for DSORG.

3099 LRECL greater than 32760.

3100 Invalid value for RECFM.

3101 BLKSIZE greater than 32760.

3102 SVC99—unavailable resource.

3103 SVC99—data set not found.

3104 SVC99—environmental error.

3105 SVC99—system error.

3106 SVC99—unknown error.

3107 Invalid value for initial disposition.

3108 Invalid value for exit disposition.

3109 Both DDNAME and DSNAME specified.

3110 DGTCLD01 provided return code 12.

3111 DGTCLD01 provided an unknown return code.

3112 DGTCGM01 provided return code 12.

3113 DGTCGM01 provided an unknown return code.

3114 DGTCDT01 provided return code 12.

3115 DGTCDT01 provided an unknown return code.

3116 Module could not be loaded successfully.

3117 Error—SMS configuration services.

3118 CDS put base failed.

3119 CDS unaccess failed.

3120 Specified SCDS does not contain a system name. Optical libraries require at least 1 system name in the SCDS.

3126 Specified SCDS does not contain a system name. Object and object backup storage group require at least 1 system name.

3127 Invalid option not detected by DGTFFLSV.

3130 Edit failed.

3131 Translate failed.

3135 ISPLINK (DISPLAY) provided an unknown return code; return code 8 was expected.

3136 ISPLINK (CONTROL) failed with an unknown return code.

3138 Translator prepare input failed.

3140 Translator failed.

3142 Prolog failure in DGTCIOAL.

3143 Unable to allocate.

3146 Prolog failure in DGTECFPT.

3148 Browse failed.

3149 Browse termination failed.

3154 Count parameter out of range.

3172 SMS construct saved.

3173 Put element failed.

3174 SVC99 error.

3175 DGTCFM01 provided return code 12.

3183 Invalid option was not detected by panel verification.

3184 Volume was defined to another storage group.

3189 Invalid type was not detected by panel verification.

3218 Unable to convert to displayable form.

3219 Unable to convert to internal form.

3220 Display SG volume selection panel failed.

3221 SG volume selection dialog failed.

3222 CDS 8-Name mode conflicts with system 32-Name mode.

3229 Prepare DGTDCSG5 failed.

3230 Display DGTDCSG5 failed.

3231 Prepare DGTDCSG4 failed.

3232 Display DGTDCSG4 failed.

3233 SMS has identified an error condition.

3234 Data set was not found.

3235 Caller is not authorized to access the data set.

3236 Read access was requested, and data set is empty.

3237 Element was not found.

3238 No matching elements were found.

3239 ACS routine was not translated because of user errors; listing contains error messages.

3240 The CDS failed SMS full validation.

3241 CDS 32-Name mode conflicts with system 8-Name mode.

3242 Volume serial number delete failed.

3243 Add volume serial number failed.

3244 No DDNAME specified in the free data set control block (FRCB).

3245 Invalid units of allocation.

3246 Function name was not found in function table.

3247 Specified panel could not be found; return code 12 from ISREDIT.

3248 Truncation error while storing ZCMD or ZSEL; ISREDIT—return code 16.

3249 Severe error from ISREDIT—return code 20.

3250 Unexpected error return code from ISREDIT.

3251 Unexpected return code from DGTFIOS1.

3252 Release command processor modules failed.

3253 DGTFIOF2 detected a nonzero return code.

3254 Prolog failure.

3255 No IACB to release.

3256 Nonzero return code from DGTCLD01.

3257 SMS indicated that a resource is not available.

3258 DGTEGMEM provided a nonzero return code.

3259 Error—ACS translator.

3260 Error—ACS executor.

3261 The data set was previously updated.

3262 DGTEFMEM provided a nonzero return code.

3263 DGTELDMD (LOAD) error.

3264 DGTEDTMD (DELETE) error.

3265 Element already exists.

3266 DGTEFMEM error and DGTEDTMD error.

3267 Invalid value of APPFMODE from the ISMF profile variable table variables (DGTMPVTV).

3268 CDS is not formatted.

3269 Invalid access mode parameter for CSI.

3270 Invalid function code parameter for CSI.

3271 A CDS is accessed with the access token.

3272 No CDS is accessed with the access token.

3273 ABEND logging failed with an unexpected return code.

3274 The data set is not an SCDS.

3275 Invalid element type.

3276 Blank SMS construct name.

3277 Retrieve information from VLA failed.

3278 The data set is not a CDS.

3279 Cannot delete the CDS base.

3280 List services is unable to find a match on the DAB column name.

3281 List services is unable to add a row of data to the array.

3282 List services is unable to release the ISMF reserved storage.

3283 Array services modules could not be loaded.

3284 Format services modules could not be loaded.

3285 Duplicate APID found.

3286 DGTEGMEM (GET STORAGE) error.

3287 DGTEFMEM (FREE STORAGE) error.

3288 Initialize format services failed.

3289 Move display window failed.

3290 Initialize line operator executor failed.

3291 Unable to release line operator executor control block.

3292 Creation of the list area image (LAI) failed.

3293 A status check of the array failed.

3294 A sort of the array failed.

3295 Obtaining of the cursor position failed.

3296 Positioning of the cursor failed.

3297 Updating of the fixed area failed.

3298 Highlighting of the error entry failed.

3299 Display of a panel failed.

3300 Unable to process input.

3301 Line operator and list command conflict.

3302 DGTCCFAC (CSI ACCESS CDS) provided return code 12.

3303 DGTCCFGT (CSI GET ELEMENT) provided return code 12.

3304 Terminating error from DGTFDA00 (data acquisition).

3305 Invalid SG type detected by convert.

3306 No matching elements found.

3307 Names too long—elements deleted.

3308 Common filter service error.

3309 List tailoring, DGTFCFLI, failed.

3310 Validation indicated errors.

3311 Not authorized to activate.

3312 SSI did not schedule an activation.

3313 Data set is in use.

3314 Validation failed.

3315 List services anchor block does not exist.

3316 SMS is not operational.

3317 Screen logical number is not within the bounds of the array.

3318 SMS is not active; there is no active configuration.

3319 Translation of column tags into column names failed.

3320 Maximum number of valid commands allowed on panel (20) exceeded.

3321 Invalid value was encountered during conversion of data from displayable format to internal format.

3322 Either GDRBLSAB or GDRBLSVT is invalid, or both are invalid.

3323 Nonzero return code from security program.

3324 Unable to branch to security bridge program.

3325 Unknown return DSORG from SVC99.

3326 OPEN of test library failed.

3327 CLOSE of test library failed.

3328 READ of test library failed.

3329 Unrecognizable string in test library member.

3330 Data set is not a partitioned data set—listing writer.

3331 No member was specified for a partitioned data set—listing writer.

3332 OPEN failed—listing writer.

3333 BLDL failed—listing writer.

3334 OBTAIN failed—listing writer.

3335 RDJFCB failed—listing writer.

3336 Type of data set is invalid—listing writer.

3337 SYNAD error on GET—listing writer.

3338 SYNAD error on PUT—listing writer.

3339 Invalid function specified—listing writer.

3440 OAM error—return code and reason code from OAM will be placed in feedback area.

3341 Invalid DCB pointer—listing writer.

3342 Invalid pointer to data—listing writer.

3343 Invalid title specified—listing writer.

3344 Load of DGTFIORT failed—listing writer.

3345 Invalid DD specified—listing writer.

3346 Duplicate system name.

3347 System name list is full.

3348 System name was not found.

3349 CSI termination failed.

3350 Write of test library failed.

3351 ACS routine type is unknown.

3352 Invalid function specified—listing writer.

3353 SYNAD detected an I/O error—listing writer.

3354 LRECL for listing data set is invalid.

3355 Invalid use of member name with a sequential data set.

3356 No member specified for a partitioned data set.

3357 Nonexistent member specified for a partitioned data set.

3358 FIND macro failed; see feedback area for reason code from FIND macro.

3359 VIO storage group types are not volume associated.

3360 Listing writer failed.

3361 Cannot change active configuration.

3362 Caller is not authorized to invoke this function.

3363 LISTSYS valid only on pool storage group type and active CDS.

3364 ACS test library cannot have an LRECL of less than 60.

3365 Not authorized to open the listing data set for output.

3366 Unable to process the line operator.

3367 The data set record organization is not a linear data set.

3368 Out of space for the data set.

3369 Invalid subroutine call to DGTFLCSB/DGTFRCSB/DGTFLMSB.

3370 Invalid storage group type.

3371 Element is not connected to current system.

3372 SCDS access is outstanding.

3373 SCDS access is not outstanding.

3374 CAF error.

3375 SQL error.

3376 DB2 row was not found.

3377 DB2 row already exists.

3378 DB2 setup error.

3379 System index value is missing.

3380 Attempt to save before system status defined.

3381 OAM is active.

3382 CTC/port combination already exists.

3383 CTC/SCSI combination already exists.

3384 OAM optical configuration database initialization failed.

3385 OAM optical configuration database volume table setup failed.

3386 OAM optical configuration database volume row retrieval failed.

3387 OAM optical configuration database volume table termination failed.

3388 OAM optical configuration database termination failed.

3389 No matching volume table rows were found.

3390 OAM data acquisition invalid function.

3391 No storage available for volume table data.

3392 Invalid volume type.

3393 OSMC is not active.

3394 CBRHLOPP call error.

3395 Invalid volume location.

3396 OAM is not active.

3397 OAM eject error.

3398 256 drives already defined.

3399 A pseudo library already exists.

3400 Duplicate drive number.

3401 VOLLIST parameter error.

3402 Volume serial number is not found.

3403 Volume is shelf resident.

3404 Unexpected reason code from OAM.

3405 OAM tape configuration database initialization failed.

3406 OAM tape configuration database volume retrieval setup failed.

3407 OAM tape configuration database volume row retrieval failed.

3408 OAM tape configuration database volume retrieval termination failed.

3409 OAM tape configuration database termination failed.

3410 No matching tape volume records found.

3411 ATL data acquisition invalid function.

3412 No storage available for tape volume record data.

3413 Invalid EJECT parameter.

3414 Caller is not authorized to access the catalog that contains the TCDB.

3415 Catalog access error.

3416 Catalog entry was not found.

3417 Catalog entry already exists.

3418 Catalog setup error.

3419 OAM1 subsystem was not started.

3420 AUDIT error.

3421 DEFRAG error.

3422 Catastrophic error propagated to end of volume list.

3423 Unexpected return code/reason code from DGTFDAGE.

3424 List control blocks might have been damaged - exit to POM.

3425 Not enough output array entries for DGTSCFST.

3426 OAM function CBRSCMDA returned an invalid drive address.

3427 Not authorized to write to the data set. Volume is READ only.

6000 Unsupported DFSMS/dfp version number was found.

6001 Unable to locate associations or volume serial numbers for VSAM cluster.

Chapter 10. Media Manager Diagnostic Aids

The following major diagnostic aids are provided for the media manager:
- A sample summary dump
- A description of media manager return and status codes

If a message or abend code is related to the failure, the following publications might help you identify additional symptom keywords:
- *z/OS MVS System Codes*
- *z/OS MVS System Messages, Vol 1 (ABA-AOM)*
- *z/OS MVS System Messages, Vol 2 (ARC-ASA)*
- *z/OS MVS System Messages, Vol 3 (ASB-BPX)*
- *z/OS MVS System Messages, Vol 4 (CBD-DMO)*
- *z/OS MVS System Messages, Vol 5 (EDG-GFS)*
- *z/OS MVS System Messages, Vol 6 (GOS-IEA)*
- *z/OS MVS System Messages, Vol 7 (IEB-IEE)*
- *z/OS MVS System Messages, Vol 8 (IEF-IGD)*
- *z/OS MVS System Messages, Vol 9 (IGF-IWM)*
- *z/OS MVS System Messages, Vol 10 (IXC-IZP)*

A Sample Summary Dump

A program check or other abend that occurs either in the media manager or in an exit routine results in a summary dump of the media manager control blocks. The dump is written to the SYS1.DUMPxx data set using the SDUMP macro instruction. If an indeterminate error occurs, an error record may be written to SYS1.LOGREC.

The media manager uses SDUMP to get a dump only if the operating environment was unable to establish a functional recovery routine (FRR). An FRR is not established for functions invoked via the MMCNVT or MMINIT macros. Functions invoked via the MMCALL procedure are protected by an FRR.

Figure 13 on page 188 is a sample of a summary dump. The first area is the SUMLIST RANGE, which shows the media manager vector table (MMVT). The second area is the range that shows the media manager process block (MMPB), which contains information on the current status of the media manager request. The third area, identified as the SDWA, is the system diagnostic work area and contains the name of the module most recently in control.

© Copyright IBM Corp. 1979, 2002

```
----- SUMLIST RANGE -------- ASID FFFF.


00FD6920                                            045820C9 C3E8D4D4 E5E340F0 F961F0F3  *                ...ICYMMVT 09.03*
00FD6940   61F8F7C8 C4D7F3F3 F1F040D5 D6D5C540   404040F5 F6F6F560 E7C1F340 4DC35D40  *.87HDP3310 NONE    5665.XA3 .C. *
00FD6960   C3D6D7E8 D9C9C7C8 E340C9C2 D440C3D6   D9D74B40 F1F9F8F7 6B40F1F9 F8F840D3  *COPYRIGHT IBM CORP. 1987. 1988 L*
00FD6980   C9C3C5D5 E2C5C440 D4C1E3C5 D9C9C1D3   E2406040 D7D9D6D7 C5D9E3E8 40D6C640  *ICENSED MATERIALS . PROPERTY OF *
00FD69A0   C9C2D440 00FD6A10 00FD6B10 81007F98   8100B068 81007510 81009CA0 8100AA98  *IBM ............................*
00FD69C0   81006520 81A0B000 81BD0078 8100B770   81006B68 8100B3F8 810062F8 81005698  *......................8...8....*
00FD69E0   81008FD8 81006700 81009A40 81008650   81005278 81008DA0 81009FC0 8100A578  *...Q....... ...................*
00FD6A00   81006D90 810078A8 81007C90 00000000   00000000 00000000 00000000 00000000  *................................*
COMPON=MEDIA MANAGER, COMPID=28415, ISSUER=ICYFRR            MODULE IEAVTSDT DATE 09/14/87  TIME 10.50.41  PAGE 00000029
00FD6A20   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
********   LINES FROM 00FD6A40 TO 00FD6B00 ARE THE SAME AS THE ABOVE LINE
00FD6B00   00000000 00000000 00000000 00000000   C0000000 03000000 81009FC0 000014C0  *................................*
00FD6B20   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
00FD6B40   00000000 00000000 00000000 81008650   81005278 81006700 00000000 00000000  *................................*
00FD6B60   00000000 00000000 08000000 00000000   00000000 00000000 00000000 00000000  *................................*
00FD6B80   C9C3E8D4 D4E2E5F1 0000FFFF 019A7000   0000FFFB 00000000 0000FFFB 00000000  *ICYMMSV1.......................*
00FD6BA0   0000FFFB 00000000 0000FFFB 00000000   0000FFFB 00000000 0000FFFB 00000000  *................................*
********   LINES FROM 00FD6BC0 TO 00FD6C80 ARE THE SAME AS THE ABOVE LINE
00FD6C80   80000000 00000000 D4D4D7E3 00000000   019A7020 00000000 00000000 00000000  *........MMPT...................*
00FD6CA0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
********   LINES FROM 00FD6CC0 TO 00FD6D60 ARE THE SAME AS THE ABOVE LINE
00FD6D60   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
00FD6D80   00000000 00000000                                                            *........                        *


----- SUMLIST RANGE -------- ASID FFFF.


019A7000                                            019A7000 019A77A8 00000000 0000000F  *                ................*
019A7020   D4D4D7C2 50000060 00C30B70 00C30C18   0001BA14 0000000F 008D61A8 10890000  *MMPB.....C...C.................*
019A7040   00000000 019A7010 019A77A8 00000000   00002800 00000000 00000000 00000000  *................................*
019A7060   00000000 0550FF05 019A7258 00000000   00000000 00FD6C90 00000000 00000000  *................................*
019A7080   00000CC8 00C30B70 C4000000 00000000   01000000 00021461 00C30C68 C4100000  *...H.C..D................C..D...*
019A70A0   0300000F 81009FC0 004114C0 00FA2138   019A72A8 0C50020F 019A7108 019A7020  *................................*
019A70C0   00000000 40170000 00000000 00000000   00F9CA00 00000000 81008DA0 81008DA0  *.... ............9.............*
019A70E0   81009A40 003BB278 019A7278 00C30C18   0100000E 08600000 00000000 00000000  *... .........C.................*
019A7100   00000000 00000200 E2D9C240 00000000   00F79080 0000000F 008D61A8 810C7D10  *........SRB .....7.............*
019A7120   810EAFE0 019A709C 00000000 00040000   00000000 00000000 00000000 00800000  *................................*
019A7140   008FF13C 810C86D2 00000000 00F9CA00   019A709C 019A709C 00FA20E8 00000CC8  *..1....K.....9............Y...H*
019A7160   810C7D12 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
019A7180   019A7084 00000000 00000000 00000000   81008EBA 81006520 00FA2138 019A7084  *................................*
019A71A0   00C30C18 019A7298 019A7618 019A7298   810013F1 00FA2138 019A72A0 00000000  *.C................1...........*
019A71C0   81008DA2 019A7020 019A7000 00000000   00C30B70 00C30C18 019A709C 019A7618  *................C...C..........*
019A71E0   019A7298 810013F1 00FA2138 019A72A0   00000000 81008DA2 019A7020 019A709C  *.......1.......................*
019A7200   019A7184 81008EBA 00000000 0002000A   019A7578 50000060 019A709C 00C30C18  *............................C..*
019A7220   80FD6B88 0000FFFF 019A7084 00000000   00000000 00FD6C90 80FD6C88 00000000  *................................*
019A7240   019A7020 81009CD8 0001BA14 019A7FF0   00000000 00C30B88 019A7290 019A7268  *.......Q.......0.....C.........*
019A7260   01050000 00021461 00C30C68 00C30BA0   C0000098 00000002 23600001 003BB091  *.........C...C.................*
019A7280   31400005 003BB103 08000000 003BB280   08000000 003BB298 1DC00008 003BB618  *. ..............................*
019A72A0   00440200 003BB578 1DC00008 003BB620   00440200 003BB580 1DC00008 003BB628  *................................*
019A72C0   00440200 003BB588 1DC00008 003BB630   00440200 003BB590 1DC00008 003BB638  *................................*
019A72E0   00440200 003BB598 1DC00008 003BB640   00440200 003BB5A0 1DC00008 003BB648  *............... ...............*
019A7300   00440200 003BB5A8 1DC00008 003BB650   00440200 003BB5B0 1DC00008 003BB658  *................................*
019A7320   00440200 003BB5B8 1DC00008 003BB660   00440200 003BB5C0 1DC00008 003BB668  *................................*
019A7340   00440200 003BB5C8 1DC00008 003BB670   00440200 003BB5D0 1DC00008 003BB678  *.......H........................*
019A7360   00440200 003BB5D8 1DC00008 003BB680   00440200 003BB5E0 1DC00008 003BB688  *.......Q........................*
019A7380   00440200 003BB5E8 1DC00008 003BB690   00440200 003BB5F0 1DC00008 003BB698  *.......Y...............0........*
019A73A0   00440200 003BB5F8 1DC00008 003BB6A0   00440200 003BB600 1DC00008 003BB6A8  *.......8........................*
019A73C0   00440200 003BB608 1DC00008 003BB6B0   00440200 003BB610 1DC00008 003BB6B8  *................................*
019A73E0   00440200 003BB578 1DC00008 003BB6C0   00440200 003BB580 1DC00008 003BB6C8  *...............................H*
019A7400   00440200 003BB588 1DC00008 003BB6D0   00440200 003BB590 1DC00008 003BB6D8  *...............................Q*
019A7420   00440200 003BB598 1DC00008 003BB6E0   00440200 003BB5A0 1DC00008 003BB6E8  *...............................Y*
```

Figure 13. Media Manager—Summary Dump (Part 1 of 3)

```
COMPON=MEDIA MANAGER, COMPID=28415, ISSUER=ICYFRR           MODULE IEAVTSDT DATE 09/14/87  TIME 10.50.41  PAGE 00000030
019A7440   00440200 003BB5A8 1DC00008 003BB6F0   00440200 003BB5B0 1DC00008 003BB6F8  *...............0...............8*
019A7460   00440200 003BB5B8 1DC00008 003BB700   00440200 003BB5C0 1DC00008 003BB708  *................................*
019A7480   00440200 003BB5C8 1DC00008 003BB710   00440200 003BB5D0 1DC00008 003BB718  *.......H........................*
019A74A0   00440200 003BB5D8 1DC00008 003BB720   00440200 003BB5E0 1DC00008 003BB728  *.......Q........................*
019A74C0   00440200 003BB5E8 1DC00008 003BB730   00440200 003BB5F0 1DC00008 003BB738  *.......Y...............0........*
019A74E0   00440200 003BB5F8 1DC00008 003BB740   00440200 003BB600 1DC00008 003BB748  *.......8...... ................*
019A7500   00440200 003BB608 1DC00008 003BB750   00440200 003BB610 1DC00008 003BB758  *................................*
019A7520   00440200 003BB578 1DC00008 003BB760   00440200 003BB580 1DC00008 003BB768  *................................*
019A7540   00440200 003BB588 1DC00008 003BB770   00440200 003BB590 1DC00008 003BB778  *................................*
019A7560   00440200 003BB598 1DC00008 003BB780   00040200 003BB5A0 002DC000 00000000  *................................*
019A7580   002DC200 00000000 002DC400 00000000   002DC600 00000000 002DC800 00000000  *..B.......D.......F.......H.....*
019A75A0   002DCA00 00000000 002DCC00 00000000   002DCE00 00000000 000F2000 00000000  *................................*
019A75C0   000F2200 00000000 000F2400 00000000   000F2600 00000000 000F2800 00000000  *................................*
019A75E0   000F2A00 00000000 000F2C00 00000000   000F2E00 00000000 00103000 00000000  *................................*
019A7600   00103200 00000000 00103400 00000000   00103600 00000000 00000002 01000200  *................................*
019A7620   00000002 02000200 00000002 03000200   00000002 04000200 00000002 05000200  *................................*
019A7640   00000002 06000200 00000002 07000200   00000002 08000200 00000002 09000200  *................................*
019A7660   00000002 0A000200 00000002 0B000200   00000002 0C000200 00000002 0D000200  *................................*
019A7680   00000002 0E000200 00000002 0F000200   00000002 10000200 00000002 11000200  *................................*
019A76A0   00000002 12000200 00000002 13000200   00000002 14000200 00000002 15000200  *................................*
019A76C0   00000002 16000200 00000002 17000200   00000002 18000200 00000002 19000200  *................................*
019A76E0   00000002 1A000200 00000002 1B000200   00000002 1C000200 00000002 1D000200  *................................*
019A7700   00000002 1E000200 00000002 1F000200   00000002 20000200 00000002 21000200  *................................*
019A7720   00000002 22000200 00000002 23000200   00000002 24000200 00000002 25000200  *................................*
019A7740   00000002 26000200 00000002 27000200   00000002 28000200 00000002 29000200  *................................*
019A7760   00000002 2A000200 00000002 2B000200   00000002 2C000200 00000002 2D000200  *................................*
019A7780   00000002 2E000200 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
019A77A0   00000000 00000000                                                           *........                        *
```

Figure 13. Media Manager—Summary Dump (Part 2 of 3)

```
 -------------- SUMMARY DUMP OUTPUT FOR HASID = 000F PASID = 000F SASID = 000F PSW = 070C0000 810070A4


----- SDWA ----------------- ASID FFFF.


 -SDWA----------AT LOCATION  00F7DB88
  -------------------------

+0000      00000CE8 940C1000 00000000 00000000   00000000 00000000 00000000 00C30B70  *...Y.........................C..*
+0020      00C30C18 019A709C 00BA8000 5001BFD8   810013F1 00FA2138 019A72A0 00000000  *.C.............Q...1............*
+0040      81008DA2 019A7020 019A709C 019A709C   81008EF4 00000004 00000000 00000000  *..................4............*
+0060      00000000 00000000 070C1000 00008EF6   00020001 00D69000 070C0000 81006D90  *...............6.....O..........*
+0080      00020001 00D69000 00000000 00C30B70   00C30C18 019A709C 00BA8000 5001BFD8  *.....O.......C...C............Q*
+00A0      810013F1 00FA2138 019A72A0 00000000   81008DA2 019A7020 019A709C 019A709C  *...1..........................*
+00C0      81008EF4 00000004 00000000 00000000   00000000 00000000 00000000 00000000  *...4..........................*
+00E0      00000000 00000000 40030001 00000040   00000000 00F7DD1C 00000000 00800000  *........ ...... .....7........*
+0100      00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
+0120      000F001D C9C5C1D5 E4C3F0F1 C9C3E8D7   C6C1D7D7 C9C3E8C6 D9D94040 00F7DCC8  *....IEANUC01ICYPFAPPICYFRR  .7.H*
+0140      00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
+0160      00000000 00000000 00000000 FFFF0003   00F7DEE8 8000000F 0000000F 00000000  *................7.Y............*
+0180      C9C3E8D7 C6D4E340 00000000 0005F509   00334031 C3D6D4D7 D6D57ED4 C5C4C9C1  *ICYPFMT ......5... .COMPON.MEDIA*
+01A0      40D4C1D5 C1C7C5D9 6B40C3D6 D4D7C9C4   7EF2F8F4 F1F56B40 C9E2E2E4 C5D97EC9  * MANAGER. COMPID.28415. ISSUER.I*
+01C0      C3E8C6D9 D9000000 00000000 00000000   00000000 00000000 00000000 00000000  *CYFRR..........................*
+01E0      00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
+0200      00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
+0220      00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
+0240      00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
+0260      00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
+0280      00000000 00000000 00000000 00000000   000000E2 C4E6C140 F2F8F4F1 F5D4C5C4  *..................SDWA 28415MED*
+02A0      C9C140D4 C1D5C1C7 C5D96B40 C9C3E8D7   C6C1D7D7 00000000 00000000 00000000  *IA MANAGER. ICYPFAPP...........*
+02C0      00000000 00301314 C9C3E8C6 D9D94040   F5F6F6F5 00400000 00000001 00000000  *........ICYFRR  5665. ..........*
+02E0      00000000 81006D90 019A709C 00000010   00000000 00000000 00000000 00000000  *................................*
+0300      00000000 04000800 48000000 0000000F   008D61A8 92CF0826 00CF0838 00000001  *................................*
+0320      00F79080 0026D07F 940C1000 00000001   00000000 00000000 00000000 00000000  *.7.............................*
+0340      00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
+0360      00F7DF00 00F7DE20 00F7DEB8 00F7DFF0   00F7DEC8 00000000 00000000 00000000  *.7...7...7...7.0.7.H...........*
+0380      00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
+03A0      00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
+03C0      00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
```

Figure 13. Media Manager—Summary Dump (Part 3 of 3)

Media Manager Return Codes

When the media manager is invoked and errors are found or a termination routine is called, the media manager provides a return code to identify the nature of the failure.

The media manager provides a return code in register 15. It also puts this return code in the MMPRETCD field of the MMPB.

Return codes have the following format:

```
cccffss

 where:
 cccc is the return code
 ff identifies the module that detected the error
 ss is the status byte
```

The status byte in the *ss* field has the following meanings:

ss field	Error Description
X'00'	No error.
X'04'	Warning.
X'08'	Extent error.
X'0C'	Logic error.
X'10'	Permanent I/O error.
X'14'	Error cannot be determined.

In combination with one of these status bytes, return codes in the *cccc* field indicate the following errors (the *ff* field value has been listed where this value is permanent):

ss	cccc	ff	Error Description
00	0000	—	No errors.
04	0000	—	No errors. DIE exit is receiving control in SRB mode.
	0004	0A	For MMCALL OP=RDWR VOLATILE=YES, device is not volatile cache or user does not have volatile cache token (MMIVCT=0). Normal I/O (without use of volatile cache) is done instead. The return code will be returned at the front end in MMPRETCD.
08	0010	—	The end of a CI is out of its extent.
	0014	—	Extent not found.
	0018	—	Extent not active.
	0420	—	Overlapping extents.
	0424	—	Invalid ending RBA.
	0428	—	Block size is too large for the device.
	042C	—	Inconsistent CI sizes.
0C	0004	—	Two writes to the same RBA.
	0004	18	The master MMIB was not passed, or there was no MMRE chain, or the user storage was not passed in.
	0004	19	GETMAIN Failure.

ss	cccc	ff	Error Description
	0008	—	Two reads to the same buffer.
	0008	19	PURGE Failure.
	000C	19	The master MMIB was not passed.
	000C	—	An MMRE specifies multiple CIs to be sorted.
	0010	—	MMRE specifies output but MMIB specifies input only.
	0010	17	MMIB specifies READ only, on an update write request.
	0016	02	Format write requested for replicated indexes. A replicated index format write request is not valid.
	0016	03	Format write requested for replicated indexes. A replicated index format write request is not valid.
	0016	15	Format write requested for replicated indexes. A replicated index format write request is not valid.
	001C	—	Load real address error. Buffers not fixed.
	001C	02	Extended format request not allowed.
	001C	03	Extended format request not allowed.
	0020	—	Starting RBA not on CI boundary—preformat only.
	0024	—	Insufficient storage for channel program because of media manager storage constraints, or invalid request element—starting RBA is greater than ending RBA specified in MMRE for MMCALL OP=FORMAT.
	0028	15	MMCALL OP=WRITE was specified with format write and VOLATILE=YES.
	0030	14	No volatile cache token provided for MMCAL OP=COMMIT. **Note:** Volatile cache token is ignored by OP=DISCARD.
	0034	14	MMIB not pointing to an EDB.
	0038	14	Error in media manager building of AOMSERV parameter list for MMCALL OP=COMMIT or OP=DISCARD.
	003C	14	Error in media manager building of extent address list for MMCALL OP=COMMIT or OP=DISCARD.
	0040	—	The real buffer address list contains addresses that are zeros.
	0044	—	The real buffer address list contains addresses that are not at 4K-byte boundaries.
	0048	—	Real buffer address list is provided for CIs that are not n*4K bytes.
	0400	—	Getmain error for LPMB.
	0408	—	Insufficient storage supplied for LPMB.
	0410	—	Getmain error for EDB.
	0418	—	Insufficient storage supplied for EDB.
	041C	40	MMINIT with MIEXTEND=ON issued without any previous MMINIT.
	0420	40	MIFXRBA set by MMINIT EXTEND=OFF.
	0424	40	MMINIT with MIEXTEND=ON can only be issued for one volume whose LPMB address is in MMIB.
	0428	40	MMINIT following an MMINIT with MIEXTEND=ON must also be MMINIT MIEXTEND=ON or MIVCTK=ON.
	042C	40	Volatile token not obtained for MMINIT with MIVOLT=ON because control unit does not have not volatile cache.

ss	cccc	ff	Error Description
	0430	40	Volatile token not obtained for MMINIT with MIVOLT=ON because error in building AOMSERV parameter list.
	0434	—	The parameter list is incomplete.
	0438	40	Invalid extended format request.
	043C	40	Stripe number mismatch.
	0440	40	Invalid number of stripes.
	0444	40	Invalid device block specified for the extended format data set.
10	0010	—	I/O prevented.
	0020	11	Permanent I/O error.
	0030	14	Extent address list missing for MMCALL OP=COMMIT or OP=DISCARD.
	0034	14	Volatile cache token has become invalid—data in cache may be lost before MMCALL OP=COMMIT request occurs.
	0038	14	Invalid track format error—data set destroyed or logic error in caller of media manager.
	003C	14	Unable to destage data for MMCALL OP=COMMIT.
	0040	14	MMCALL OP=COMMIT or OP=DISCARD cannot be done because control unit is not volatile cache.
	0040	0C	Hardware pad errors, detected on extended format data set only.
	0050	11	Hardware errors caused records to be skipped for the read I/O.
	0050	0C	RBN errors detected in MMRE chain.
	0410	40	AOM failed to obtain volatile token because of AOM/hardware errors.
14	0030	—	Error cannot be determined.
	0034	—	Indeterminate error—I/O has been started.

Note: The first byte in the *cccc* field X'00' indicates MMCALL or MMCNVT requests; X'04' indicates MMINIT requests.

The *ff* field of the return code identifies the module that detected the error.

ff field	Module
—	ICYMMSRV
—	ICYMMVT
—	ICYMSTAE
01	ICYRDWR
02	ICYFW
03	ICYPFMT
04	ICYRBA
05	ICYCCHHR
06	ICYSTOR
07	ICYSORT
08	ICYBLIST
09	ICYELE

ff field	Module
0A	ICYBLDCP
0B	ICYPFCP
0C	ICYDIE
0D	ICYPFDIE
0E	ICYNRM
0F	ICYABN
10	ICYPFAPP
11	ICYPGAD
12	ICYPURG
14	ICYVCACH
15	ICYWRITE
16	ICYBMFI
17	ICYPIOCP
18	ICYREPCP
40	ICYINIT
41	ICYIEDB
42	ICYILPMB

Chapter 11. OAM Diagnostic Aids

The following major diagnostic aids are provided here for the Object Access Method (OAM):

- Procedures for retrieving information from the OAM DB2 Databases
- A listing of the return and reason codes that are generated by the OSREQ macro
- A listing of the return and reason codes that are generated by the OAM macro
- A listing of the return and reason codes that are generated and invoked through the CBRXLCS macro
- A listing of Library Automation Communication Services (LACS) return and reason codes
- OAM Records Written to SYS1.LOGREC
- An explanation of how to dump OAM
- An explanation of how to use the SMS trace facility for gathering OAM diagnostics
- An explanation of OAM IPCS processing
- Descriptions of some common DB2 problems when using OAM and some related DB2 service aids

Before using these diagnostic aids, be sure that all of the installation requirements have been met. Verify that you have performed all of the following tasks:

- Properly installed your optical and tape library hardware
- Installed OAM correctly (see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* and *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*)
- If using optical storage, defined the DB2 database
- If using tape libraries, defined the tape configuration database (TCDB)
- Activated the IODF with your hardware configuration
- Started the OAM address space
- Entered at least one cartridge into each library

© Copyright IBM Corp. 1979, 2002

Retrieving Information from the OAM DB2 Databases for Diagnosis

The storage administrator can use SPUFI to retrieve information from the object directory and object tables. This might be helpful in diagnosing problems. SQL statements can retrieve a variety of information, as follows:

- Object directory

  ```
  SELECT * FROM GROUPxx.V_OSM_OBJ_DIR WHERE ODNAME = 'yy';
  ```

- 4K object table

  ```
  SELECT * FROM GROUPxx.V_OSM_04K-OBJ_TBL WHERE OTNAME = 'yy';
  ```

- 32K object table

  ```
  SELECT * FROM GROUPxx.V_OSM_32K_OBJ_TBL WHERE OTNAME = 'yy';
  ```

- Management class conversion table

  ```
  SELECT * FROM OAMADMIN.CBR_MGT_CLASS_TBL;
  ```

- Storage class conversion table

  ```
  SELECT * FROM OAMADMIN.CBR_STO_CLASS_TBL;
  ```

- Collection names conversion table

  ```
  SELECT * FROM OAMADMIN.CBR_COLLECTION_TBL;
  ```

- Optical Configuration Database

  ```
  SELECT * FROM XXXXXXXX.TAPEVOL;
  ```

where:

xx - Storage group number from 00-99
yy - Object name.
xxxxxxxx is the qualifier of the ID which created the table (this information can be obtained from SYSIBM.SYSTABLES).

The storage administrator can also use IDCAMS LISTCAT function to retrieve information on the collection names in the catalog.

Note: The information from the preceding tables is crucial to the operation of OSMC. IBM strongly advises against altering these tables.

OSREQ Return and Reason Codes

OAM issues return and reason codes during processing of OSREQ macro functions. At the macro exit, register 15 will contain one of the values shown in Table 72.

Table 72. OSREQ Return Codes

Value	Meaning
0 (X'00')	The requested function was successfully completed.
4 (X'04')	The requested function was only partially completed.
8 (X'08')	The requested function was not completed; the request was incorrectly issued.
12 (X'0C')	The requested function was not completed; there is an error in the environment.
16 (X'10')	The requested function was not completed due to a programming error.

OAM also places a 4-byte reason code in register 0. Table 73 presents the reason codes that are generated by the OSREQ macro. The 4 bytes are defined as follows:

Byte 0 Contains a unique OSREQ reason code.

Byte 1 Contains an internal OSREQ function code used by IBM service representatives in debugging programming errors. This byte is identified as **x** in the following table.

Bytes 2 and 3 These bytes are identified as **y** and **z** in the following table. They represent either a 2-byte field **yz**, or two single-byte fields **y** and **z**, where:

y is an error indicator.

z is reserved for IBM use, unless otherwise indicated.

Table 73. OSREQ Return and Reason Codes

Return Code	Reason Code (Bytes)				Description
	0	1	2	3	
0 (X'00')	X'00'	X'00'	X'00'	X'00'	Request successfully completed. No action required.
4 (X'04')	X'04'	x	y	z	Request completed with a warning condition. Correct program, if necessary.
4 (X'04')	X'04'	x	X'01'	z	QEL buffer segments are too short to accommodate all available entries; as many entries that could fit in the segment(s) are returned. Execute query with a larger QEL buffer.
4 (X'04')	X'04'	x	X'02'	z	Query element list is incomplete due to unavailable database(s). Activate database(s), if necessary.
4 (X'04')	X'04'	x	X'03'	z	UNACCESS has completed; the token has been cleared; there is one or more requests outstanding; the outstanding requests were not affected by the UNACCESS. Copies of the token are no longer valid. When the last outstanding request associated with the token is completed, the remaining resources will be freed. Correct the program, if necessary.

Table 73. OSREQ Return and Reason Codes (continued)

Return Code	Reason Code (Bytes) 0	1	2	3	Description
4 (X'04')	X'04'	x	X'04'	z	Store or change has completed but one or more of the following conditions occurred, as indicated by bits set in byte 3 (z). Z=BIT MAP: **1***xxx xxxx* Catalog entry was created for the collection *x***1***xx xxxx* Reserved *xx***1***x xxxx* Storage class specified for the collection was overridden *xxx***1** *xxxx* Management class specified for the collection was overridden *xxxx* **1***xxx* Retention period specified for the object was overridden *xxxx x***1***xx* Reserved *xxxx xx***1***x* Storage class specified for the object was overridden *xxxx xxx***1** Management class specified for the object was overridden Issue query to see new parameters, if desired.
4 (X'04')	X'04'	x	X'05'	z	DB2 SQL return code conversion module DSNTIAR not found.
4 (X'04')	X'04'	x	X'06'	z	Automatic Access Backup active, primary object on an unavailable resource, backup object retrieved successfully.
4 (X'04')	X'04'	x	X'07'	z	Second backup copy is retrieved. Primary copy is not available with access backup active.
8 (X'08')	X'24'	x	y	z	Parameter is unusable, incorrect, invalid, or incomplete. **y** First parameter with an error **z** Type of error Correct calling program.
8 (X'08')	X'24'	x	X'01'	X'01'	Parameter list (MF=L) in unusable storage.
	X'24'	x	X'01'	X'02'	Invalid value in parameter list (MF=L). Correct calling program.
8 (X'08')	X'24'	x	X'02'	X'01'	SIZE word in unusable storage.
	X'24'	x	X'02'	X'02'	SIZE word invalid. Correct calling program.
8 (X'08')	X'24'	x	X'03'	X'01'	RETPD area in unusable storage.
	X'24'	x	X'03'	X'02'	RETPD invalid value. Correct calling program.

Table 73. OSREQ Return and Reason Codes (continued)

Return Code	Reason Code (Bytes) 0	1	2	3	Description
8 (X'08')	X'24'	x	X'04'	X'01'	STORCLAS area in unusable storage.
	X'24'	x	X'04'	X'02'	STORCLAS invalid value.
	X'24'	x	X'04'	X'03'	STORCLAS name length invalid value.
					Correct calling program.
8 (X'08')	X'24'	x	X'05'	X'01'	MGMTCLAS area in unusable storage.
	X'24'	x	X'05'	X'02'	MGMTCLAS invalid value.
	X'24'	x	X'05'	X'03'	MGMTCLAS name length invalid value.
					Correct calling program.
8 (X'08')	X'24'	x	X'06'	X'01'	QEL buffer list in unusable storage.
	X'24'	x	X'06'	X'02'	QEL buffer list contains incorrect value (for example, incorrect ID, incorrect length field, incorrect version field, reserved bit is on).
	X'24'	x	X'06'	X'04'	QEL buffer in unusable storage.
					Correct calling program.
8 (X'08')	X'24'	x	X'07'	X'01'	Reason code area in unusable storage.
	X'24'	x	X'07'	X'02'	Return code area in unusable storage.
	X'24'				Correct calling program.
8 (X'08')	X'24'	x	X'08'	X'01'	BUFLIST in unusable storage.
	X'24'	x	X'08'	X'02'	BUFLIST contains incorrect value (for example, incorrect ID, incorrect length field, incorrect version field, reserved bit is on).
	X'24'	x	X'08'	X'04'	Buffer in unusable storage.
	X'24'	x	X'08'	X'05'	Amount of buffer data provided on store request is less than specified size of object.
	X'24'	x	X'08'	X'06'	Amount of buffer data provided on the store request is greater than specified size of object.
	X'24'	x	X'08'	X'08'	Amount of buffer data space provided on the retrieve request is insufficient for object.
					Correct calling program.
8 (X'08')	X'24'	x	X'09'	X'01'	Token area in unusable storage.
	X'24'	x	X'09'	X'02'	Token contains invalid contents.
					Correct calling program.
8 (X'08')	X'24'	x	X'0A'	X'01'	Name in unusable storage.
	X'24'	x	X'0A'	X'02'	Fully qualified name is required.
	X'24'	x	X'0A'	X'03'	Qualifier longer than 8 characters.
	X'24'	x	X'0A'	X'05'	Null qualifier.
	X'24'	x	X'0A'	X'06'	More than one asterisk.

Table 73. OSREQ Return and Reason Codes (continued)

Return Code	Reason Code (Bytes) 0	1	2	3	Description
	X'24'	x	X'0A'	X'07'	First character in name or qualifier is not alphabetic.
	X'24'	x	X'0A'	X'08'	Blank other than at right end.
	X'24'	x	X'0A'	X'09'	Name length is 0 or more than 44.
					Correct calling program.
8 (X'08')	X'24'	x	X'0B'	X'02'	First positional parameter code for first positional parameter is invalid.
8 (X'08')	X'24'	x	X'0C'	X'01'	Offset in unusable storage.
	X'24'	x	X'0C'	X'02'	Offset value is larger than length of object.
	X'24'	x	X'0C'	X'03'	Offset value is negative.
8 (X'0D')	X'24'	x	X'0D'	X'01'	Length in unusable storage.
	X'24'	x	X'0D'	X'02'	Length value is larger than offset value plus the length of the object from offset.
	X'24'	x	X'0D'	X'03'	Length value is negative.
					Correct calling program.
8 (X'24')	X'24'	x	X'0E'	X'01'	MSGAREA in unusable storage.
	X'24'	x	X'0E'	X'02'	MSGAREA length value negative.
					Correct calling program.
8 (X'08')	X'24'	x	X'0F'	X'01'	The collection name passed to OAM on the OSREQ macro is in unusable storage. This means that OAM encountered a virtual storage translation exception (that is, 0C4 abend) when it attempted to reference the area of storage containing the collection name or the collection name length.
	X'24'	x	X'0F'	X'02'	The collection name passed to OAM on the OSREQ macro is not fully qualified. The collection name contains an asterisk (*) as the last character in the name.
	X'24'	x	X'0F'	X'03'	The collection name passed to OAM on the OSREQ macro contains a qualifier longer than 8 characters.
	X'24'	x	X'0F'	X'04'	The collection name passed to OAM on the OSREQ macro contains an invalid character. One of the characters in the collection name is not an uppercase alphabetic (A – Z), numeric (0 – 9), or national (@, #, $) character.
	X'24'	x	X'0F'	X'05'	The collection name passed to OAM on the OSREQ macro contains a null qualifier. This means that one of the following is true: • The first character of the collection name is a period. • The last character of the collection name is a period. • The collection name contains two consecutive periods.
	X'24'	x	X'0F'	X'06'	Reserved.
	X'24'	x	X'0F'	X'07'	Collection name passed to OAM on the OSREQ macro contains an invalid qualifier. One of the qualifiers does not start with an uppercase alphabetic character (A – Z) or national character (@, #, $).

Table 73. OSREQ Return and Reason Codes (continued)

Return Code	Reason Code (Bytes) 0	1	2	3	Description
	X'24'	x	X'0F'	X'08'	The collection name passed to OAM on the OSREQ macro contains an imbedded blank.
	X'24'	x	X'0F'	X'09'	The collection name passed to OAM on the OSREQ macro has an invalid length. The collection name length is zero or longer than 44 characters. Correct calling program.
8 (X'08')	X'24'	x	X'10'	X'10'	IADDRESS points to unusable storage. Correct calling program.
8 (X'08')	X'24'	x	X'11'	X'01'	TTOKEN area in unusable storage. Correct calling program.
8 (X'08')	X'28'	x	y	z	IADDRESS routine error. **xyz** System/User completion code. Correct calling program.
8 (X'08')	X'2C'	x	y	z	No valid object found.
8 (X'08')	X'2C'	x	X'01'	z	Directory entry not found.
	X'2C'	x	X'02'	z	Object segment(s) not found.
	X'2C'	x	X'03'	z	Backup copy was not found. Correct calling program.
	X'2C'	x	X'04'	z	Second backup copy was not found.
8 (X'08')	X'30'	x	y	z	Object already exists. Correct calling program.
8 (X'08')	X'30'	x	X'01'	z	Directory entry already exists.
	X'30'	x	X'02'	z	Object segment already exists. The directory entry did not exist for this object, however a data segment already exists in the DASD 4K or 32K table for this object.
8 (X'08')	X'34'	x	y	z	Request rejected for this task.
8 (X'08')	X'34'	x	X'01'	z	Request issued from TCB other than initial access request TCB.
	X'34'	x	X'02'	z	Access issued from TCB with prior access still active. Correct calling program.
12 (X'0C')	X'60'	x	y	z	OTIS interface error detected.
	X'60'	x	X'01'	X'01'	DB2 service inactive.
	X'60'	x	X'01'	X'02'	OTIS not initialized.
	X'60'	x	X'01'	X'03'	OTIS not initialized, or error establishing OTIS environment.
	X'60'	x	X'02'	X'04'	Application running in unsupported key.
	X'60'	x	X'02'	X'05'	Application supplied parameter in fetch protected storage.

Table 73. OSREQ Return and Reason Codes (continued)

Return Code	Reason Code (Bytes) 0	1	2	3	Description
	X'60'	x	X'02'	X'06'	Reserved for IBM use.
	X'60'	x	X'02'	X'07'	Application supplied a parameter that is not in the PRIVATE area. Correct application program.
	X'60'	x	X'03'	X'0B'	Error during collection table update. DB2 not connected.
	X'60'	x	X'03'	X'0C'	Error during collection table update. DB2 thread not OPEN.
12 (X'0C')	X'64'	x	y	z	CICS/DB2 communication failure. DB2 not initialized, or CICS not connected to DB2. Correct DB2 connection to the CICS region.
12 (X'0C')	X'68'	x	y	z	OSR not initialized. Correct the environment.
12 (X'0C')	X'68'	x	X'01'	z	Control blocks not initialized.
	X'68'	x	X'02'	z	OSR not enabled as CICS resource manager.
12 (X'0C')	X'6C'	x	y	z	OAM address space detected failure. **yz** LCS reason code. See “OAM Macro Return and Reason Codes” on page 206 and look for the associated OAM macro reason code. The return code 12 is an OSREQ return code, and may not match the return code for the OAM macro.
12 (X'0C')	X'6D'	x	y	z	OAM address space detected failure: • Primary copy was not available for retrieve (on unreadable volume, in nonoperational library, or in offline library). • Automatic access to backup was active for BACKUP2. • Retrieve of second backup copy failed.
12 (X'0C')	X'6E'	x	y	z	OSMC/OAM address space detected failure. Automatic Access Backup active, primary object on an unavailable resource, retrieve of the backup copy of the object failed. **yz** LCS reason code. See “OAM Macro Return and Reason Codes” on page 206 and look for the associated OAM macro reason code. The return code 12 is an OSREQ return code, and may not match the return code for the OAM macro.
12 (X'0C')	X'6F'	x	X'01'	z	Automatic Access Backup active, primary object on an unavailable resource, no backup object available.
12 (X'0C')	X'6F'	x	X'02'	z	An OSREQ retrieve was received. Primary copy was not available. Access backup active for BACKUP2. However, a second backup copy does not exist for the object.
12 (X'0C')	X'70'	x	y	z	mf=lDB2 detected failure (media). yz represents the DB2 SQL error. Convert the yz to decimal, and look up the resulting SQL code in DB2 Messages and Codes. **yz** DB2 SQL code

Table 73. OSREQ Return and Reason Codes (continued)

Return Code	Reason Code (Bytes)				Description
	0	1	2	3	
12 (X'0C')	X'74'	x	y	z	DB2 detected failure (other). yz represents the DB2 SQL error. Convert the yz to decimal, and look up the resulting SQL code in DB2 Messages and Codes. **yz** DB2 SQL code
12 (X'0C')	X'78'	x	y	z	DB2 detected inadequate DASD space. yz represents the DB2 SQL error. Convert the yz to decimal, and look up the resulting SQL code in DB2 Messages and Codes. **yz** DB2 SQL code OSMC was notified so that it can perform DASD space management.
12 (X'0C')	X'7C'	x	y	z	Insufficient virtual storage
12 (X'0C')	X'80'	x	y	z	Database contents inconsistent.
12 (X'0C')	X'80'	x	X'01'	z	Directory entry location invalid.
	X'80'	x	X'02'	z	Directory entry object size not equal to sum of segment sizes.
	X'80'	x	X'03'	z	The object size returned by OAM interface not equal to size requested.
12 (X'0C')	X'84'	x	y	z	Caller not authorized. yz represents the DB2 SQL error. Convert the yz to decimal, and look up the resulting SQL code in DB2 Messages and Codes. **yz** DB2 SQL code
12 (X'0C')	X'88'	x	y	z	Resource not available. yz represents the DB2 SQL error. Convert the yz to decimal, and look up the resulting SQL code in DB2 Messages and Codes. **yz** DB2 SQL code
12 (X'0C')	X'8C'	x	y	z	Deadlock or time out occurred. Database rolled back to previous synchronization point. yz represents the DB2 SQL error. Convert the yz to decimal, and look up the resulting SQL code in DB2 Messages and Codes. **yz** DB2 SQL code
12 (X'0C')	X'90'	x	y	z	Deadlock or time out occurred. Database not rolled back to previous synchronization point. yz represents the DB2 SQL error. Convert the yz to decimal, and look up the resulting SQL code in DB2 Messages and Codes. **yz** DB2 SQL code
12 (X'0C')	X'94'	x	y	z	OTIS DB2 error during collection table processing. yz represents the DB2 SQL error. Convert the yz to decimal, and look up the resulting SQL code in DB2 Messages and Codes. **yz** DB2 SQL code
12 (X'0C')	X'A0'	x	y	z	SMS error during SMS review. **yz** SMS reason code in hexadecimal, see “Storage Management Subsystem Return and Reason Codes” on page 437.
12 (X'0C')	X'A4'	x	y	z	DB2 error during SMS interface utility processing. yz represents the DB2 SQL error. Convert the yz to decimal, and look up the resulting SQL code in DB2 Messages and Codes. **yz** DB2 SQL code

Table 73. OSREQ Return and Reason Codes (continued)

Return Code	Reason Code (Bytes) 0	1	2	3	Description
12 (X'0C')	X'A8'	x	y	z	Internal error during SMS review. **y** Error indicator
12 (X'0C')	X'A8'	x	X'01'	z	Invalid function code in CBRSMSI.
	X'A8'	x	X'02'	z	Parameter list error.
	X'A8'	x	X'03'	z	SMS subsystem not active.
	X'A8'	x	X'04'	x	SMS subsystem not installed.
	X'A8'	x	X'05'	z	Error invoking SMS subsystem.
	X'A8'	x	X'06'	z	SMS is restarting.
	X'A8'	x	X'08'	z	Storage group not connected.
	X'A8'	x	X'09'	z	Constructs not assigned by SMS.
12 (X'0C')	X'AC'	x	X'01'	z	SMS not active.
	X'AC'	x	X'02'	z	Default DB2 name not in SMS control blocks.
12 (X'0C')	X'B0'	x	y	z	Request denied due to storage group state.
12 (X'0C')	X'B0'	x	X'01'	z	Storage group state is DISABLE NEW.
	X'B0'	x	X'02'	z	Storage group state is DISABLE ALL. Resubmit the request on a system where the group is enabled, or enable the group on this system, activate the modified SCDS, and resubmit the request.
12 (X'0C')	X'B4'	x	y	z	Collection definition errors.
12 (X'0C')	X'B4'	x	X'01'	z	Collection is not defined in catalog (on any request other than a store).
	X'B4'	x	X'02'	z	Collection is not defined in catalog but is defined in collection identifier table (store only).
	X'B4'	x	X'03'	z	Collection definition in catalog is incomplete and collection is not defined in collection identifier table (for example, IDCAMS was used to create a collection catalog entry which did not previously exist).
12 (X'0C')	X'BA'	x	y	z	Installation exit CBRUXSAE has denied user authorization for this request. **x** represents the OSREQ function code **y** represents the return code from the CBRUXSAE installation exit. **z** Reserved for IBM use.
16 (X'10')	X'CA'	x	y	z	An abend occurred.
16 (X'10')	X'CC'	x	y	z	OSR detected error. Contact your IBM service representative.
16 (X'10')	X'D0'	x	y	z	OAM detected programming error. **yz** OAM reason code, see “OAM Macro Return and Reason Codes” on page 206. and look for the associated OAM macro reason code. The return code 12 is an OSREQ return code, and may not match the return code for the OAM macro.

Table 73. OSREQ Return and Reason Codes (continued)

Return Code	Reason Code (Bytes)				Description
	0	1	2	3	
16 (X'10')	X'D2'	x	y	z	OAM Delete Object Table DB2 error. yz represents the DB2 SQL error. Convert the yz to decimal, and look up the resulting SQL code in DB2 Messages and Codes. **yz** DB2 SQL code
16 (X'10')	X'D4'	x	y	z	DB2 detected error. yz represents the DB2 SQL error. Convert the yz to decimal, and look up the resulting SQL code in DB2 Messages and Codes. **yz** DB2 SQL code
16 (X'10')	X'D8'	x	y	z	DB2 Call Attach Facility E/P DSNALI/DSNTIAR/DSNCLI not found when loaded.
16 (X'10')	X'DC'	x	y	z	DB2 Call Attach Facility error. **xyz** DB2 Call Attach Facility reason code (low 3 bytes)
16 (X'10')	X'E0'	x	y	z	SMSI detected error. **y** error indicator
16 (X'10')	X'E0'	x	X'01'	z	Catalog error detected on a superlocate request for a collection.
	X'E0'	x	X'02'	z	Catalog error detected on a define request for a collection.
	X'E0'	x	X'03'	z	Catalog error detected on an alter request for a collection.
	X'E0'	x	X'06'	z	Error issuing messages from SMS.
	X'E0'	x	X'07'	z	Reserved for IBM use.
	X'E0'	x	X'08'	z	Reserved for IBM use.
	X'E0'	x	X'0A'	z	Error on ATTACH macro.
	X'E0'	x	X'0C'	z	Reserved for IBM use.
	X'E0'	x	X'0D'	z	Unexpected error from DB2.

OAM Macro Return and Reason Codes

Although the OAM macro interface is an internal interface, the return and reason codes occasionally appear to the application invoking the OSREQ macro. The return codes are also contained in messages issued by OAM when they occur. For this reason, the return and reason codes are documented as shown in 74.

Table 74. OAM Macro Return Codes

Value	Description
0 (X'00')	Request completed successfully.
4 (X'04')	Request completed with warning condition. The reason code identifies the warning condition.
8 (X'08')	Invalid request. The reason code identifies the invalid parameter.
12 (X'0C')	Request failed. A required software or hardware resource is unusable or unavailable, a permanent I/O error has occurred, or lack of optical volume space has prevented the writing of an object. The reason code identifies the error type and the specific cause of the failure.
16 (X'10')	OAM failed. The attempted use of a system service failed with a return code other than zero, or an abend occurred during request processing or a logic error occurred. The reason code indicates the failing system service. The return code from the system service or the abend code is placed in the additional error information field OPLFERR in the OPL feedback area.

Table 75 presents the reason codes that are associated with each of the return codes.

Table 75. OAM Macro Reason Codes

Return Code	Error Code	Reason Code	Description
X'00'		X'0000'	Request was successfully scheduled.
X'04'	Warning	X'0001'	Object length less than buffer length for TYPE(READ).
		X'0002'	Object length greater than buffer length for TYPE(READ).
		X'0010'	Optical VTOC entry for object not in range given by STSECTOR.
		X'0020'	Partial read extends beyond object bounds.
		X'0030'	Optical volume specified on query request is a shelf volume.
		X'0031'	Tape volume specified on a query request is outside of an ATL. The volume could be inside of an MTL or on the shelf.
		X'0040'	Part of volume list operation was successfully scheduled.
		X'0041'	Part of a library audit was successfully scheduled.
X'08'	Invalid request	X'0400'	Invalid parameter list address.
		X'0401'	Invalid control block ID.
		X'0402'	Invalid VOLLIST parameter specified.
		X'0403'	Invalid volume type parameter specified.
		X'0404'	No volume type parameter specified.
		X'0405'	Invalid volume serial number specified.
		X'0406'	Invalid media type found in OAM.

Table 75. OAM Macro Reason Codes (continued)

Return Code	Error Code	Reason Code	Description
		X'0407'	Invalid library device type specified for an OAM request.
		X'0408'	Media type and device type cannot coexist.
		X'0410'	Invalid request type.
		X'0411'	TYPE(WRITE) not specified for chained request.
		X'0412'	Only WRITEs and READs can be canceled by invoking the OAM macro, TYPE CANCEL.
		X'0420'	Object name is missing.
		X'0421'	Collection name is missing.
		X'0422'	Library name is missing.
		X'0423'	Volume serial number is missing.
		X'0424'	Sector number(if optical media) or 3480/3490 blockid (if tape media) is missing.
		X'0425'	Object size is missing.
		X'0426'	Scope is missing.
		X'0427'	Scope is invalid.
		X'0428'	SQLCAPTR is missing.
		X'0430'	Invalid volume option.
		X'0431'	Volume option does not match that of first OPL in chain.
		X'0432'	There are tape and optical OPLs in the same chain.
		X'0433'	An OPL has both volume types specified. OAM does not know if an object should go to optical or tape media.
		X'0440'	Volume serial number missing for TYPE(READ).
		X'0441'	Volume serial number not defined in optical configuration database.
		X'0442'	Volume serial number does not match that of first OPL in chain.
		X'0443'	Eject request already pending for specified volume.
		X'0444'	Invalid volume serial number on an eject request.
		X'0445'	Undefined volume serial number on an eject request.
		X'0446'	Volume to be ejected already on shelf.
		X'0447'	Volume to be ejected has some outstanding request or it has a mount pending.
		X'0448'	Library or I/O station not capable of handling this eject.
		X'0449'	The volume serial number specified is for a shelf resident volume.
		X'044A'	The volume serial specified is not listed in the tape configuration database.
		X'044B'	OAM already has an AUDIT outstanding for this volume.
		X'044C'	OAM already has an AUDIT outstanding for this library.
		X'044E'	OAM already has a REMAP outstanding for this library.
		X'044F'	The volume is not supported at this software level (volume record for this volume contains TDSI information that is not understood at this software level).
		X'0450'	Group name not defined in optical configuration database.
		X'0452'	Group name does not match that of first OPL in chain.
		X'0460'	Request for library, but no libraries defined in optical configuration database.
		X'0461'	Library name not defined in optical configuration database.
		X'0462'	NONGROUP volume set not defined in optical configuration database.
		X'0463'	Library name does not match that of first OPL in chain.
		X'0464'	Library name not defined in the tape configuration database.
		X'0465'	Library specified is a pseudo library.
		X'0466'	Library specified is an empty library.
		X'0467'	Library specified is a manual tape library.

Table 75. OAM Macro Reason Codes (continued)

Return Code	Error Code	Reason Code	Description
		X'0468'	A write request for tape media was received, but the OBJECT/OBJECT BACKUP storage group associated with the write has a blank TAPEUNITNAME.
		X'0471'	The OBJECT BACKUP storage group is not defined in the current active SMS configuration.
		X'0472'	A write request for a second backup copy was received. However, the storage group to which the object belongs does not have a second backup storage group defined.
		X'0480'	Invalid optical VTOC sector number.
		X'0481'	Invalid 3480/3490 blockid is specified for this request.
		X'0482'	Requested Object or Collection name not the same as the Object or Collection name in the prefix of the actual tape record at the specified blockid.
		X'0490'	Buffer address missing.
		X'0491'	Buffer length missing.
		X'0492'	One buffer parameter supplied and one missing for TYPE(READ).
		X'04A0'	One parameter is inconsistent for partial object read request on a TYPE(READ).
		X'04A1'	Partial read parameter invalid or exceeded object bounds.
		X'0500'	Invalid type specified on a QUERY request.
		X'0520'	Volume that was the subject of a QUERY (for optical or tape) was not found in the optical configuration database.
		X'0530'	Database error while processing some request.
		X'0540'	Volume to be reinitialized still contains active objects.
		X'0549'	Drive specified for REFORMAT not defined in SMS ACDS. Retry the REFORMAT using another drive or do not specify a drive, allowing OAM to select a drive.
		X'0550'	Volume RELABEL already in progress for volume specified on REFORMAT utility. If REFORMAT is still desired, retry after the RELABEL is complete.
		X'0551'	Volume REFORMAT already in progress for volume specified on REFORMAT utility.
		X'0552'	Volume REINIT already scheduled for volume specified on REFORMAT utility
		X'0554'	An active write is currently scheduled to the volume specified on REFORMAT utility request. If REFORMAT is still desired, wait until the write(s) complete, then use the MOVEVOL utility to move all active objects off the volume, then retry the REFORMAT request.
		X'0555'	At least one active object exists on a volume specified on a REFORMAT utility request. Use MOVEVOL utility to move all active objects off the volume, then retry the REFORMAT request.
		X'0556'	Drive specified is not an operator accessible drive. When the volume is shelf resident, the user can specify a target drive. When the volume is library resident, OAM must select the target drive.
		X'0557'	Volume specified on REFORMAT utility request is LMSI media. Only 3995 media is supported for the REFORMAT utility.
		X'0559'	Volume specified on REFORMAT request is an OBJECT BACKUP volume (belongs to the OBJECT BACKUP storage group). Only volumes belonging to OBJECT storage groups are supported for the REFORMAT utility.
X'0C'	Software resource unavailable	X'0800'	OAM address space not available.

Table 75. OAM Macro Reason Codes (continued)

Return Code	Error Code	Reason Code	Description
		X'0810'	Request sent to another OAM in an OAMPLEX to process and the request timed out (did not complete in the customer specified timeout interval in CBROAMxx). If this happens often, consider increasing the timeout value. Also interrogate RMF™ XCF transaction reports for potential performance problems in the coupling facility.
		X'0811'	An attempt was made to send the request to another OAM in an OAMPLEX to process, however XCF resources are not available (an error from XCF was received by OAM).
		X'0812'	Volume for this request is owned by another OAM in an OAMPLEX. An attempt will be made by OAM to send this request cross system, or for a write, select another volume.
X'0C'	Hardware resource unavailable	X'0900'	No eligible optical drive is capable of executing this request.
		X'0901'	OAM received a request to eject a volume, but that volume is mounted on a drive which is offline, pending offline, or not operational.
		X'0902'	OAM was unable to schedule a REMAP because one of the drives in subject library is not operational with a cartridge still mounted.
		X'0903'	No drives available on this OAM in an OAMPLEX for this request. This request was already sent cross-system once, so will not be sent again. This can happen if a request is sent cross system from one OAM to another in an OAMPLEX, but the library or drives on the target OAM are no longer available (for example, taken offline) when the request gets to the target OAM.
		X'0904'	OAM Optical Volume is LOST due to operator cancel of an outstanding mount.
		X'0909'	A specific request was made for unreadable volume-access backup activated.
		X'0910'	Specific request for unreadable volume.
		X'0911'	Specific write request for unwriteable volume.
		X'0912'	Specific request for scratch volume.
		X'0913'	Specific write request for write-protected volume in library.
		X'0914'	Specific request for library volume, but volume is not in assigned slot in library.
		X'0915'	Specific request for library volume, but volume is in a nonoperational drive.
		X'0916'	The cartridge for this request is stuck in its library slot.
		X'0917'	A duplicate volume was detected for this request.
		X'0918'	A defrag request was made for a write protected volume.
		X'0919'	Unable to read the volume label.
		X'091F'	Requesting address space or requesting task (TCB) terminated.
		X'0920'	The request was canceled by the operator.
		X'0921'	The volume required for this request could not be found to complete the pending mount.
		X'0922'	All objects on the volume required for this request expired, therefore, this volume has been deleted from the OAM volume inventory.
		X'0923'	The volume is mounted on a drive that is not write-compatible with the volume's media type.
		X'0930'	GETMAIN for object buffer in user address space failed.
		X'0940'	Object name not found in specified optical VTOC sector.
		X'0941'	Object name not found after search of entire optical VTOC.
		X'0942'	The requested object was not found on this cartridge.

Table 75. OAM Macro Reason Codes (continued)

Return Code	Error Code	Reason Code	Description
		X'0951'	SSAR (set secondary address space register) was performed, and the target address space was not found.
		X'0952'	A cross-memory move failed.
		X'0953'	A permanent I/O error occurred while writing to tape.
		X'0954'	A permanent I/O error occurred while trying to read from tape.
		X'0955'	Unable to OPEN a tape data set.
		X'0956'	RDJFCB failed to determine mounted VOLSER for MVS scratch mount.
		X'0957'	Read request failed due to a POINT macro failure.
		X'0958'	Write request failed due to a NOTE macro failure.
		X'0959'	Write request failed due to a SYNCDEV macro failure.
		X'0970'	Dynamic allocation failure while allocating a tape drive.
X'0C'	Permanent I/O error	X'0A00'	Permanent error on recording medium.
		X'0A01'	Request sent to an unformatted cartridge.
		X'0A02'	Cartridge was found to be physically incompatible.
		X'0A03'	The cartridge format was found to be unrecognizable.
		X'0A04'	The data portion of the cartridge was questionable.
		X'0A10'	Permanent error on optical drive.
		X'0A11'	OAM was unable to schedule a REMAP request because a demount in preparation for the REMAP failed.
		X'0A12'	OAM was unable to perform a unit of work because the tape drive that was allocated for the request experienced an error.
		X'0A18'	The library logical type field is blank because no library with this ID was physically attached to the host at IPL time. As a result, OAM does not know if the library corresponding to the logical type field is an ATL or an MTL.
		X'0A19'	OAM does not know anything about this library because no library with this name was defined in the current active configuration data set (ACDS).
		X'0A20'	Permanent error on optical library.
		X'0A21'	Library request purged after failure of prior request on same library.
		X'0A22'	Permanent I/O error on the channel to channel (CTC).
		X'0A23'	The I/O driver timed out while communicating with the library.
		X'0A24'	An undocumented fault code has been returned from the controller.
		X'0A29'	The Cartridge Eject Installation Exit is disabled.
		X'0A30'	None of the requests in this volume list operation were successfully scheduled.
		X'0A31'	None of the volumes in this library audit were successfully scheduled for auditing.
X'0C'	Out of space error	X'0B00'	Write request for specific volume, but not enough space on volume.
		X'0B50'	OAM was initialized with a null configuration.
		X'0B60'	Library specified is inaccessible–offline, pending offline, or nonoperational.
		X'0B61'	Tape library is in manual mode.
		X'0B62'	Tape library vision system is inoperative.
		X'0B63'	Remap is pending or in progress for this library.
		X'0B64'	A specific request (READ) for a volume failed because it is in a library which is offline or pending offline.

Table 75. OAM Macro Reason Codes (continued)

Return Code	Error Code	Reason Code	Description
		X'0B65'	A specific request (READ) for a volume was received and the library in which it resides is offline or pending offline. Access Backup is active for offline libraries, a read will be attempted for the backup copy of the object.
		X'0B66'	A specific request (READ) for a volume failed because it is in a library which is currently not operational.
		X'0B67'	A specific request (READ) for a volume was received and the library in which it resides is not operational. Access Backup is active for nonoperational libraries, a read will be attempted for the backup copy of the object.
		X'0B70'	The retry limit for the WTD was met.
		X'0B71'	Read of an object from tape failed due to a NOTE macro failure. Investigate any messages issued with NOTE macro error. Retry request after corrections or repairs are made.
		X'0B72'	Read of an object from tape failed. This is a general read failure. Investigate any messages issued with more specific information. Retry the request after corrections or repairs are made.
X'10'	OAM ABEND	X'0C00'	OAM ABEND during request processing.
X'10'	System service failure	X'0D00'	Nonzero return code from GETMAIN.
		X'0D10'	Nonzero return code from ESTAE.
		X'0D20'	Unable to make user's address space nonswappable.
		X'0D29'	OAM requested allocation of a tape device, but something other than an acceptable tape device was allocated.
		X'0D30'	Unable to add a control block to a queue.
		X'0D40'	Unable to update a volume table row.
		X'0D41'	Unable to delete a volume table row.
		X'0D44'	Database row contents in error.
		X'0D46'	OAM failed in its attempt to insert a row into TAPEVOL (tape volume) table in the optical configuration database.
		X'0D47'	OAM failed in its attempt to update a row into TAPEVOL (tape volume) table in the optical configuration database.
		X'0D50'	A TCDB (tape configuration database) catalog access error occurred.
		X'0D51'	A TCDB (tape configuration database) catalog authorization error occurred.
		X'0D60'	A DB2 error occurred when accessing the object directory table.
X'10'	OAM error	X'0E00'	Unspecified logical error during OAM processing.
		X'0E01'	An invalid library or drive command was received by the controller.
		X'0E02'	Packet incorrectly filled in.
		X'0E03'	Attempted to use side 2 of a single-sided cartridge.
		X'0E04'	Attempted to eject a cartridge with the I/O station already occupied.
		X'0E05'	The controller returned with an OS/2® error from the PC.
		X'0E06'	The controller returned that there are no more operational or available slots in the library.
		X'0E07'	The controller returned with the wrong volume found or volume misfiled.
		X'0E08'	The controller returned with volume not where expected or volume lost.
		X'0E09'	The controller returned with access denied.
		X'0E0A'	The volume is not in the outboard inventory.
		X'0E0B'	OAM was unable to find the matching optical drive control block.

Table 75. OAM Macro Reason Codes (continued)

Return Code	Error Code	Reason Code	Description
		X'0E0C'	I/O attempted to an offline CTC.
		X'0E0D'	Error attempting to write a full volume.
		X'0E0E'	Error attempting to write a write-protected volume.
		X'0E0F'	Request failed because the controller responded with a too many files open return code.

LCS External Services (CBRXLCS)

LCS External Services provides an internal and external interface to volume and library records in the tape configuration database (TCDB). The return and reason codes that may result from invocation of the CBRXLCS macro are shown in Table 76

CBRXLCS Return and Reason Codes

Notes:

1. The return code is placed in register 15 and in the LCSPL in field LCSRCODE.
2. The reason code is placed in register 0 and in the LCSPL in field LCSREAS.
3. If an asterisk (“*”) appears after a reason code in the table below, additional diagnostic information is available in the LCSPL.

Table 76. CBRXLCS Return and Reason Codes

Return Code	Error Type	Reason Code	Meaning
0	Successful	0	Successful execution.
4	Warning	4	Request was to change to scratch but volume was already scratch.
		5	Request was to change to private but volume was already private.
		8	Scratch volume threshold processing did not successfully complete. Check the console log for further diagnostic information.
		48	Storage group state is not connected, disabled new, or disabled all.
		51	Specific volume serial request for a scratch volume.
		52	Volume is not library resident.
		55	Volume is not eligible because it is library resident but the device is not defined to the same tape library.
		61	Unable to access library manager inventory. If VOLINFO keyword was specified, TVI contains only TCDB information.
		63	Volume record not found in the TCDB.
		70	Volume is not found in library manager inventory. If VOLINFO keyword was specified, TVI contains only TCDB information.
		120	Volume is not eligible because the type of media defined in the volume record cannot be mounted on the specified device.
		121	Volume is not eligible because the type of media specified in the TDSI does not match the media type in the volume record.
		122	Volume is not eligible because the volume record reflects an error status.
		123	Volume is ineligible because the specified recording technology cannot coexist with the volume media type or the specified drive type.
		130	Specified volume already resides in this same tape library.
		131	Scratch volume threshold processing not performed because library was not operational.

Table 76. CBRXLCS Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Meaning
		134	No volume record was found in the TCDB; volume resides in specified library. If VOLINFO was specified, TVI contains only library manager information.
		135	Volume resides in specified library; volume record library and specified library do not match. If VOLINFO was specified, TVI contains TCDB information and library manager information from the specified library.
		136	Volume does not reside in specified library; volume record library and specified library do not match. If VOLINFO was specified, TVI contains only volume TCDB information.
		137	Unable to access the library manager of the specified library; volume record library and specified library do not match. If VOLINFO was specified, TVI contains only volume TCDB information.
		138	Library scratch count was not updated in TCDB.
		139	Specified library and volume record MTL library name do not match.
8	Invalid request	9	Required type parameter not specified.
		10	Both storage group list and volume list input specified.
		11	Invalid value specified for type.
		12	Required function parameter not specified.
		13	Invalid value specified for function.
		14	Required use parameter not specified.
		15	Invalid value specified for use.
		16	Required volume parameter not specified.
		17	Invalid volume serial specified.
		18	Required UCB address not specified.
		19	Invalid address specified for UCB.
		20	Required volume list not specified.
		21	Invalid header value specified in volume list.
		22	Invalid header value specified in storage group list.
		23	Required library name parameter not specified.
		25	Invalid expiration date specified.
		26	Library not defined to active configuration.
		27	Invalid library name specified.
		28	Invalid write protection date specified.
		29	Invalid parameter list.
		30	LCSPL, volume list, or storage group list not aligned on fullword boundary.
		31	Required TDSI parameter not specified.
		32	Required library ID parameter not specified.
		33	Invalid library ID specified.
		34	Invalid storage group name specified.
		35	Required media type not specified for MCE volume.
		38	Invalid compaction type specified in TDSI.
		39	Invalid value specified for special attribute in TDSI.
		40	Invalid combination of TDSI values specified.
		41	Ambiguous TDSI combination specified.
		42	Special attribute specified where not allowed or not applicable.
		43	Invalid pointer to TDSI specified.
		44	Invalid value specified for the DISP keyword.
		47	Not all volumes in input list associated with same storage group.
		51	Specific volume serial request for a scratch volume.
		53	Library not defined to active configuration.
		54	Storage group not of type tape.

Table 76. CBRXLCS Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Meaning
		92	Library specified for the MCE function is not a manual tape library.
		94	Specified volume already resides in another library.
		97	Volume of same volser is a known DASD volume.
		201	Volume already ejected (not in library).
		202	Invalid value specified for eject option.
		203	Invalid value specified for bulk eject.
		204	Invalid TSO/E user ID specified.
		215	Not all volumes have same recording technology.
		216	MTL cannot be specified as LIBNAME with TCDBCHK= NO specification.
12	Failure	6	Request failed because change use attribute processing has been disabled.
		7	Request failed because installation exit vetoed the change.
		45	Library name not defined in tape configuration database.
		46	No enabled storage groups.
		48	Storage group state is not connected, disabled all, or disabled new.
		49	No device pools exist to fulfill request for TDSI specification.
		52	Volume is not library resident.
		53	Library not defined to active configuration.
		56	No library associated with input storage group(s).
		58	Failure accessing volume record in tape configuration database.
		59	Failure accessing library record in tape configuration database.
		60	Failure accessing SMS storage group constructs.
		61	Device services failure.
		62	Library control block for specified library ID not found.
		63	Volume record not found for requested volume.
		64	Unable to determine in which library the volume resides.
		65	Library logical type not defined in the active SMS configuration.
		66	No device pools exist to fulfill request for recording technology.
		67	No device pools exist to fulfill request for media type.
		69	Request failed because volume not in library installation exit (CBRUXVNL) said to cancel the job.
		70	Failure because volume not in library manager inventory.
		74	Request failed because volume serial number already exists in library manager inventory.
		75 *	Unexpected UCBSCAN error encountered while validating volume serial number.
		78 *	Error encountered when attempting to retrieve volume record.
		79 *	Error encountered when attempting to write volume record.
		82	Installation exit abended.
		83	Installation exit returned bad data.
		84 *	Abend occurred during LCS external services processing.
		95	Installation exit vetoed the entry of the volume into the library.
		96	Installation exit said to ignore the volume.
		300	OAM abend during eject request processing.
		302	Eject request already pending for volume.
		303	Unable to make user address space nonswappable.
		304	Tape configuration database access error.
		305	Tape configuration database authorization error.
		306	OAM internal error.
		307	Specified volume serial not in tape configuration database.
		310	Media type or recording technology not supported at this software level.

Table 76. CBRXLCS Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Meaning
		312	Requested function cannot coexist with the library
		313	Requested volume is currently in use
		314	Import/Export operation is already in progress or host processing of the operation is not complete.
		315	There are not enough physical drives available in the VTS to perform Import/Export operation
		316	Cancel operation for Import/Export failed because the operation is not in progress
		317	Import operation requested but there are no import volumes; Import/Export operation requested but there are no scratch physical volumes to perform the operation
		318	The maximum number of logical volumes is already defined to the VTS; therefore, no volumes can be imported into the library.
		319	Volume does not reside in specified library; no volume record retrieved. No TVI information is returned.
		320	Unable to access the library manager of the specified library; no volume record retrieved. No TVI information is returned.
		321	Manual cartridge entry failed; the media type specified by CBRUXENT does not match the media type stored in the existing volume record.
16	Environment	2	OAM control block structure not available.
		80	LCS external services unable to establish ESTAE.
		81	LCS external services unable to obtain storage.
		90	Cartridge entry disabled due to an error in the installation exit.
		91	Cartridge entry temporarily suspended.
		93	Library offline, pending offline, or not operational.
		400	OAM initialized with null configuration (no libraries).
		401	Library not accessible – offline, pending offline, or not operational.
		402	Vision system inoperative.
		403	Eject processing disabled because of an error in the installation exit
		404	OAM address space not available.

CBRXLIB Return Codes

CBRXLIB is an internal macro used to manipulate the library records in the tape configuration database (TCDB). Table 77 describes the return codes.

Table 77. CBRXLIB Return Codes

Return Code	Error Type	Description
0	Successful	Successful execution.
4	Request not executed	For tape library record creation, a tape library record with the same library name is already present in the TCDB. For tape library record retrieval the tape library record is not present in the TCDB or the TCDB does not exist. For tape library record update, the requested tape library record is not present in the TCDB.
8	Parameter list error	Parameter list invalid for requested function.
12	Environmental error	Internal processing error. The request could not be completed due to the failure of a system service.

Table 77. CBRXLIB Return Codes (continued)

Return Code	Error Type	Description
16	Catalog authorization error	The user of CBRXLIB does not have the SAF/RACF authority to perform the requested function on the user catalog that contains the TCDB. The return code, module ID, and reason code from the catalog service routine are contained in the LCPL, in fields LCPCTRET, LCPCTMOD, and LCPCTRSN, respectively.
20	Catalog access error	The return code, module ID, and reason code from the catalog service routine are contained in the LCPL, in fields LCPCTRET, LCPCTMOD, and LCPCTRSN, respectively.

CBRXVOL Return Codes

CBRXVOL is an internal macro used to manipulate the volume records in the tape configuration database (TCDB). Table 78 describes the return codes.

Table 78. CBRXVOL Return Codes

Return Code	Error Type	Description
0	Successful	Successful execution.
4	Request not executed	For tape volume record creation, a tape volume record with the same volume serial number is already present in the TCDB. For tape volume record update, replacement, or deletion, the requested tape volume record is not present in the TCDB. For tape volume record retrieval, the TCDB does not exist, the tape volume record is not present in the TCDB, or the tape volume record is present in the TCDB but is for an up-level volume. To distinguish between the latter two cases, flag VCP_UPLEVEL_TDSI is set if the volume record contains up-level TDSI information. For sequential tape-volume record retrieval, there are no (more) records to retrieve or the TCDB does not exist.
8	Parameter list error	Parameter list invalid for requested function.
12	Environmental error	Internal processing error. The request could not be completed due to the failure of a system service.
16	Catalog authorization error	The user of CBRXVOL does not have the SAF/RACF authority to perform the requested function on the user catalog that contains the TCDB. The return code, module ID, and reason code from the catalog service routine are contained in the VCPL, in fields VCPCTRET, VCPCTMOD, and VCPCTRSN, respectively.
20	Catalog access error	The return code, module ID, and reason code from the catalog service routine are contained in the VCPL, in fields VCPCTRET, VCPCTMOD, and VCPCTRSN, respectively.
24	Cancel the job	Either the volume not in library installation exit (CBRUXVNL) was invoked and returned with RC=8 (the installation wants to cancel the job immediately), or the volume record contains up-level TDSI information. This return code is only applicable when the EXIT keyword is specified with FUNC(RETRIEVE).

CBRXTCF Return Codes

CBRXTCF is an internal macro used to manipulate the volume and tape records in the tape configuration database (TCDB). Table 79 describes the return codes.

Table 79. CBRXTCF Return Codes

Return Code	Error Type	Description
0	Successful	Successful execution.
4	Request not executed	For a tape library record retrieval, the requested library record is not present in the TCDB, or the TCDB does not exist. For a tape library record update, the requested tape library record is not present in the TCDB. For a tape library record insertion, a tape library record with the same tape library name is already present in the TCDB. For a tape volume record retrieval, end of data has been reached, or the TCDB does not exist. For a tape volume record update, the requested tape volume record is not present in the TCDB.
8	Parameter list error	Parameter list invalid for requested function.
12	Environmental error	The ISMF application has not established the proper environment before making a database access call. For example, TYPE(GETVOL) is requested before TYPE(OPENVOL).
16	Catalog authorization error	The ISMF application does not have the RACF authority to perform the requested function on the user catalog that contains the TCDB. The return code, module ID, and reason code from the catalog service routine are contained in the TCCB in fields TCCCTRET, TCCCTMOD, and TCCCTRSN, respectively.
20	Catalog access error	The return code, module ID, and reason code from the catalog service routine are contained in the TCCB in fields TCCCTRET, TCCCTMOD, and TCCCTRSN, respectively.
24	An error was encountered when issuing the CBRXLCS macro to change the use attribute of a volume.	For a tape volume record update request, the shelf location and volume owner information were updated in the TCDB, however, an error was encountered when CBRXLCS attempted to change the volume's use attribute. The CBRXLCS return code and reason code will be in the TCCB in fields TCCLCSRC and TCCLCSRS. To interpret these values, see the CBRXCLS FUNC(CUA) return and reason code section. If an error occurred when attempting to update the TCDB, the return code, module ID, and reason code from the catalog service routine are contained in the TCCB in fields TCCCTRET, TCCCTMOD, and TCCCTRSN, respectively.

Library Automation Communication Services (LACS) Return and Reason Codes

Return and Reason Codes for the LACS Mount Service

Tables 80, 81, 82, 83, 84, 85, 86, and 87 describe the CBRXLACS return and reason codes.

Table 80. CBRXLACS FUNC(MOUNT) Return and Reason Codes

Return Code	Error Type	Reason Code	Description	AL	ML	Message
X'00'	Successful Execution of Library Function	X'0000'	For an asynchronous request to an ATLDS, the mount order has been successfully scheduled for later execution by the library. For a synchronous request to an ATLDS, the mount completed without error. For a request to an MTL, the WTO/WTOR message has been issued, if requested.	Yes	Yes	N/A
X'04'	Successful Execution of Non-Library Function	X'0000'	The drive does not reside in a library. If requested, the WTO/WTOR message has been issued.	No	No	N/A
X'08'	Execution of Library Function Complete with Warning	X'0101'	The requested ATLDS mount is complete, but the library vision system has failed, preventing verification of the external volume serial number. This code is set only after a synchronous request.	Yes	No	CBR4001I
		X'0102'	The requested ATLDS mount is complete, but the external label on the cartridge is either missing or unreadable. This code is set only after a synchronous request.	Yes	No	CBR4002I
		X'010C'	The ATLDS mounted the scratch volume as requested, but during the attempt to load the volume into the drive, it was determined that the cartridge is damaged. The cartridge has been ejected. The scratch mount can be retried. This code is set only after a synchronous request.	Yes	No	CBR4012I
X'0C'	Invalid Parameter List	X'0121'	A UCBPTR value has not been supplied, or the UCB address is invalid.	N/A	N/A	CBR4033I
		X'0122'	A VOLSER value has not been supplied, or the volume serial number is all blanks.	N/A	N/A	CBR4034I
		X'0123'	A TOKEN address has not been supplied.	N/A	N/A	CBR4035I
		X'0125'	Either a WTOCONS value or a WTOCART value has been supplied, but a WTOPTR value has not.	N/A	N/A	CBR4037I
		X'0127'	More than one synchronization option (ECBPTR, EXITPTR, or WAIT) has been supplied.	N/A	N/A	CBR4039I

Table 80. CBRXLACS FUNC(MOUNT) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
		X'012C'	Either a WTOCONS value or a WTOCART value has been supplied, but the WTO parameter list is in the normal (WPL) format, rather than the extended (WPX) format.	N/A	N/A	CBR4044I
		X'0130'	A DEVTYPE address for tape device selection information has not been supplied.	N/A	N/A	CBR4048I
		X'0131'	The tape device selection information media type is invalid.	N/A	N/A	CBR4049I
		X'0132'	The volume serial number supplied is invalid. The volume serial number must consist of all alphanumeric characters, national characters (@, $, #) or special characters (, . / ' ( ) * & + — =) with no imbedded blanks.	No	Yes	CBR4050I
X'10'	Environmental Error	None	CBRXLACS FUNC(MOUNT) does not use the environmental error return code.			
X'14'	Execution Failed with Permanent Error	X'0161'	The library where the mount is to be performed is offline.	Yes	No	CBR4097I
		X'0162'	The library where the mount is to be performed is not operational.	Yes	No	CBR4098I
		X'0163'	A permanent I/O error was detected by the Asynchronous Operations Manager; a library sense record is not available.	Yes	No	CBR4099I
		X'0164'	An equipment check occurred in the library attachment facility or the library manager.	Yes	No	CBR4100I
		X'0165'	The microcode levels in the control unit and the library manager cannot coexist.	Yes	No	CBR4101I
		X'0166'	One of the following situations has occurred: • The library responded to the mount order with a unit check. The code in the sense record is an unexpected and inappropriate response to the order; for example, a code appropriate to a scratch mount was received in response to a specific mount. • The attention message which signaled the completion of the mount order contained an undefined mount completion code.	Yes	No	CBR4102I

Table 80. CBRXLACS FUNC(MOUNT) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
		X'0167'	The requested volume is currently in use. One of the following conditions is present in the library: • The volume is already mounted on some drive in the library. • A mount of the volume is pending on some drive in the library.	Yes	No	CBR4103I
		X'0168'	The requested volume is not in the library.	Yes	Yes	CBR4104I
		X'0169'	There are no volumes in the library which are assigned to the required scratch category or categories.	Yes	No	CBR4105I
		X'016A'	An invalid sequence of orders has been sent to the library. One of the following conditions is present in the library: • A volume is already mounted on the drive, and no demount request is pending. • A mount request is already queued for the drive.	Yes	No	CBR4106I
		X'016B'	The requested volume is not in the location assigned to it in the library manager inventory.	Yes	No	CBR4107I
		X'0170'	An incompatible function has been requested of the library. The control unit supports library commands, but the library attachment interface is not installed.	Yes	No	CBR4112I
		X'0171'	No libraries are defined in the current configuration.	Yes	No	CBR4113I
		X'0172'	The library configuration has not yet been set.	Yes	No	CBR4114I
		X'0174'	The library manager is offline.	Yes	No	CBR4116I
		X'0175'	The requested volume is inaccessible in the library.	Yes	No	CBR4117I
		X'0176'	The drive on which the volume was to have been mounted is no longer available. It was made unavailable by the library manager after repeated device errors or by the library operator. This code is set only after a synchronous request.	Yes	No	CBR4118I
		X'0177'	The request cannot be completed because the library name associated with the tape drive cannot be determined.	No	Yes	CBR4119I

Table 80. CBRXLACS FUNC(MOUNT) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
		X'0178'	The mount request was sent to the library, but no response has been received from the library. Either the request completed and the completion message was lost, or the request was lost in the library. This code is set only after a synchronous request.	Yes	No	CBR4120I
		X'0179'	The requested cartridge is damaged in such a way that it cannot be loaded into a tape drive. The library manager has ejected the cartridge. This code is set only after a synchronous request.	Yes	No	CBR4122I
		X'017A'	The library mounted the volume as requested, however the attempt to load the volume into the drive failed. This code is set only after a synchronous request.	Yes	No	CBR4011I
		X'017B'	An incompatible function has been requested of the library. The media type of the volume to be mounted cannot coexist with the drive. The media type in the library manager database is inconsistent with the media type of the volume in the tape configuration database.	Yes	No	CBR4123I
		X'017C'	The drive on which volume was to have been mounted was left in stand-alone mode at the library.	Yes	No	CBR4124I
		X'017D'	A valid copy of the requested volume is inaccessible in the library.	Yes	No	CBR4125I
		X'017E'	The drive on which the scratch volume was to have been mounted is in read-only mode.	Yes	No	CBR4126I
X'18'	Execution failed due to system service failure	X'0181'	LACS was unable to establish an ESTAE exit routine. The ESTAE return code is stored in the LAPL in field LAPSRET.	N/A	N/A	CBR4129I
		X'0183'	LACS was unable to write the caller's message to the requested destination. The WTO return code is stored in the LAPL in field LAPSRET. This reason code is associated only with a non-ATLDS mount request; failure to write to the system log does not cause failure of an ATLDS mount request.	No	Yes	CBR4131I
		X'0184'	LACS was unable to send the mount order to the library using the LIBSERV service. The LIBSERV return code and reason code are stored in the LAPL in fields LAPSRET and LAPSREAS, respectively.	Yes	No	CBR4132I

Table 80. CBRXLACS FUNC(MOUNT) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
		X'0191'	The library does not have enough physical drives available.	Yes	No	CBR4127I
		X'0192'	The library is out of empty stacked volumes.	Yes	No	CBR428I
X'1C'	Abnormal Termination in LACS	X'01A1'	An abnormal termination occurred during LACS execution. The system completion code is stored in the LAPL in field LAPSRET. The ABEND reason code, if any, is stored in the LAPL in field LAPSREAS.	N/A	N/A	CBR4161I

Return and Reason Codes for the LACS Demount Service

CBRXLACS FUNC(DEMOUNT) sets the following return and reason codes.

Table 81. CBRXLACS FUNC(DEMOUNT) Return and Reason Codes

Return Code	Error Type	Reason Code	Description	AL	ML	Message
X'00'	Successful Execution of Library Function	X'0000'	For a request to an ATLDS, the demount order has been scheduled for later execution by the library. For a request to an MTL, the WTO/WTOR message has been issued, if requested.	Yes	Yes	N/A
X'04'	Successful Execution of Non-Library Function	X'0000'	The drive does not reside in a library. If requested, the WTO/WTOR message has been issued.	No	No	N/A
X'08'	Execution of Library Function Complete with Warning	X'0203'	The demount request has been scheduled, or the WTO/WTOR message has been issued, but the error status code was not successfully recorded in the tape volume record. An error return code was received from the CBRXVOL service.	Yes	Yes	CBR4003I
		X'0204'	The demount request has been scheduled, or the WTO/WTOR message has been issued, but the volume was not successfully returned to scratch status. An error return code was received from the CBRXLCS service.	Yes	Yes	CBR4004I
X'0C'	Invalid Parameter List	X'0221'	A UCBPTR value has not been supplied, or the UCB address is invalid.	N/A	N/A	CBR4033I
		X'0223'	A TOKEN address has not been supplied.	N/A	N/A	CBR4035I
		X'0225'	Either a WTOCONS value or a WTOCART value has been supplied, but a WTOPTR value has not.	N/A	N/A	CBR4037I
		X'0228'	Both an ERRCODE value and the SCRATCH option have been supplied.	N/A	N/A	CBR4040I

Table 81. CBRXLACS FUNC(DEMOUNT) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
		X'022C'	Either a WTOCONS value or a WTOCART value has been supplied, but the WTO parameter list is in the normal (WPL) format, rather than the extended (WPX) format.	N/A	N/A	CBR4044I
X'10'	Environmental Error	None	CBRXLACS FUNC(DEMOUNT) does not use the environmental error return code.			
X'14'	Execution Failed with Permanent Error	X'0261'	The library where the demount is to be performed is offline.	Yes	No	CBR4097I
		X'0262'	The library where the demount is to be performed is not operational.	Yes	No	CBR4098I
		X'0263'	A permanent I/O error was detected by the Asynchronous Operations Manager; a library sense record is not available.	Yes	No	CBR4099I
		X'0264'	An equipment check occurred in the library attachment facility or the library manager.	Yes	No	CBR4100I
		X'0265'	The microcode levels in the control unit and the library manager cannot coexist.	Yes	No	CBR4101I
		X'0266'	The library responded to the demount order with a unit check. The error code in the sense record is an unexpected and inappropriate response to the order.	Yes	No	CBR4102I
		X'026A'	An invalid sequence of orders has been sent to the library. One of the following conditions is present in the library: • No volume is mounted on the drive, and no mount request is pending. • A demount request is already queued for the drive.	Yes	No	CBR4106I
		X'0270'	An incompatible function has been requested of the library. The control unit supports library commands, but the library attachment interface is not installed.	Yes	No	CBR4112I
		X'0271'	No libraries are defined in the current configuration.	Yes	No	CBR4113I
		X'0272'	The library configuration has not yet been set.	Yes	No	CBR4114I
		X'0274'	The library manager is offline.	Yes	No	CBR4116I
		X'0275'	The volume to be demounted is inaccessible in the library.	Yes	No	CBR4117I
X'18'	Execution Failed Due to System Service Failure	X'0281'	LACS was unable to establish an ESTAE exit routine. The ESTAE return code is stored in the LAPL in field LAPSRET.	N/A	N/A	CBR4129I

Table 81. CBRXLACS FUNC(DEMOUNT) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
		X'0283'	LACS was unable to write the caller's message to the requested destination. The WTO return code is stored in the LAPL in field LAPSRET. This reason code is associated only with a non-ATLDS demount request; failure to write to the system log does not cause failure of an ATLDS demount request.	No	Yes	CBR4131I
		X'0284'	LACS was unable to send the demount order to the library using the LIBSERV service. The LIBSERV return code and reason code are stored in the LAPL in fields LAPSRET and LAPSREAS, respectively.	Yes	No	CBR4132I
X'1C'	Abnormal Termination in LACS	X'02A1'	An abnormal termination occurred during LACS execution. The system completion code is stored in the LAPL in field LAPSRET. The ABEND reason code, if any, is stored in the LAPL in field LAPSREAS.	N/A	N/A	CBR4161I

Return and Reason Codes for the LACS Wait Service

CBRXLACS FUNC(WAIT) sets the following return and reason codes.

Table 82. CBRXLACS FUNC(WAIT) Return and Reason Codes

Return Code	Error Type	Reason Code	Description	AL	ML	Message
X'00'	Successful Execution of Library Function	X'0000'	For a request to an ATLDS, the requested mount completed without error. For a request to an MTLDS, no action has been taken.	Yes	Yes	N/A
X'04'	Successful Execution of Non-Library Function	X'0000'	The drive does not reside in a library. No action has been taken.	No	No	N/A
X'08'	Execution of Library Function Complete with Warning	X'0301'	The requested mount is complete, but the library vision system has failed, preventing verification of the external volume serial number.	Yes	No	CBR4001I
		X'0302'	The requested mount is complete, but the external label on the cartridge is either missing or unreadable.	Yes	No	CBR4002I
		X'030C'	The library mounted the scratch volume as requested, but during the attempt to load the volume into the drive, it was determined that the cartridge is damaged. The cartridge has been ejected. The scratch mount can be retried.	Yes	No	CBR4012I

Table 82. CBRXLACS FUNC(WAIT) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
X'0C'	Invalid Parameter List	X'0321'	A UCBPTR value has not been supplied, or the UCB address is invalid.	N/A	N/A	CBR4033I
		X'0323'	A TOKEN address has not been supplied.	N/A	N/A	CBR4035I
		X'032D'	The requested drive resides in a library, and the LACS token value is zero.	Yes	No	CBR4045I
		X'032E'	The mount request specified either ECBPTR or EXITPTR as the synchronization option. A FUNC(WAIT) request requires that mount completion be posted in the UCB library automation extension	Yes	No	CBR4046I
X'10'	Environmental Error	X'0342'	The token mount request was not found. The request is not pending, and it is not the most recently completed library order.	Yes	No	CBR4066I
X'14'	Execution Failed with Permanent Error	X'0364'	An equipment check occurred in the library attachment facility or the library manager.	Yes	No	CBR4100I
		X'0366'	The attention message which signaled the completion of the mount order contained an undefined mount completion code.	Yes	No	CBR4102I
		X'0369'	There are no volumes in the library which are assigned to the required scratch category.	Yes	No	CBR4105I
		X'036B'	The requested volume is not in the location assigned to it in the library manager inventory.	Yes	No	CBR4107I
		X'0375'	The requested volume is inaccessible in the library.	Yes	No	CBR4117I
		X'0376'	The drive on which the volume was to have been mounted is no longer available. It was made unavailable by the library manager after repeated device errors or by the library operator.	Yes	No	CBR4118I
		X'0378'	The mount request was sent to the library, but no response has been received from the library. Either the request completed and the completion message was lost, or the request was lost in the library.	Yes	No	CBR4120I
		X'0379'	The requested cartridge is damaged in such a way that it cannot be loaded into a tape drive. The library manager has ejected the cartridge.	Yes	No	CBR4122I
		X'037A'	The library mounted the volume as requested, however the attempt to load the volume into the drive failed.	Yes	No	CBR4011I

Table 82. CBRXLACS FUNC(WAIT) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
X'18'	Execution Failed Due to System Service Failure	X'0381'	LACS was unable to establish an ESTAE exit routine. The ESTAE return code is stored in the LAPL in field LAPSRET.	N/A	N/A	CBR4129I
		X'0385'	LACS was unable to determine whether the mount request was still pending using the AOMQUE service. The AOMQUE return code and reason code are stored in the LAPL in fields LAPSRET and LAPSREAS, respectively.	Yes	No	CBR4133I
X'1C'	Abnormal Termination in LACS	X'03A1'	An abnormal termination occurred during LACS execution. The system completion code is stored in the LAPL in field LAPSRET. The ABEND reason code, if any, is stored in the LAPL in field LAPSREAS.	N/A	N/A	CBR4161I

Return and Reason Codes for the LACS Verify Service

CBRXLACS FUNC(VERIFY) sets the following return and reason codes.

Table 83. CBRXLACS FUNC(VERIFY) Return and Reason Codes

Return Code	Error Type	Reason Code	Description	AL	ML	Message
X'00'	Successful Execution of Library Function	X'0000'	For a scratch mount request on an ATLDS drive, the internal and external volume serial numbers match. For a specific volume mount request on an ATLDS drive, the requested, internal, and external volsers all match. For a scratch mount request on an MTLDS drive, an acceptable scratch volume has been mounted. For a specific volume mount request on an MTLDS drive, the requested and internal volsers match.	Yes	Yes	N/A
X'04'	Successful Execution of Non-Library Function	X'0000'	The drive does not reside in a library. No action has been taken.	No	No	N/A
X'08'	Execution of Library Function Complete with Warning	X'0405'	The internal and external volume serial numbers do not match for a scratch volume mount on an ATLDS drive.	Yes	No	CBR4005I
		X'0406'	The internal and external volume serial numbers do not match for a specific volume mount performed while the ATLDS was in manual mode.	Yes	No	CBR4006I

Table 83. CBRXLACS FUNC(VERIFY) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
		X'0407'	The volume mounted by the operator in response to a mount scratch request for an MTLDS drive is not defined in the tape configuration database	No	Yes	CBR4007I
		X'0408'	The volume mounted by the operator in response to a mount scratch request for an MTLDS drive is defined in the TCDB but does not reside in the MTLDS.	No	Yes	CBR4008I
		X'0409'	The volume mounted by the operator in response to a mount scratch request for an MTLDS drive is defined in the TCDB and resides in the MTLDS but is not a scratch tape.	No	Yes	CBR4009I
		X'040A'	The internal and requested volume serial numbers do not match for a specific volume mount on an MTLDS drive.	No	Yes	CBR4010I
X'0C'	Invalid Parameter List	X'0421'	A UCBPTR value has not been supplied, or the UCB address is invalid.	N/A	N/A	CBR4033I
		X'0422'	A VOLSER value has not been supplied.	N/A	N/A	CBR4034I
		X'0423'	A TOKEN address has not been supplied.	N/A	N/A	CBR4035I
		X'042D'	The requested drive resides in a library, and the LACS token value is zero.	N/A	N/A	CBR4045I
		X'0432'	The volume serial number supplied is invalid. The volume serial number must consist of all alphanumeric characters, national characters (@, $, #) or special characters (, . / ' ( ) * & + — =) with no imbedded blanks.	No	Yes	CBR4050I
X'10'	Environmental Error	X'0442'	The token mount request was not found. The request is not pending in the library, and it is not the most recently completed library order.	Yes	No	CBR4066I
		X'0443'	The token mount request is still pending in the library.	Yes	No	CBR4067I
X'14'	Execution Failed with Permanent Error	X'046C'	The external volume serial number of the cartridge mounted on the ATLDS drive cannot be determined. The information is not in the UCB and is not available from LIBSERV.	Yes	No	CBR4108I
		X'046D'	The external volume serial number of the volume mounted by the ATLDS does not match the VOLSER requested in the mount order, and the ATLDS has not provided any error indication.	Yes	No	CBR4109I
		X'046E'	The internal and external volume serial numbers do not match for a specific volume mount performed while the ATLDS was in automatic mode.	Yes	No	CBR4110I

Table 83. CBRXLACS FUNC(VERIFY) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
		X'046F'	The internal and external volume serial numbers do not match for AVR volume mount verification.	Yes	No	CBR4111I
		X'0477'	The request cannot be completed because the library name associated with the tape drive cannot be determined.	No	Yes	CBR4119I
X'18'	Execution Failed Due to System Service Failure	X'0481'	LACS was unable to establish an ESTAE exit routine. The ESTAE return code is stored in the LAPL in field LAPSRET.	N/A	N/A	CBR4129I
		X'0484'	LACS was unable to determine the external volume serial number of the volume currently mounted on the drive, using the LIBSERV service. The LIBSERV return code and reason code are stored in the LAPL in fields LAPSRET and LAPSREAS, respectively.	Yes	No	CBR4132I
		X'0485'	LACS was unable to determine whether the mount request was still pending using the AOMQUE service. The AOMQUE return code and reason code are stored in the LAPL in fields LAPSRET and LAPSREAS, respectively.	Yes	No	CBR4133I
		X'0487'	LACS was unable to read the tape volume record for the mounted volume from the tape configuration database using the CBRXVOL service. The CBRXVOL return code is stored in the LAPL in field LAPSRET.	No	Yes	CBR4135I
X'1C'	Abnormal Termination in LACS	X'04A1'	An abnormal termination occurred during LACS execution. The system completion code is stored in the LAPL in field LAPSRET. The ABEND reason code, if any, is stored in the LAPL in field LAPSREAS.	N/A	N/A	CBR4161I

Return and Reason Codes for the LACS Cancel Service

CBRXLACS FUNC(CANCEL) sets the following return and reason codes.

Table 84. CBRXLACS FUNC(CANCEL) Return and Reason Codes

Return Code	Error Type	Reason Code	Description	AL	ML	Message
X'00'	Successful Execution of Library Function	X'0000'	For a request to an ATLDS, the mount request represented by the token was successfully canceled. For a request to an MTL, the operator message represented by the token, if any, has been DOMed.	Yes	Yes	N/A

Table 84. CBRXLACS FUNC(CANCEL) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
X'04'	Successful Execution of Non-Library Function	X'0000'	The drive does not reside in a library. The operator message represented by the token, if any, has been DOMed.	No	No	N/A
X'08'	Execution of Library Function Complete with Warning	None	CBRXLACS FUNC(CANCEL) does not use the warning return code.			
X'0C'	Invalid Parameter List	X'0521'	A UCB address has not been supplied, or the UCB address is invalid. The error may be in the LAPL, or in an entry in the UCB/token list.	N/A	N/A	CBR4033I
		X'0523'	A token address has not been supplied. The error may be in the LAPL, or in an entry in the UCB/token list.	N/A	N/A	CBR4035I
		X'0526'	Neither a UCBPTR value nor a UCBTLIST value has been supplied in the LAPL.	N/A	N/A	CBR4038I
		X'0529'	Both a UCBPTR value and a UCBTLIST value have been supplied in the LAPL.	N/A	N/A	CBR4041I
		X'052D'	The requested drive resides in a library, and the LACS token value is zero. The error may be in the LAPL, or in an entry in the UCB/token list.	Yes	No	CBR4045I
X'10'	Environmental Error	X'0542'	The token mount request was not found. The request is not pending in the library, and it is not the most recently completed mount order.	Yes	No	CBR4066I
X'14'	Execution Failed with Permanent Error	X'0561'	The library where the mount is to be canceled is offline.	Yes	Yes	CBR4097I
		X'0562'	The library where the mount is to be canceled is not operational.	Yes	No	CBR4098I
		X'0563'	A permanent I/O error was detected by the Asynchronous Operations Manager; a library sense record is not available.	Yes	No	CBR4099I
		X'0564'	An equipment check occurred in the library attachment facility or the library manager.	Yes	No	CBR4100I
		X'0565'	The microcode levels in the control unit and the library manager cannot coexist.	Yes	No	CBR4101I
		X'0566'	The library responded to the demount order with a unit check. The error code in the sense record is an unexpected and inappropriate response to the order.	Yes	No	CBR4102I

Table 84. CBRXLACS FUNC(CANCEL) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
		X'056A'	An invalid sequence of orders has been sent to the library. One of the following conditions is present in the library: • No volume is mounted on the drive, and no mount request is pending. • A demount request is already queued for the drive.	Yes	No	CBR4106I
		X'0574'	The library manager is offline.	Yes	No	CBR4116I
X'18'	Execution Failed Due to System Service Failure	X'0581'	LACS was unable to establish an ESTAE exit routine. The ESTAE return code is stored in the LAPL in field LAPSRET.	N/A	N/A	CBR4129I
		X'0584'	LACS was unable to send the demount order to the library using the LIBSERV service. The LIBSERV return code and reason code are stored in the LAPL in fields LAPSRET and LAPSREAS, respectively.	Yes	No	CBR4132I
		X'0585'	LACS was unable to manipulate the AOM queue element for the mount request using the AOMQUE service. The AOMQUE return code and reason code are stored in the LAPL in fields LAPSRET and LAPSREAS, respectively.	Yes	No	CBR4133I
X'1C'	Abnormal Termination in LACS	X'05A1'	An abnormal termination occurred during LACS execution. The system completion code is stored in the LAPL in field LAPSRET. The ABEND reason code, if any, is stored in the LAPL in field LAPSREAS.	N/A	N/A	CBR4161I

Return and Reason Codes for the LACS WTO Service

CBRXLACS FUNC(WTO) sets the following return and reason codes.

Table 85. CBRXLACS FUNC(WTO) Return and Reason Codes

Return Code	Error Type	Reason Code	Description	AL	ML	Message
X'00'	Successful Execution of Library Function	X'0000'	For a request to an ATLDS, the WTO/WTOR message has been written to the system log, or the WTO/WTOR message has been successfully issued. For a request to an MTLDS, the WTO/WTOR message has been successfully issued.	Yes	Yes	N/A
X'04'	Successful Execution of Non-Library Function	X'0000'	The WTO/WTOR message has been successfully issued.	No	No	N/A

Table 85. CBRXLACS FUNC(WTO) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
X'08'	Execution of Library Function Complete with Warning	None	CBRXLACS FUNC(WTO) does not use the warning return code.			
X'0C'	Invalid Parameter List	X'0621'	A UCBPTR value has not been supplied, or the UCB address is invalid.	N/A	N/A	CBR4033I
		X'0623'	A TOKEN address has not been supplied.	N/A	N/A	CBR4035I
		X'0625'	A WTOPTR value has not been supplied.	N/A	N/A	CBR4037I
		X'062C'	Either a WTOCONS value or a WTOCART value has been supplied, but the WTO parameter list is in the normal (WPL) format, rather than the extended (WPX) format.	N/A	N/A	CBR4044I
X'10'	Environmental Error	None	CBRXLACS FUNC(WTO) does not use the environmental error return code.			
X'14'	Execution Failed with Permanent Error	None	CBRXLACS FUNC(WTO) does not use the permanent error return code.			
X'18'	Execution Failed Due to System Service Failure	X'0681'	LACS was unable to establish an ESTAE exit routine. The ESTAE return code is stored in the LAPL in field LAPSRET.	N/A	N/A	CBR4129I
		X'0683'	LACS was unable to write the caller's message to the requested destination. The WTO return code is stored in the LAPL in field LAPSRET.	Yes	Yes	CBR4131I
X'1C'	Abnormal Termination in LACS	X'06A1'	An abnormal termination occurred during LACS execution. The system completion code is stored in the LAPL in field LAPSRET. The ABEND reason code, if any, is stored in the LAPL in field LAPSREAS.	N/A	N/A	CBR4161I

Return and Reason Codes for the LACS DOM Service

CBRXLACS FUNC(DOM) sets the following return and reason codes.

Table 86. CBRXLACS FUNC(DOM) Return and Reason Codes

Return Code	Error Type	Reason Code	Description	AL	ML	Message
X'00'	Successful Execution of Library Function	X'0000'	The operator message has been successfully DOMed.	Yes	Yes	N/A
X'04'	Successful Execution of Non-Library Function	X'0000'	The operator message has been successfully DOMed.	No	No	N/A

Table 86. CBRXLACS FUNC(DOM) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
X'08'	Execution of Library Function Complete with Warning	None	CBRXLACS FUNC(DOM) does not use the warning return code.			
X'0C'	Invalid Parameter List	X'0721'	A UCBPTR value has not been supplied, or the UCB address is invalid.	N/A	N/A	CBR4033I
		X'0723'	A TOKEN address has not been supplied.	N/A	N/A	CBR4035I
X'10'	Environ- mental Error	None	CBRXLACS FUNC(DOM) does not use the environmental error return code.			
X'14'	Execution Failed with Permanent Error	None	CBRXLACS FUNC(DOM) does not use the permanent error return code.			
X'18'	Execution Failed Due to System Service Failure	X'0781'	LACS was unable to establish an ESTAE exit routine. The ESTAE return code is stored in the LAPL in field LAPSRET.	N/A	N/A	CBR4129I
X'1C'	Abnormal Termination in LACS	X'07A1'	An abnormal termination occurred during LACS execution. The system completion code is stored in the LAPL in field LAPSRET. The ABEND reason code, if any, is stored in the LAPL in field LAPSREAS.	N/A	N/A	CBR4161I

Return and Reason Codes for the LACS Error Message Construction Service

CBRXLACS FUNC(ERRTEXT) sets the following return and reason codes.

Table 87. CBRXLACS FUNC(ERRTEXT) Return and Reason Codes

Return Code	Error Type	Reason Code	Description	AL	ML	Message
X'00'	Successful Execution of Library Function	X'0000'	The drive resides in a library; the requested messages have been constructed and placed into the message buffer.	Yes	Yes	N/A
X'04'	Successful Execution of Non-Library Function	X'0000'	The drive does not reside in a library; the requested messages have been constructed and placed into the message buffer.	No	No	N/A
X'08'	Execution of Library Function Complete with Warning	None	CBRXLACS FUNC(ERRTEXT) does not use the warning return code.			
X'0C'	Invalid Parameter List	X'0824'	A BTOKPTR value has not been supplied.	N/A	N/A	CBR4036I
		X'082A'	Either the return code or the reason code is invalid.	N/A	N/A	CBR4042I

Table 87. CBRXLACS FUNC(ERRTEXT) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
		X'082F'	The input return code and reason code do not represent an error or warning condition. There is no LCS error message for this situation.	N/A	N/A	CBR4047I
X'10'	Environmental Error	None	CBRXLACS FUNC(ERRTEXT) does not use the environmental error return code.			
X'14'	Execution Failed with Permanent Error	None	CBRXLACS FUNC(ERRTEXT) does not use the permanent error return code.			
X'18'	Execution Failed Due to System Service Failure	X'0881'	LACS was unable to establish an ESTAE exit routine. The ESTAE return code is stored in the LAPL in field LAPSRET.	N/A	N/A	CBR4129I
		X'0882'	LACS was unable to construct an error message using the OAM message construction service. The return code from the service is stored in the LAPL in field LAPSRET.	N/A	N/A	CBR4130I
		X'0886'	LACS was unable to place an error message into the message buffer using the IEEMIFSV service. The IEEMIFSV return code and reason code are stored in the LAPL in fields LAPSRET and LAPSREAS, respectively.	N/A	N/A	CBR4134I
X'1C'	Abnormal Termination in LACS	X'08A1'	An abnormal termination occurred during LACS execution. The system completion code is stored in the LAPL in field LAPSRET. The ABEND reason code, if any, is stored in the LAPL in field LAPSREAS.	N/A	N/A	CBR4161I

OAM Records Written to SYS1.LOGREC

This section covers OAM software error records and hardware error records that the SYS1.LOGREC log file might contain.

Software Error Records

When OAM issues message CBR7100I to report a failure (abend), it writes a software error record in SYS1.LOGREC.

Also, when OAM detects a programming failure, it constructs in SYS1.LOGREC a symptom record that contains a description of the failure.

To print these records, use the EREP service aid. For information on EREP, see *EREP User's Guide* and *EREP Reference*.

Hardware Error Records

When an error occurs during an attempt to access either a library or drive, data is gathered to determine recovery from the error. When the error recovery procedures are invoked, an outboard recording record (OBR) is built and written to SYS1.LOGREC. To print these records, use the EREP service aid.

Tables 88, 89, 90, 91, and 92 provide mapping of the OBR.

Table 88. 9246 Optical Disk Library Long OBR

Offset Dec(Hex)	Name	Length	Description
0(X'0')	OBRKEY	1	OBR record type. Set to X'30'.
1(X'1')	OBRKEY2	1	System release level. Set to X'83'.
2(X'2')	OBRSMS	1	Record independent switches.
3(X'3')	OBRSW2	1	Record dependent switches (bit 1 is a temporary error indicator).
4(X'4')	OBRXASW	1	370/XA mode record switch (bit 0 on indicates CHPID is valid).
5(X'5')	OBRSCID	1	Channel set ID, set to zeros.
6(X'6')	OBRRCDCT	1	Record count.
7(X'7')		1	Reserved.
8(X'8')	OBRDT	8	System date and time.
8(X'8')	OBRTIME	4	System date of failure.
12(X'C')	OBRDATE	4	System time of failure.
16(X'10')	OBRCPUID	8	CPU identification.
24(X'18')	OBRJOBID	8	Job name or user ID.
32(X'20')	OBRFCCW	8	Failing CCW, set to zeros.
40(X'28')	OBRCSW	8	Channel status word, set to zeros.
48(X'30')	OBRDEVDC	1	Number of doublewords of device-dependent data, set to X'04'.
49(X'31')	OBRSECUA	3	Secondary control unit address, set to zeros.
52(X'34')	OBRCODE	4	Set to X'08002180' for the IBM 9246 Optical Disk Library.
56(X'38')	OBRLSDRC	1	Statistical data amount, set to 0.

Table 88. 9246 Optical Disk Library Long OBR (continued)

Offset Dec(Hex)	Name	Length	Description
57(X'39')	OBRPCUA	3	Device number of CTC adapter.
60(X'3C')	OBRRETRY	2	Number of retries attempted.
62(X'3E')	OBRSBCNT	2	Number of sense bytes, set to X'0010'.
64(X'40')	OBRDEVDP	32	Device-dependent data.
64(X'40')	OBRLNAME	8	Library name.
72(X'48')	OBRLSER	7	Library serial number.
79(X'4F')	*	1	Reserved.
80(X'50')	OBRFCMD	5	Failing library command.
85(X'55')	*	3	Reserved.
88(X'58')	OBRPROT	2	Protocol status.
90(X'5A')	OBRFCRC	2	Library adapter return code.
92(X'5C')	*	4	Reserved.
96(X'60')	OBRSENSE	16	Device-dependent sense.
96(X'60')	OBRLBST	4	Overall machine status from IBM 9246 Optical Disk Library.
100(X'64')	OBRLBFST	3	Fault code from Filenet library.
103(X'67')	*	1	Reserved.
104(X'68')	OBRBST	3	Backup mode code.
107(X'6B')	*	5	Reserved.

Mapping of the 9247 Optical Disk Drive OBR follows.

Table 89. 9247 Optical Disk Drive Long OBR

Offset Dec(Hex)	Name	Length	Description
0(X'0')	OBRKEY	1	OBR record type, set to X'30'.
1(X'1')	OBRKEY2	1	System release level, set to X'83'.
2(X'2')	OBRSMS	1	Record independent switches.
3(X'3')	OBRSW2	1	Record dependent switches (bit 0 is a temporary error indicator).
4(X'4')	OBRXASW	1	370/XA mode record switch (bit 1 on indicates CHPID is valid).
5(X'5')	OBRSCID	1	Channel set ID, set to zeros.
6(X'6')	OBRRCDCT	1	Record count.
7(X'7')		1	Reserved.
8(X'8')	OBRDT	8	System date and time.
8(X'8')	OBRDATE	4	System date of failure.
12(X'C')	OBRTIME	4	System time of failure.
16(X'10')	OBRCPUID	8	CPU identification.
24(X'18')	OBRJOBID	8	Job name or user ID.
32(X'20')	OBRFCCW	8	Failing CCW, set to zeros.
40(X'28')	OBRCSW	8	Channel status word, set to zeros.

Table 89. 9247 Optical Disk Drive Long OBR (continued)

Offset Dec(Hex)	Name	Length	Description
48(X'30')	OBRDEVDC	1	Number of doublewords of device-dependent data, set to X'05'
49(X'31')	OBRSECUA	3	Secondary control unit address, set to zeros.
52(X'34')	OBRCODE	4	Set to DASD device type X'08002181' for the IBM 9247 Optical Disk Drive.
56(X'38')	OBRLSDRC	1	Statistical data amount set to 0.
57(X'39')	OBRPCUA	3	Device number of CTC adapter.
60(X'3C')	OBRRETRY	2	Number of retries attempted.
62(X'3E')	OBRSBCNT	2	Number of sense bytes, set to X'0010'.
64(X'40')	OBRDEVDP	40	Device-dependent data.
64(X'40')	OBRNAME	8	Drive name.
72(X'48')	OBRSER	6	Volume serial number of mounted volume.
78(X'4E')	OBRFSCSI	10	Failing SCSI command.
88(X'58')	OBRDD	8	Diagnostic data.
96(X'60')	*	2	Reserved.
98(X'62')	OBRPRTD	2	Protocol status.
100(X'64')	OBRFCRC	1	SCSI adapter return code.
101(X'65')	OBRCCC	1	SCSI adapter completion code.
102(X'66')	OBRDCSB	1	SCSI completion status byte.
103(X'67')	OBRDBUS	1	SCSI bus address.
104(X'68')	OBRSENSE	16	Device-dependent sense data.
104(X'68')	OBRDEC	4	Error class and error code.
104(X'68')	OBRDVA	1...	Address in information bytes is valid.
104(X'68')	OBRDECL	.xxx	Error class.
104(X'68')	OBRDECO	 xxxx	Error code.
105(X'69')	OBRDSN	1	Segment number.
106(X'6A')	*	xxxx	Reserved.
106(X'6A')	OBRDSK	 xxxx	Sense key.
107(X'6B')	OBRDIB	4	Information bytes.
111(X'6F')	OBRDASL	1	Additional sense length.
112(X'70')	OBRDSRC	2	Skip/residual count.
114(X'72')	OBRDSB	1	Status byte.
115(X'73')	OBRDFC	1	Fault code.
116(X'74')	OBRDDS1	1	Device status 1.
117(X'75')	OBRDDS2	1	Device status 2.
118(X'76')	OBRDES1	1	Error status 1.
119(X'77')	OBRDES2	1	Error status 2.

The mapping of the 3995 library long OBR follows.

Table 90. 3995 Library Long OBR Format

Offset Dec(Hex)	Name	Length	Description
0(X'0')	OBRKEY	1	OBR record type, set to X'30'.
1(X'1')	OBRKEY2	1	System release level, set to X'83'.
2(X'2')	OBRSMS	1	Record independent switches.
3(X'3')	OBRSW2	1	Record dependent switches.
4(X'4')	OBRXASW	1	370/XA mode record switch.
4(X'4')		1...	CHPID is valid.
5(X'5')	OBRSCID	1	Channel set ID, set to zeros.
6(X'6')	OBRRCDCT	1	Record count.
7(X'7')		1	Reserved.
8(X'8')	OBRDT	8	System date and time.
8(X'8')	OBRDATE	4	System date of failure.
12(X'C')	OBRTIME	4	System time of failure.
16(X'10')	OBRCPUID	8	CPU identification.
24(X'18')	OBRJOBID	8	Job name or user ID.
32(X'20')	OBRFCCW	8	Failing CCW, set to zeros.
40(X'28')	OBRCSW	8	Channel status word, set to zeros.
48(X'30')	OBRDEVDC	1	Number of doublewords of device-dependent data, set to X'23'.
49(X'31')	OBRSECUA	3	Secondary control unit address, set to zeros.
52(X'34')	OBRCODE	4	Set device type X'08002182'.
56(X'38')	OBRLSDRC	1	Statistical data amount set to 0.
57(X'39')	OBRPCUA	3	Device number of CTC adapter.
60(X'3C')	OBRRETRY	2	Number of retries attempted, set to 0.
62(X'3E')	OBRSBCNT	2	Number of sense bytes, set to X'0020'.
64(X'40')	OBRDEVDP	280	Device-dependent data.
64(X'40')	OBRNAME	8	Library name.
72(X'48')	OBRSER	8	Library serial number.
80(X'50')	OBRFORDR	260	Failing command.
340(X'154')		4	Reserved.
344(X'158')	OBRDSN	32	Device-dependent sense data.
344(X'158')	OBRTRBRC	2	Task request block return code.
346(X'15A')	OBRFSC	2	Fault symptom code.
348(X'15C')	OBRDEVSN	28	Device sense.

The mapping of the 3995 drive long OBR follows.

Table 91. 3995 Optical Disk Drive Long OBR

Offset Dec(Hex)	Name	Length	Description
0(X'0')	OBRKEY	1	OBR record type, set to X'30'.
1(X'1')	OBRKEY2	1	System release level, set to X'83'.
2(X'2')	OBRSMS	1	Record independent switches.
3(X'3')	OBRSW2	1	Record dependent switches.
4(X'4')	OBRXASW	1	370/XA mode record switch.
4(X'4')		1...	CHPID is valid.
5(X'5')	OBRSCID	1	Channel set ID, set to zeros.
6(X'6')	OBRRCDCT	1	Record count.
7(X'7')		1	Reserved.
8(X'8')	OBRDT	8	System date and time.
8(X'8')	OBRDATE	4	System date of failure.
12(X'C')	OBRTIME	4	System time of failure.
16(X'10')	OBRCPUID	8	CPU identification.
24(X'18')	OBRJOBID	8	Job name or user ID.
32(X'20')	OBRFCCW	8	Failing CCW, set to zeros.
40(X'28')	OBRCSW	8	Channel status word, set to zeros.
48(X'30')	OBRDEVDC	1	Number of doublewords of device-dependent data, set to X'23'.
49(X'31')	OBRSECUA	3	Secondary control unit address, set to zeros.
52(X'34')	OBRCODE	4	Set device type X'08002183'.
56(X'38')	OBRLSDRC	1	Statistical data amount set to 0.
57(X'39')	OBRPCUA	3	Device number of CTC adapter.
60(X'3C')	OBRRETRY	2	Number of retries attempted, set to 0.
62(X'3E')	OBRSBCNT	2	Number of sense bytes, set to X'0020'.
64(X'40')	OBRDEVDP	280	Device-dependent data.
64(X'40')	OBRNAME	8	Drive name.
72(X'48')	OBRSER	6	Volume serial number of mounted volume.
78(X'4E')		2	Reserved.
80(X'50')	OBRFORDR	260	Failing command.
340(X'154')		4	Reserved.
344(X'158')	OBRDSN	32	Device-dependent sense data.
344(X'158')	OBRTRBRC	2	Task request block return code.
346(X'15A')	OBRFSC	2	Fault symptom code.
348(X'15C')	OBRDEVSN	28	Device sense.

Refer to the *IBM 3995 Optical Library: Maintenance Information Manual* for an example of a formatted 3995 OBR record.

The mapping for the 3995 library miscellaneous data record (MDR) follows.

Table 92. 3995 Library MDR

Offset Dec(Hex)	Name	Length	Description
0(X'0')	MDRCLSRC	1	MDR type, set to X'91'.
1(X'1')	MDRSYREL	1	System release level, set to X'83'.
2(X'2')	MDRSWTCH	1	Record switches.
3(X'3')		1	Reserved.
4(X'4')	MDRDEVTP	1	Device type, set to X'50'.
5(X'5')		1	Reserved.
6(X'6')	MDRRCDCT	1	Record count.
7(X'7')	MDRCHPID	1	Channel path identifier.
8(X'8')	MDRDT	8	System date and time.
8(X'8')	MDRDATE	4	System date of receipt.
12(X'C')	MDRTIME	4	System time of receipt.
16(X'10')	MDRMVERN	1	Machine version number.
17(X'11')	MDRCPUSR	3	CPU serial number.
20(X'14')	MDRCPMOD	2	CPU machine model number.
22(X'16')		2	Reserved.
24(X'18')	MDRDEVAD	2	Device address of data identified in record.
26(X'1A')	MDRDEVDP	282	Device-dependent information.
26(X'1A')	MDRVOL	6	Volume serial number. The volume serial number which is demounted from an optical drive.
32(X'20')	MDRDRVNM	8	Drive name.
40(X'28')	MDRLIBNM	8	Library name.
48(X'30')	MDRLIBSN	8	Library serial number.
56(X'38')	MDRBOXID	2	The box in the configuration that the buffer log data belongs to.
58(X'3A')	MDROWNER	64	Owner information. A 64-byte data area for owner information or comments.
122(X'7A')	MDRSTAMP	4	Time stamp at demount. The time when the cartridge is demounted from the drive. The time stamp is a 4-byte sequential number of the seconds since January 1, 1980.
126(X'7E')	MDRTWRT	4	Number of temporary write errors. The number is increased by 1 whenever the SCSI sense key reports a recovered error and the additional sense code is write error.
130(X'82')	MDRPWRT	4	Number of permanent write errors. The number is increased by 1 whenever the SCSI sense key reports a medium error and the additional sense code is write error.

Table 92. 3995 Library MDR (continued)

Offset Dec(Hex)	Name	Length	Description
134(X'86')	MDRKBWRT	4	Estimated number of kilobytes written. This is an estimated number of data written on the cartridge. The unit is kilobytes which is equal to 1024 bytes.
138(X'8A')	MDRTRD	4	Number of temporary read errors. The number is increased by 1 whenever the SCSI sense key reports a recovered error and the additional sense code is read error.
142(X'8E')	MDRPRD	4	Number of permanent read errors. The number is increased by 1 whenever the SCSI sense key reports a medium error and the additional sense code is read error.
146(X'92')	MDRKBRD	4	Estimated number of kilobytes read. This is an estimated number of data read from the cartridge. The unit is kilobytes which is equal to 1024 bytes.
150(X'96')	MDRTSK	4	Number of temporary seek errors. The number is increased by 1 whenever the SCSI sense key reports a recovered error and the additional sense code is seek positioning error.
154(X'9A')	MDRPSK	4	Number of permanent seek errors. The number is increased by 1 whenever the SCSI sense key reports a medium error and the additional sense code is no-seek-complete.
158(X'9E')	MDRTLD	4	Number of temporary load errors. The number is increased by 1 whenever the autochanger reports a recovered error and the hardware error code is drive access error.
162(X'A2')	MDRPLD	4	Number of permanent load errors. The number is increased by 1 whenever the autochanger reports a hardware error and the hardware error code is drive access error.
166(X'A6')	MDRDRVNO	1	Drive number. This is the number of the drive on which the volume is mounted.
167(X'A7')	MDRKBERS	4	Estimated number of kilobytes erased. This is an estimated number of data erased from the cartridge. The unit is kilobytes which is equal to 1024 bytes.
171(X'AB')	MDRUSED	4	Percent used of spare sectors. This is the percentage of spare sectors that have been used.
175(X'AF')	MDRFREE	4	Total number of spare sectors. This is the total number of spare sectors initially available when the cartridge was first formatted.

Table 92. 3995 Library MDR (continued)

Offset Dec(Hex)	Name	Length	Description
179(X'B3')	MDRMFLGS	4	Media type flags.
183(X'B7')	MDRMFLGS	125	Reserved for future use.

See *EREP User's Guide* and *EREP Reference* for an example of a formatted 3995 MDR.

OAM Dump

Although an abnormal termination not covered by a recovery environment is unlikely to occur in the OAM address space, a SYSABEND DD statement is included in the default cataloged procedure (member OAM in SYS1.PROCLIB) shipped with the product. This DD statement directs the system to produce a dump if an abnormal termination is encountered. Because SYSOUT=A is specified, any dump produced is written to a system output data set. The system output data set is printed when the OAM address space stops.

Depending on the type of dump required, you can change the SYSABEND DD statement in the default cataloged procedure to a SYSMDUMP or SYSUDUMP DD statement.

Most OAM code is covered by some type of recovery environment (ESTAE, ESTAI or FRR). If an abnormal termination occurs, the recovery routine will cause an SVC dump to be written to a system dump data set (SYS.DUMPxx). The title that is associated with the SVC dump indicates the:
1. SVC dump that is associated with OAM
2. Component ID that is associated with OAM, 5695DF180
3. Name of the OAM module that is generating the SVC dump.

For diagnostic purposes, there are occasions when an SVC dump of the OAM address space and its associated data is required. To make this task easier, OAM provides a DUMP command that will take an SVC dump of the OAM address space, with all of the SDATA parameters that OAM normally requests for problem diagnosis. This command also provides the ability to dump, along with OAM, any address spaces which have work queued to the OAM address space, specific address spaces or specific MVS jobs, specified by the operator on the command invocation.

For more information on the OAM DUMP command, including the command syntax, see *z/OS DFSMS OAM Application Programmer's Reference*. Also, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*.

OAM IPCS Processing

The exit routine is invoked in the standard manner, by using the IPCS VERBEXIT subcommand, with the following syntax.

Operation	Parameters
VERBX OAMDATA 'parameters'	**{ACTIVE|MAIN|STORAGE}** **{DSNAME(***dsname***)|DATASET(***dsname***)}** **{FILE(***ddname***)|DDNAME(***ddname***)}** **(AMASK(***mask***))** **{PRINT|NOPRINT}** **{TERMINAL|NOTERMINAL}** **{TEST|NOTEST}**

The following list defines the valid parameters.

Parameter	Description
CBDUMP	Format a subset of the OAM control blocks (all of the following).
CTCB[(*address***)]**	Format the OAM Storage Manager component control task control block (CTCB) at the specified address.
CTCL[(*address***)]**	Format the CTC adapter list (CTCL) at the specified address.
CTCWA[(*address***)]**	Format the channel to channel work area (CTCWA) at the specified address.
DTCB[(*address***)]**	Format the drive task control block (DTCB) at the specified address.
HMCB[(*address***)]**	Format the OAM Storage Manager component master control block (HMCB) at the specified address.
IBUFL[(*address***)]**	Format the Object Storage and Retrieval component buffer list (IBUFL) at the specified address.
IQEL[(*address***)]**	Format the Object Storage and Retrieval component query element (IQEL) at the specified address.
IRWA[(*address***)]**	Format the Object Storage and Retrieval component request work area (IRWA) at the specified address.
LAMB[(*address***)]**	Format the library attention message block (LAMB) at the specified address.
LAQ[(*address***)]**	Format the library attention queue (LAQ) entry at the specified address.
LCB[(*address***)]**	Format the library control block (LCB) at the specified address.
LPAQ[(*address***)]**	Format the library attention queue (LPAQ) entry at the specified address.

LQE[(*address***)]**	Format the library queue element (LQE) at the specified address.
LTCB[(*address***)]**	Format the library task control block (LTCB) at the specified address.
MCB[(*address***)]**	Format the OAM master control block (MCB) at the specified address.
MWA[(*address***)]**	Format the Object Storage and Retrieval component main work area (MWA) at the specified address.
MWAX[(*address***)]**	Format the Object Storage and Retrieval component main work area extension (MWAX) at the specified address.
ODCB[(*address***)]**	Format the optical drive control block (ODCB) at the specified address.
OMCB[(*address***)]**	Format the Object Storage and Retrieval component object manager control block (OMCB) at the specified address.
OPCH[(*address***)]**	Format the OAM Storage Manager component optical request chain header (OPCH) at the specified address.
OPCM[(*address***)]**	Format the OAM Storage Manager component operator command parameter block (OPCM) at the specified address.
OPL[(*address***)]**	Format the OAM parameter list (OPL) at the specified address.
OSMA[(*address***)]**	Format the Object Storage and Retrieval component object storage manager area (OSMA) at the specified address.
OVT[(*address***)]**	Format the optical vector table (OVT) at the specified address.
PCB[(*address***)]**	Format the OAM Storage Manager component process control block (PCB) at the specified address.
RWA[(*address***)]**	Format the recovery work area (RWA) at the specified address.
SCB[(*address***)]**	Format the slot control block (SCB) at the specified address.
SECB[(*address***)]**	Format the OTIS Subsystem event control block (SECB) at the specified address.
TVCB[(*address***)]**	Format the tape volume control block (TVCB) at the specified address.
VCB[(*address***)]**	Format the volume control block (VCB) at the specified address.

VSCB[(*address*)] Format the volume set control block (VSCB) at the specified address.

WTD[(*address*)] Format the work to do entry (WTD) at the specified address.

Note: When specifying control block(s) to be formatted, the address is optional. If an address is specified, only the control block at that address is formatted. If an address is not specified, all control blocks of that type are formatted.

For information about the other VERBEXIT parameters, see *z/OS MVS IPCS Commands*.

Diagnosing DB2 Problems

These are some common DB2 problems that may occur during initialization and use of OAM.

When running CBRSAMPL to establish the optical configuration database:

- You must have SYSADM (system administrator) authority to create the required storage group. If the job fails because of a lack of authority, run it under a user ID having SYSADM authority.
- Be sure references to a storage group have been changed to match the name chosen for the storage group created in the first step of CBRSAMPL.

When using DB2I:

- If DSNE110E ("ssid NOT VALID SUBSYSTEM ID, COMMAND TERMINATED," where "ssid" is most often DSN, the default subsystem identifier (ssid)) is issued, set the subsystem name to the name of your DB2 subsystem.

During OAM initialization:

- If OAM fails to initialize and message CBR0100I, CBR0120I, CBR0140I, CBR0160I, or CBR0180I is issued, consult the accompanying DSNT501I message for the corresponding reason and type codes. If the SQLCODE in CBR0xxxI is -904, and the type and reason codes in DSNT501I are 200 and 00C90081 respectively, the table space corresponding to the table in CBR0xxxI is probably not started. Use a -DISPLAY DATABASE command from DB2I option 7 to verify that this is the case, and then use a -START DATABASE command to correct the problem.
- If OAM fails to initialize and message DSNE106E is issued, OAM does not have the proper authority to execute application plan CBROAM. Use SPUFI to execute an SQL GRANT command which gives execute authority on plan CBROAM to the public.
- If OAM fails to initialize with SQLCODE of -818, the level of your DBRMs does not match the level of your object code. Run CBRABIND, the OAM bind job. After receiving message DSNT200I BIND FOR PLAN CBROAM SUCCESSFUL, try to start OAM again.

When using OAM:

- If frequent resource contention problems occur, check the isolation level parameter used when each application plan was bound. For the optical configuration database, it must be CS for cursor stability, except in the CBRHDUPD plan used by OSMC, which uses level RR for repeatable read.

Call Attachment Facility (CAF) Trace

The DB2 Call Attachment Facility (CAF), which OAM uses to connect to DB2, has a trace that provides important diagnostic information. To activate the CAF trace, add the following JCL statement as the last JCL statement in the OAM cataloged procedure (PROCLIB member OAM) and restart OAM:

```
//DSNTRACE  DD  SYSOUT=A
```

This data definition statement sends all of the CAF diagnostic trace messages to SYSOUT. To examine the messages, view SYSOUT after stopping OAM. For more information and examples about the CAF trace output, see *DB2 Diagnosis Guide and Reference*.

DB2 Trace

Use the DB2 START TRACE and STOP TRACE commands to trace DB2 system activity. You can tailor these commands to trace specific application plans or all application plans. You can record the output of the trace using the generalized trace facility (GTF), remote entry services (RES), or system management facilities (SMF). For more information on the DB2 trace, see *DB2 Command Reference* and *DB2 Utility Guide and Reference*.

DB2 Log Data

The DB2 service aid DSN1LOGP can be used to read the contents of the recovery log and format it for display. You can specify both (1) the range of the recovery log you want processed, and (2) certain criteria within the range to display only those records you want to see. For information about how to use DSN1LOGP, see *DB2 Diagnosis Guide and Reference*.

Asynchronous Adapter Device Driver

With the exception of the Get Version Number function call, when the device driver returns control to the calling program after a function call, a function call return code is returned in the AX register. These status codes are shown in Table 93.

Table 93. Function Call Return Codes in the AX Register

AX Register Contents	Description
00	Function call complete.
01	Function call unknown.
02	Asynchronous adapter not responding.
03	Library returns NACK to message.
04	Library not responding.
05	Communications not enabled.
06	Not used at this time.
07	Not used at this time.
08	No pending messages in receive message buffer.
09	Receive message buffer overrun attempted.
0A	Library failed to complete message it was sending.
0B	Library responded to command message with other than ACK or NACK.
0C	Asynchronous adapter failed to complete sending message to library.

SCSI Adapter Device Driver

When the device driver returns control to the calling program after initiating (and completing in most cases) the prescribed function, a function call return code is returned in the AX register. These codes are shown in Table 94.

Table 94. Function Call Return Codes in the AX Register

AX Register Contents	Description
00	Function call accepted.
01	Function call unknown.
02	A device with this SCSI ID is not available.
03	Invalid data pointer.
04	Previous function call for this drive not yet complete.
05	Previous data transfer for any drive not yet complete.
06	Unknown error.
07	Device driver internal error.
08	Device time out.
09	Device driver busy.
0A	DMA boundary crossing.

SCSI Adapter Completion Code

The SCSI Adapter Completion Code is the code that is returned by the device driver when the SCSI command is completed. These codes are shown in Table 95.

Table 95. SCSI Adapter Completion Code

Completion Code Contents	Description
00	Command completed without error.
21	Adapter command tag already in use, prior command with this tag not completed.
24	SCSI CDB byte count error.
25	Invalid SCSI ID.
41	Adapter data buffer pointer conflict.
42	Adapter data buffer pointer conflict.
43	Adapter DMA error.
44	SCSI bus parity error.
45	SCSI device disconnected on odd byte boundary.
47	Adapter unsuccessful in selecting target.
48	SCSI status byte not received from device.
81	Adapter detected differential sense fault.
82	Adapter detected bad SCSI terminator power.
83	Adapter detected SCSI bus timeout.
84	Adapter detected timeout.

Chapter 12. OPEN/CLOSE/EOV (Common) Diagnostic Aids

The following major diagnostic aids are provided for common Open/Close/End-of-Volume:

- An explanation of how to use the transfer control table
- An explanation of how to use error records for debugging
- Description of system fields in various control blocks

Using the Transfer Control Table

The transfer control table can be used to add a temporary change to an O/C/EOV module and to locate a given CSECT in a storage dump.

Locating an Object Module in a Storage Dump

Unlike other system control program components, the module name of O/C/EOV modules appears at the **end** of the module in the storage dump. The 8-byte module name follows the maintenance patch area and precedes the date of assembly, the release or PTF number, and length of the module. Figure 14 shows the module name of module IFG0202J. The object code **precedes** this module name.

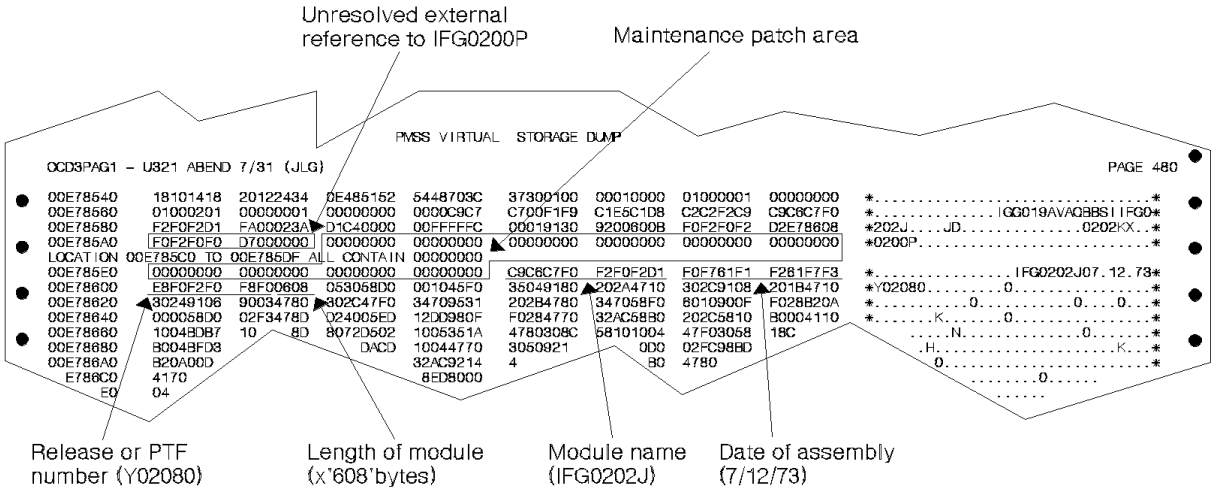


Figure 14. Storage Dump Showing the Transfer Control Table for Module IFG0202J

© Copyright IBM Corp. 1979, 2002

Adding a Temporary Change in the Maintenance Patch Area of an O/C/EOV Module

A maintenance patch area is included in each O/C/EOV object module as the result of the expansion of the XCTLTABL macro. The maintenance patch area is 5% of the size of the object module and is located at the end of each module, immediately preceding the 8-byte module name. You can use the AMASPZAP utility to add a modification to an object module:

1. Locate the transfer control table in the source code.
2. Locate the maintenance patch area in the transfer control table. The beginning of the patch area is the third label from the end of the transfer control table (and the end of the module).
3. Using the displacement (location-counter value) of the maintenance patch area, verify that the patch area is all zeros, and apply the modification to the object module.

Notes:

1. Remember to perform a cold start of the system after the alteration. A warm start will not reformat the PLPA to include your modification.
2. Unresolved V-type address constants (VCONs), which are 4 bytes of zeros, sometimes immediately precede the maintenance patch area. Be careful not to modify the VCONs when adding a patch.

Using Error Records for Debugging

The O/C/EOV problem determination and system recovery routines record work areas and control blocks when abnormal termination conditions are encountered during O/C/EOV processing. The sections that follow describe the kinds of information recorded and how to use it. How to invoke the optional work area tracing and module-name listing routines is also described.

Abend Error Recording Initiated by O/C/EOV

Both the O/C/EOV problem determination and recovery routines trace control blocks and work areas when abnormal termination conditions are encountered in O/C/EOV processing.

The problem determination routines are invoked only when the error is associated with known, describable errors. These errors, called **determinate errors**, are identified with a system completion code and a return code in register 15. The problem determination routines are not invoked when an indeterminate error (for example, a program check) is encountered.

The O/C/EOV system recovery routines are called by the system recovery/termination manager (R/TM) as a result of the ESTAE macro issued during open, close, or EOV initialization. These recovery routines are called on every abend condition whether determinate or indeterminate.

The problem determination and recovery routines pass information to the Generalized Trace Facility (GTF). The records passed to GTF appear in the abend dump. The records will also be written to the GTF data set under the following conditions: (1) if GTF is active and operating in external mode while the job to be traced is running and (2) if the operator responds **TRACE=USR** when the GTF message **SPECIFY TRACE OPTIONS** appears at the console. To correlate these events with other events, you may also want to trace SVCs, I/O events, and others. GTF makes the records available to the ABDUMP routine and writes them to the

GTF data set specified by the user. The user must define the data set for the GTF output. For information on how to use GTF, see *z/OS MVS Diagnosis: Tools and Service Aids*.

The results of the GTF trace that are put into the GTF data set can be examined by formatting and printing the records through IPCS. While in an IPCS session, enter the following IPCS subcommand:

```
GTFTRACE USR(DMA1)
```

DMA1 is the name GTF uses to identify data put in the data set by O/C/EOV routines.

See *z/OS MVS IPCS User's Guide* and *z/OS MVS IPCS Commands* for detailed information on using IPCS.

Recording of abends within O/C/EOV can be controlled by 2 bits at CVTDMSR (X'110' into the CVT).

Bit 0	DMSSDUMP	Setting bit 0 on will cause an SDUMP and message IEC999I to be issued for all abends within O/C/EOV.
Bit 1	DMSUDUMP	Setting bit 1 on will cause an abend dump for all abends within O/C/EOV.

Abend Error Recording by Problem Determination Routines

When a determinate error arises during O/C/EOV processing or during open or close access method executor processing, the problem determination routines gather data related to the abnormal termination condition and related information, and then issue the GTRACE macro to pass the information to GTF. After issuing the GTRACE macro, the O/C/EOV routines issue the abend macro instruction to pass control to the R/TM. The information passed to GTF will also appear in the abend dump and will be written to the GTF data set. Examples of GTF output appears in Figure 15 on page 250 and Figure 18 on page 258.

```
SAMPLE GTF TRACE OF 213.04 ABEND WITH OPEN WORK AREA TRACE        EXTERNAL TRACE  -  DO TRACE                                                    PAGE   003

*** DATE   DAY 274    YEAR 1973   TIME                01.22.18
                        DEVICE INTERFACE
                        OOOOOOOO  OOOOOOOO  OOOOOOOO  OO8OOOOO  OOOOOOOO
                        DCBUFND   DCBBUFCR  DCBBUFCR  DCBDSORG  DCB1OBAD  DCHLARC   DCBEDDAD  DCBRECFM  DCBLXLST  DCBT1OT
                        OO        OOOOOO1   OOOO      4OOO      OOOOOOO1  OO        OO4859C   8O        OOOOOO    OO54
                        DCBMACP6  DCBIFIGS  DCBDEBAD  DC8OFLGS
                        5OOO      OO        O63DEO    C3
                        ACCESS METHOD INTERFACE
                        OO5OOOO   OOOOOOO1  OOOOOO50  OOOOOOOO  OOOOOOOO  OOOOOOO1  OOOOOO5O  OOOOOOO1  OOOOOO5O  OOOOOOO1
                        OOOOOO    OOOOOOO1
                TIME    8O738.51586C
USRFF FFF WKAREA 1      VOL LBL. FILE  LRD. DSCBS. CR  MSG AREA                    +10       +14       +18       +1C       +2C       +24
                        OOOOOOO1O      OOO63DD8       8CCAB73C   4OOOA3AA  OOO63DEO  8OO4B514  OOO63FC8  OOO48518  OOO63FFO  OOO1DFOC
                        +28            +2C            +3O        +34       +38       +3C       +4O       +44       +48       +4C
                        OOO1EC8        OOOOBSC8       OOOOOO6    OOOOO1O   OOOOOO8   OOOOOOO   OOOOOOO   OOOOOOO   OOOOOOO   OOOOOOO
                        +5O            +54            +58        +5C       +6O
                        OOOOA4EE       OOO63680       OOC63DEQ   2OFAOOOO  OOOOOOO

        TIME      8O738.702779
*** DATE   DAY  274   YEAR  1973    TIME       01.22.18                                                                                          ***

USRFF  FFF  JFCBDSNM
                        D86CC9D3  C5E74040  40404040  40404040  40404040  40404040  40404040  40404040  40404040  40404040  40404040
                        4O4O4O4O

        TIME            8O739.393241
```

1 A formatted copy of the DCB that was being opened when
2 The first 100 bytes of the O/C/EOV main work area associated with the DCB
3 Had been read in when the error
4 The JFCB being used by the Open routine.
5 The internal I/O control blocks used by the Open routine.
Various fields from the main work area

Figure 15. Output from the GTF Data Set Resulting from Problem Determination Abend Error Recording Routine (213–04 Abend)

This GTF output resulted from a 213 system completion code with a return code of X'04', meaning that the format-1 DSCB for the data set to be opened could not be found.

The problem determination routines write the data set name from the JFCB into the GTF data set.

Abend Error Recording from the O/C/EOV System Recovery Routines

The O/C/EOV system recovery environment is established through the ESTAE macro, issued when storage was obtained during initialization of the O/C/EOV function. The O/C/EOV system recovery routines receive control from R/TM when any abend condition (determinate or indeterminate) is encountered during O/C/EOV processing.

When given control, the O/C/EOV recovery routines free storage, dequeue resources, and force data sets to closed status. Before doing any of these things, however, the O/C/EOV system recovery routines attempt to preserve a picture of the O/C/EOV chained work areas as they exist at time of entry. This is done by building a recovery routine control block save area (RRCBSA). Copies of the O/C/EOV chained work areas are saved in the RRCBSA.

As with the problem determination routines, the O/C/EOV system recovery routines issue the GTRACE macro instruction in order to record the contents of the RRCBSA built on entry. The RRCBSA segments are made available to the ABDUMP routine and written to the GTF data set by GTF.

When O/C/EOV recovery processing has been completed, the RRCBSA is used to refresh the O/C/EOV chained work areas, which may have been modified during recovery processing. The RRCBSA is then freed. Note that an RRCBSA will not appear in a dump of the original error, but it will appear in a dump of an error that occurs during recovery processing.

The contents of the RRCBSA are defined in Figure 16. An example of a dump of this save area is shown in Figure 17 on page 252.

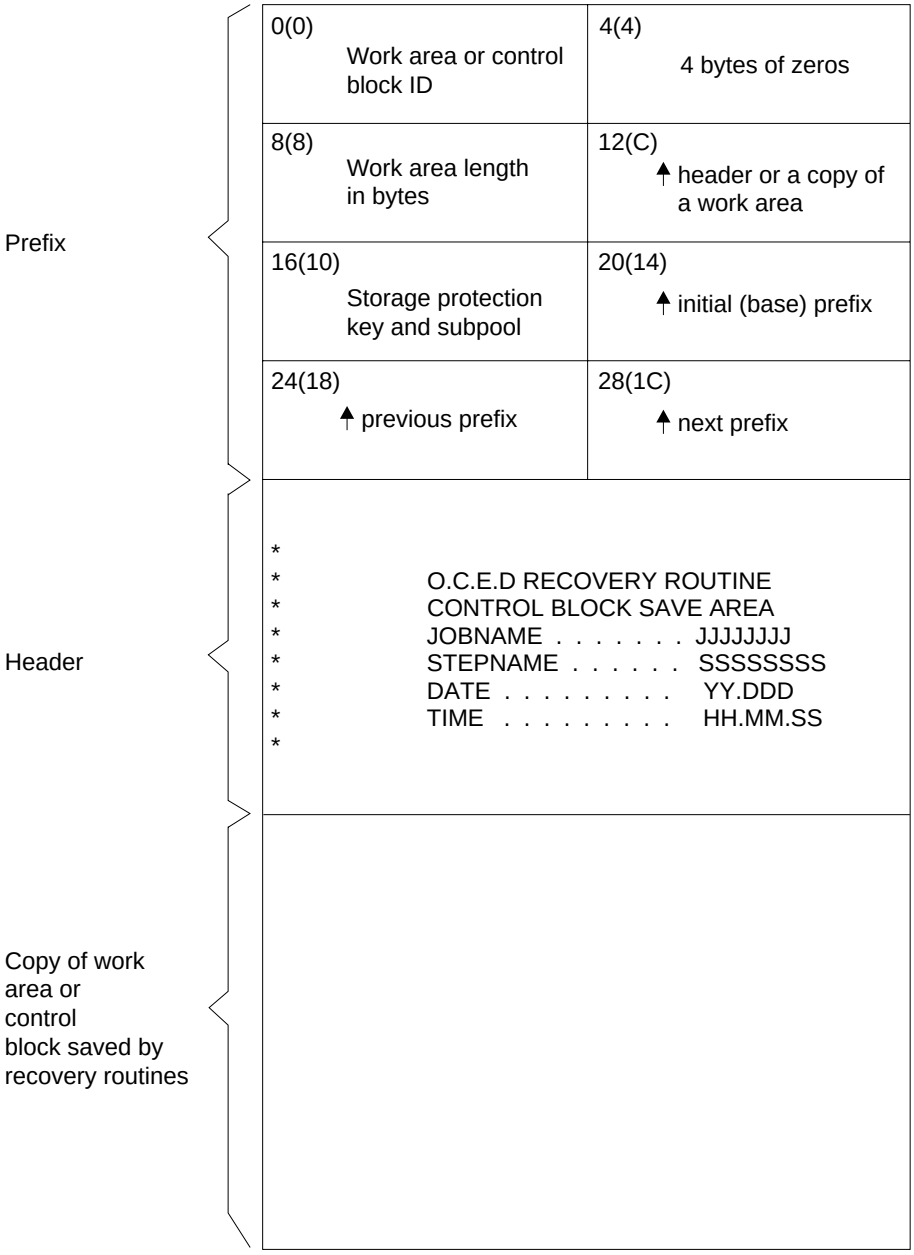


Figure 16. Contents of Recovery Routine Control Block Save Area (RRCBSA)

```
                                PMSS VIRTUAL STORAGE DUMP                                                    PAGE 001
Fourth
segment  0A100  06C3C5C4 00010101 01100108 11A1A1E  50111056 01CAA5D8 110AA260 011AA508  *CBSA..............W...Q......Q*
         0A180  C3C3F6F2 00010101 011A8108 11A1A88  50111056 01CAAF20 110AAC18 011AAF20  *CCWS..............W...........*
         1110A200 81CAAE50 6C010151 011A8108 11101000  29111D16 01000020 11000000 01100001  *..........0...................*
         1110A220 08CAAB58 00010161 211A0110 11101002  08111880 00000000 11CAACA8 01100060  *...V..........................*
         1110A240 12CAAF58 00010181 01100108 11101000  00111000 00000000 00000000 01100000  *..............................*
Third    0A260  D6C3C5C4 00010101 01100108 11101028  50111056 010AA508 110AA218 01100060  *CBSA..............W...Q......*
segment  0A280  C4C3C240 00010101 01100108 11101028  50111056 010AA508 110AA218 01100060  *DCB...............W...........*
         0A2A0  C4C3C240 00010101 01100108 11101028  50111056 010AA508 110AA218 01100060  *.............................*
         1110A200 C4C3C240 00010101 01100108 11101028  50111056 010AA508 110AA218 01100060  *.............................*
         1110A250 C4C3C240 00010101 01100108 11101028  50111056 010AA508 110AA218 01100060  *.......................CBSA...*
         1110A300 C4C3C240 00010101 01100108 11101028  50111056 010AA508 110AA218 01100060  *...........W..0...0.........*
         1110A320 C4C3C240 00010101 01100108 11101028  50111056 010AA508 110AA218 01100060  *OPWA..............W...........*
         1110A340 D6D7F6C1 00010108 01100108 11101028  50111056 010AA508 110AA218 01100060  *A.............................*
         1110A360 F4D7F6C1 00010141 01100108 11141000  50111056 010AA508 110AA218 01100060  *.............................*
         1110A380 0007F6C1 00010141 01100108 11141000  50111056 010AA508 110AA218 01100060  *.............................*
         1110A3A0 0007F6C1 00010141 01100108 11141000  50111056 010AA508 110AA218 01100060  *....OFID5X                  *
         1110A300 0007F6C1 00010141 01100108 11141000  50111056 010AA508 110AA218 01100060  *                        ....  *
         1110A390 0007F6C1 00010141 01100108 11141000  50111056 010AA508 110AA218 01100060  *.........................  ...*
         1110A400 1007F6C1 11010141 01100108 11141000  50111056 010AA508 110AA218 01100060  *.......................000LIB*
         1110A420 0107F6C1 11010141 01100108 11141000  50111056 010AA508 110AA218 01100060  *                        ........*
Second   1A440  0017F6C1 10110141 01100108 11141000  50111056 010AA508 110AA218 01100060  *.............................*
segment  1A460  0107F6C1 10110141 01100108 11141000  50111056 010AA508 110AA218 01100060  *....................8.......*
         1A480  1007F6C1 11010141 01100108 11141000  50111056 010AA508 110AA218 01100060  *.............4...8...........*
         1110A4A0 0017F6C1 10110141 01100108 11141000  50111056 010AA508 110AA218 01100060  *.............................*
         1110A400 0107F6C1 01010141 01100108 11141000  50111056 010AA508 110AA218 01100060  *.........H.................Y.0*
         1110A450 1007F6C1 01010141 01100108 11141000  50111056 010AA508 110AA218 01100060  *........5G4A7............5A....*
         1110A500 0017F6C1 01010141 01100108 11141000  50111056 010AA508 110AA218 01100060  *.......................000LIB*
         1110A520 0107F6C1 00110141 01100108 11141000  50111056 010AA508 110AA218 01100060  *.............................K.*
         1110A540 1007F6C1 10110141 01100108 11141000  50111056 010AA508 110AA218 01100060  *..|...|....8..H..............*
         1110A560 0007F6C1 00010141 01100108 11141000  50111056 010AA508 110AA218 01100060  *....................M.........*
         1110A580 0007F6C1 00010141 01100108 11141000  50111056 010AA508 110AA218 01100060  *.......................CBSASA..*
         1110A5A0 0007F6C1 00010141 01100108 11141000  50111056 010AA508 110AA218 01100060  *...........W..0..............*
         1110A590 0007F6C1 00010141 01100108 11141000  50111056 010AA508 110AA218 01100060
         1110A590 0007F6C1 00010141 01100108 11141000  50111056 010AA508 110AA218 01100060
         111CTION 0007F6C1 0011F1L1L 11TA1IN 11101404
         1110A620 40404040 40014101 01100108 11141000  50111056 010AA508 110AA218 01100060  *    0.0.5.0  HFCOVERV ROUTINE  *
         1110A640 40404040 40014101 01100108 11141000  50111056 010AA508 110AA218 01100060  *    CONTROL BLOCK SAVE AREA    *
         1110A650 40404040 40014101 01100108 11141000  50111056 010AA508 110AA218 01100060  *    JOBNAME ....... 000T001X   *
         1110A660 40404040 40014101 01100108 11141000  50111056 010AA508 110AA218 01100060  *    STEPNAME ...... 000T001    *
         1110A680 40404040 40014101 01100108 11141000  50111056 010AA508 110AA218 01100060  *    DATE .......... 73.258     *
         1110A690 40404040 40014101 01100108 11141000  50111056 010AA508 110AA218 01100060  *    TIME .......... 10.46.40   *
First       TION 0007F6C1 0011F1L1L 11TA1IN 11101404
segment  A700  0007F6C1 00010141 01100108 11141000  50111056 010AA508 110AA218 01100060  *Q.C........... ...W...........*
         1110A720 0007F6C1 00010141 01100108 11141000  50111056 010AA508 110AA218 01100060  *............ 7...............*
         1110A740 0007FFC4 00010141 01100108 11141000  50111056 010AA508 110AA218 01100060  *...D..H.......................*
         1110A760 0007F001 00010141 01100108 11141000  50111056 010AA508 110AA218 01100060  *...............M.......0.......*
         1110A780 0007F001 00010141 01100108 11141000  50111056 010AA508 110AA218 01100060  *....IFG0198N..................*
         1110A7A0 0007F004 00010141 01100108 11141000  50111056 010AA508 110AA218 01100060  *....8N........................*
         1110A730 0007F004 00010141 01100108 11141000  50111056 010AA508 110AA218 01100060  *.............................*
         1110A750 0007F004 00010141 01100108 11141000  D9                                  *................8*
```

(A) Prefixes to the RRCBSA segments.

(B) RRCBSA header.

(C) Previs included in the copy of the work area or control block.

(D) Copy of a control block or work area obtained by the main-line Open routines. Each area is identified by an ID in the first four bytes of its prefix (see C).

Storage locations used to align IDs in ythe EBCDIC column.

Figure 17. Storage Dump Containing the RRCBSA Passed by the O/C/EOV Recovery Routines

The recovery routine gained control in this example when a 213-04 error condition was encountered (no format-1 DSCB could be found for the data set being opened).

The RRCBSA is made up of segments. Each segment begins with an ID of CBSA, which appears in the EBCDIC column of the dump. The first CBSA segment obtained is different from subsequent segments; in addition to a prefix and a section containing a copy of a control block or work area, the first segment contains a header. The header identifies the RRCBSA, provides the job and step name of the step that encountered the 213 abend condition and the date and time the error occurred. In Figure 17, the bottom-most segment, which contains the header, is the first segment. This segment is chained to the segments obtained subsequently using the prefix. In addition to the header, the first segment contains a copy of the initial work area obtained to open the DCB. The ID of this work area is O/C, which

appears as “O.C” in the EBCDIC column. This initial work area contains a copy of the base prefix, an extended prefix, and a WTG table. Addresses in these copies reflect original storage locations. The contents of these locations may have been changed by recovery or termination processing. The address in these copies can be used, however, to find information in the copies.

The second segment of the example contains a copy of the main work area associated with the DCB. Its ID is “OPWA”. The third segment contains a copy of the DCB (ID is “DCB”), and the fourth segment is an area that contains a channel program (ID is “CCWS”). Each of these segments is chained from the first segment using the prefixes. The prefix of each segment begins with the ID “CBSA”.

Each of the work areas obtained by the O/C/EOV routines has been assigned a 4-byte ID to make the work areas easy to identify in storage dumps. These IDs appear in the first word of every prefix of every work area obtained by O/C/EOV routines and saved in the RRCBSA. Each of the work areas and control blocks is obtained with a prefix. The contents of the prefix are described in Figure 16 on page 251.

Abend Error Recording in SYS1.LOGREC

The system diagnostic work area (SDWA) is passed among the various levels of system and user recovery (STAE, ESTAE, and FRR) routines. It is used to pass information among the recovery routines and to collect and record information about the environment in which the error occurred. The O/C/EOV recovery routines put the following information in the SDWA:

- The CSECT name of the O/C/EOV routine that encountered the error will be moved into the SDWAMODN field.
- The CSECT name of the O/C/EOV routine that encountered the error will also be moved into the SDWACSCT field.
- The CSECT name of the O/C/EOV recovery routine handling the error will be moved into the SDWAREXN field.
- The O/C/EOV recovery routine moves the O/C/EOV recovery routine parameter list (RRPLIST) into the variable area of the SDWA. The contents of the list is shown in Table 96.

Table 96. O/C/EOV Recovery Routine Parameter List

Offset Dec(Hex)	Length	Name	Description
0 (0)	1	RRFUNCTN	Identifies which O/C/EOV or DADSM function was in control at the time the error occurred.
	...1 ..11	RRFOPEN	Open.
	...1 .11.	RRFOPENJ	Open, Type=J.
	...1 .1..	RRFCLOS	Close.
	...1 .111	RRFTCLOS	Close, Type=T.
	..11 .111	RRFEOV	End-of-Volume.
	...1 1111	RRFFEOV	FEOV.
	.1..	RRFRJFCB	Read JFCB.
	1...	RRFDADSM	Indicates that one of the following DADSM functions was in control at the time the error occurred.
	1.1.	RRFALLOC	DA space allocation.

Table 96. O/C/EOV Recovery Routine Parameter List (continued)

Offset Dec(Hex)	Length	Name	Description
	1.11 .111	RRFEXTND	Extend.
	1..1 .1..	RRFPRLSE	Partial release.
	11.. 111.	RRFLSPAC	LSPACE.
	1..1 1.11	RRFOBTN	Obtain.
	1..1 111.	RRFRENAM	Rename.
	1..1 11.1	RRFSCRTH	Scratch.
1 (1)	1	RRFLAGS1	Identifies the functions that have received control from the O/C/EOV and DADSM modules.
	.1..	RRFUSREX	User exit routine in control.
	..1.	RRFAMEXG	Access method executor in control.
	...1	RRFAMEXR	Access method executor returned control.
	.... 1...	RRFAMSTR	Access method executor string (for example, VTAM or VSAM) in control.
	.... .1..	RRFNSL	User's NSL routine given control.
	.... ..1.	RRFSMF	SMF in control.
	1	RRFTRACE	Trace routine in control.
2 (2)	1	RRFLAGS2	Indicates key functions completed.
	1...	RRFFIN1	First object module of the function has completed.
	..1.	RRFENQPW	Enqueue on the PASSWORD data set.
	...1	RRFQMNGR	QMNGR component given control.
	.... ..1.	RRFPRDET	The O/C/EOV problem determination routines have been given control.
	1	RRFENQUL	Enqueue on a user label track.
3 (3)	1	RRFLAGS3	Indicates the type of termination that is in progress.
	.1..	RRFNRTCA	No SDWA (STAE diagnostic work area, previously called RTCA) was passed.
	..1.	RRFINFC	Final pass made by the FORCE CLOSE function.
	...1	RRFFCEXC	FORCE CLOSE executor in control.
	.... 1...	RRFTERM	Special termination entry.
4 (4)	4	RR1WRKAR	Address of the recovery routine work area.
8 (8)	4	RR1RETRY	First-level recovery routine bypass address.
12 (C)	4	RRXRETRY	For O/C/EOV, this is the recovery routine executor bypass address.
		RRUCBPTR	For DADSM, this is the UCB address.

Table 96. O/C/EOV Recovery Routine Parameter List (continued)

Offset Dec(Hex)	Length	Name	Description
16 (10)	4	RRMLRTRY	O/C/EOV retry pointer.
20 (14)	4	RRFWORK	Internal communications word.
24 (18)	4	RRFWORK1	Internal communications word.

Problem Determination Optional Work Area Trace

The optional work area trace, normally used when a task fails repeatedly with evidence of an O/C/EOV problem, traces the DCB and the O/C/EOV work area and extended work area associated with the DCB. This trace also includes a record of the address of the WTG table and the address of the current entry in the WTG table after each O/C/EOV module executes.

The user requests the optional work area trace by either specifying DCB=DIAGNS=TRACE in the DD statement of a data set for which the trace is desired or by using the problem determination optional work area trace started task, IFGOCETR, instead of specifying DCB=DIAGNS=TRACE. See for information on the started task. The work area and control blocks are written to the GTF data set every time IFG019RA is entered. No tracing is done when control is passed to the O/C/EOV problem determination and recovery routines, the initial and final modules of each function of O/C/EOV. Tracing occurs only during access method executor processing for the SAM, BDAM create, and BPAM executors.

Figure 18 on page 258 shows the GTF output resulting from activating the optional work area trace. In this example, the Open routines abnormally terminated the task because the format-1 DSCB for the data set could not be found. The associated system and return codes are 213-04. The trace record in Figure 18 includes a copy of the DCB, a copy of the O/C/EOV main work area associated with the DCB, various fields from the main work area, and the WTG table. This record will appear several times in the GTF output, once each time an Open object module has been given control.

Because the optional work area trace increases execution time considerably and produces copious output, it should be used only when other diagnostic techniques fail.

Non-VSAM Problem Determination Work Area Trace Started Task

DFSMSdfp provides a problem determination work area trace started task, IFGOCETR, to allow tracing of dynamically allocated non-VSAM data sets. The started task will create a trace table in CSA/ECSA storage. Once the area is created it will remain for the duration of the IPL. This is intended to be used for dynamically allocated data sets but can also be used instead of specifying DCB=DIAGNS=TRACE on the DD statement of a data set for which tracing is desired.

The work area trace started task is activated by the operator specifying **S IFGOCETR**. The operator must then respond to message IEC980A, providing the data set name and/or the job name and/or the DD statement name identifying the data set(s) for which tracing is desired. Any combination of the above specifications is allowed, but the more specific the information provided, the more selective O/C/EOV will be in determining which data set(s) are to be traced. For example, if only a job name is specified, then all data sets processed by that job would be eligible to be traced. This is probably not what was intended. Therefore, it is

recommended to specify as many keys as necessary to limit tracing to the specific data set to be traced. Message IEC980I is issued for both informational and error purposes. The maximum number of trace table entries that can be active is 10. If a request is made to activate an entry when the maximum has been reached, then the operator will be requested to delete one or all of the currently active entries. The operator also has the option to display the active entries list. See “Work Area Trace Started Task Messages” for more information concerning messages IEC980A and IEC980I. The active trace table entries will not actually be traced until the operator starts GTF activating the optional work area trace.

The started task does not have to be active in order to activate the GTF trace. Once the table entries are built, the started task can be terminated.

Work Area Trace Started Task Messages

IEC980A SPECIFY TRACE KEYS: DSN= ,DDN= ,JN= ,DISPLAY ,DELETE, ALL|NUM END

Descriptions of the keys follow:

- DSN=data set name. The 1 to 44 character name of the data set to be traced should be specified. If this is a GDG, then either the entire GDS name can be specified (limiting tracing to that specific GDS) or the GDG base name can be specified (all GDS data sets which satisfy any other keys specified will be traced). If DDN and JN aren’t specified, then tracing will occur for that data set independent of the job and DD statement names.
- DDN=DD statement name. The 1 to 8 character name of the DD statement whose data set should be traced. If DSN and JN aren’t specified, then every data set that is processed by every job which has the specified DD statement name will be traced.
- JN=job name. The 1 to 8 character name of the job that is to be traced. If DSN and DDN aren’t specified, then every data set on every DD statement processed by the specified job will be traced.
- DISPLAY will display all active entries.
- DELETE will delete one or all of the active entries in the table. If a specific entry is to be deleted, specify the NUM of that entry. Specify ALL if all entries are to be deleted.
- END to end the started task.

When a table entry is created, at least one of the keys (DSN, DDN, or JN) must be specified and if more than one of those keys are specified, they must be specified in one response of message IEC980A. For instance, if DSN=A is specified, then data set A will be traced. If DSN=A,JN=JOB1 is specified, then data set A when processed by job JOB1 will be traced. However, if DSN=A is specified and then in the next response to message IEC980A JN=JOB1 is specified, then data set A and all data sets processed by JOB1 will be traced. Each response to message IEC980A is related to one table entry. The next response to message IEC980A will be related to the next available entry. In order for a data set to be traced it must match all of the keys specified. Any keys that aren't specified will not be checked.

IEC980I INVALID REPLY|NO ACTIVE ENTRY(S) FOUND|MAXIMUM TRACES ACTIVE. DELETE 1 OR ALL, OR REPLY END|START GTF TO ACTIVATE OCE TRACE

Descriptions of the keys follow:

- INVALID REPLY-a key specified in the response to message IEC980A is invalid. Message IEC980A will be reissued so the correct key can be specified.
- NO ACTIVE ENTRY(S) FOUND- a request to display active entries was specified, but there are no active entries in the table. Message IEC980A will be reissued.
- MAXIMUM TRACES ACTIVE-DELETE 1 OR ALL, OR REPLY END - a request was made to add an entry to the table but there are already 10 active entries. Therefore, one or all of the active entries must be deleted before a new entry can be added, or specify END. Message IEC980A will be reissued.
- START GTF TO ACTIVATE OCE TRACE is an informational message to inform the operator that one or more table entries have been successfully added, but that GTF must still be started to activate the optional work area trace if it is not already started.

```
SAMPLE GTF TRACE OF 213-04 ABEND WITH OPEN WORK AREA TRACE      EXTERNAL TRACE - DO TRACE                          PAGE 0003
*** DATE   DAY 274    YEAR 1973   TIME          01.22.18                                                                  ***
USRFF FFF DCB      DEVICE INTERFACE
                   00000000  00000000   00000000  00000000  00000000
                   DCBBUFNO  DCBBUFCR   DCBBUFL   DCBDSORG  DCBIOBAD   DCHIARC   DCBEDDAD  DCBRECFM  DCBEXLST  DCBTIOT
                   00        000001     0000      4000      00000001   00        04859C    80        000000    0054
        (A)        DCBMACPF  DCBIFLGS   DCBDEBAD  DCBOFLGS
                   5000      00         063DE0    C3
                   ACCESS  METHOD  INTERFACE
                   005000    00000001   00000050  00000000  00000001   00000001  00000050  00000001  00000050  00000001
                   00000000  00000001
          TIME    80738.51586C
USRFF FFF WKAREA 1 VOL LBL, FILE LRD, DSCBS, CR MSG AREA    +10       +14       +18       +1C       +20       +24
                   00000010  00063DD8   8CCAB73C  4000A3AA  00063DE0  8004B514  00063FC8  0004B518  00063FF0  000IDF0C
                   +28       +2C        +30       +34       +38       +3C       +40       +44       +48       +4C
        (B)        0001E3C8  0000BSC8   00000006  00000010  00000008  00000000  00000000  00000000  00000000  00000000
                   +50       +54        +58       +5C       +60
                   0000A4EE  00063680   00C63DEQ  20FA0000  00000000
          TIME    80738.519374
USRFF FFF WKAREA 2 JFCR-JFCRCSNM
                   DCB60903  C5E74041   40404040  40404040  40404040  40404040  40404040  40404040  40404040  40404040
                   (CONT.)   JFCMEINW             JFCRTSDM  JFCDSCR   JFCFCRIC  JFCAMPTR  JECRCAPT  JFCRLTYR  JFCROTTR/
                   40404040  40404040   40404040  00        000000    00000000  000000    000000    02        00
                   /JFCRFLSC JFCRVLSA   JFCRMASK            JFCHCRDT  JFCBXPCT  JFCBIND1  JFCBIND2  JFCBUFNO  JFCRWRAP
                   0000      0000       00000000  0000240   21DCF0    000000    00        40        00        00
                   JFCRUEL   JFCFRCRT   DEV CHAR  JFCDEN    JFCLIMCT  JFCDSORG  JFCRECFM  JFCOPTD   JFCALKSI  JFCLRECL
                   0000      00         00        00        000000    0000      00        00        0000      0000
        (C)        JFCNCP    JFCNTM     SEGMENTS            JFCCPRI   JFCSOWA   RESERVED  JFCBNVOL
                   00        00         00000000  00        00        0000      00        01
                   JFCRVCLS                                                                JFCREXTL  JFCREXAD
                   D8D8D8D3  D9C24040   40404040  40404040  40404040  40404040  40404040  4040      00        000000
                   JFCPQTY   JFCRCTWI   JFCBSQTY  RESERVED  JFCBDQTY  JFCBSPNM  JFCBABST  JFCBSBNM  JFCBDRLH  JFCBVLCT
                   000000    00         000000    00        000000    000000    0000      000000    000000    01
                   JFCBSPTN
                   00
          TIME    80738.522724
USRFF FFF WKAREA 3 DXECP
                   00000000
                 * IOB-IOBFLAG1          IOBFLAG2  IOBSENS0  IOBSENS1  IOBECBCC  IOBECRPT  IOBFLAG3  IOBCOMAD  IOBSTAT0
                   02                    00        00        00        7F        C63EF4    00        000000    00
                   IOBSTAT1  IOBCNT      IOBSIGCC  IOBSTART  IOBDCEPT  IOBRESTP  IOBRESTP  IOBINCAM  IOBFPRCT
                   00        0000        40        063F50    00        063F20    00000000  0000      0000
                   DXCAADCR
                   00000000  0000140
        (D)      * DEB-DXCEBDER          DEBOFLGS  DEBIRBAD  DEBOPATR  DEBQSCAT  DEBFLGS1  RESERVED
                   00000000  00         000000    00        00        00        00
                   DEBNMFXT  DEBUSRPG   DEBECRAD  DEBPROTG/DEMDFPIC   DEBDCHAD  DEBEXSCI  DEBAPPAD  DEBDVMOD
                   01        000000     00        000000    0F                  063F20    00        003350    00
                   DEBUCBAD  DEBBINUM   DEBSTRCC  DEBSTRHH  DEBFNOCC  DEBFNDHH  DEBNMTRK
                   000810    0000       0000      0000      000H      0014      7FFF
                 * DCP-DCPIFLGS/DCPDEFAC
                   0006310
          TIME   *0738.526022
USRFF FFF WKAREA 4 7 CCW'S-DXCCW1        DXCCW2              DXCCW3              DXCCW4    DXCCW5
                   31063F18  40400005    08263F23  00000000  92063F80  40400009  29066F44  4000002C  08063F60  00000000
                   DXCCW6    DXCCW7
                   06163DE0  00000000    00000001  08200060
                 * REG SAVE-DXREGS       DXREGA    DXREGR    DXREGD    DXREG5    DXREG6    DXREG1
                   00010500  90000812    00008904  00010F10  00000010  00000000  00000000  00003508  00063F58
                 * MISC-DXRESIND         DXIDENT/DXCALLID    DXRFTMOD/DXRFTIC    DXRFTTTR/AR,REICODF DXXCTL
                   00                    0000                0000                000000              80063F08  00000000
        (E)      * OPEN/CLOSE WORK AREA END WITH MISC FIELDS ON NEXT LINE.  START OF WTG TABLE FOR ENV WORK AREA ONLY.

                 * WTG IRL,MISC-         DXXMCDNM/DXVOLMTI   DXXMODID/DXVOLMT2/DXVOLSEC/DXWORK1        DXWORK2/DXXTTR/
                                         D8D8D803  0902      0000      000000                          00000000
                   DXXECBAD              DXXATTRB  DXXMODSZ  DXXTXTLY  DXOPCLS1  DXXPATHS  DXXENTRY/ /DXXICIIR  DXXCCRE
                   090607F0  F1F9        F4C5      000000    0000      00        0079      0806      800800     000379
                 * DXVOLSR1               DXVOLSR2            DXVOLSR3            DXNGLAIT  DXDCBLST
                   00040024  0206        0080F501  007A      08163DE0  0000      0000      00000000
          TIME   *0738.529350
USRFF FFF WKAREA 5 DCB6090M              MTGMCDIC  PSUEDD                  TGTT            WTGATTRR  WTGMODSZ
                                         F4C4      00000                                   0379      000400
                                         WTQPATH
```

(A) A formatted copy of the DCB that was being opened when the error condition was encountered.

(B) The first 100 bytes of the O/C/EOV main work area associated with the DCB being opened. Because no labels or DSCBs had been read in when the error occurred, this information is not significant.

(C) The JFCB being used by the Open routine.

(D) The internal I/O control blocks used by the Open routine.

(E) Various fields from the main work area followed by portions of the WTG table associated with the OPEN request

Figure 18. Output from GTF Data Set Resulting from the Optional Work Area Trace

DCB for EXCP, SAM, and BPAM

Common Name: Data Control Block for EXCP, SAM, and BPAM
Macro ID: DCBD
DSECT Name: IHADCB

Owning Component: Data Management, subcomponent OPEN/CLOSE/EOV

Eye-Catcher ID: None

Subpool and Key: Problem program subpool and key

Size:

- EXCP — 52 or 72 bytes
- BSAM, BPAM — 88 bytes
- QSAM — 96 bytes

Created by: Problem program

Pointed to by:

- DEBDCBAD field of the DEB data area
- IOBDCBPT field of the IOB data area
- DECDCBAD field of the DECB data area (BSAM and BPAM)
- Register 1 passed to BLDL, STOW, and other macros
- OPEN, CLOSE, and RDJFCB parameter lists
- JSCBDCB field of the JSCB data area (scheduler DCB)
- LWAPDCB field of the LWA data area (UADS DCB)
- TCBJLB field of the TCB data area (JOBLIB DCB)

Serialization: The user is responsible for serialization. While OPEN/CLOSE/EOV processes the DCB, a temporary protected copy of the DCB is made to ensure serialization. Use the DEBCHK macro to check for the validity of the DCB.

Function: The data control block (DCB) is the data area within which data pertinent to the current use of a data set is stored. The DCB formats used with basic sequential access method (BSAM), queued sequential access method (QSAM), basic partitioned access method (BPAM), and execute channel program (EXCP) are very similar.

Offset	Type/Value	Len	Name (Dim)	Description
DATA CONTROL BLOCK DEFINITION				
	X'80'		DCBBIT0	″128″
	X'40'		DCBBIT1	″64″
	X'20'		DCBBIT2	″32″
	X'10'		DCBBIT3	″16″
	X'8'		DCBBIT4	″8″
	X'4'		DCBBIT5	″4″
	X'2'		DCBBIT6	″2″
	X'1'		DCBBIT7	″1″
	DEVICE INTERFACES			
	DIRECT ACCESS DEVICES			
0 (0)	ADDRESS	4	DCBDCBE	DCBE ADDRESS ASSOCIATED WITH THIS DCB IF A DCBE EXISTS. A DCBE EXISTS IF BOTH DCBH0 AND DCBH1 ARE SET ON. (DSORG=PS OR PO)

Offset	Type/Value	Len	Name (Dim)	Description
0 (0)	CHARACTER	4	DCBRELAD	—PARTITIONED ORGANIZATION DATA SET – ADDRESS (IN THE FORM TTRN) OF MEMBER CURRENTLY USED (UNLESS A DCBE EXISTS IN WHICH CASE THIS OFFSET POINTS TO THE DCBE, SEE DCBDCBE). IF A DCBE EXISTS, THE TTRN IS STORED IN THE DCBE (DCBERELA). HIERARCHY BITS DEFINE THE EXISTENCE OF A DCBE. —SEQUENTIAL DATA SET –ADDRESS OF DCBE IF ONE EXISTS.
4 (4)	SIGNED	1	DCBKEYCN	–KEYED BLOCK OVERHEAD CONSTANT
5 (5)	CHARACTER	8	DCBFDAD	–FULL DISK ADDRESS IN THE FORM OF MBBCCHHR OF RECORD THAT WAS JUST READ OR WRITTEN
12 (C)	ADDRESS	4	DCBDVTBL (0)	–SAME AS DCBDVTBA BELOW
12 (C)	BITSTRING	1		– LAST BYTE OF DCBFDAD
13 (D)	ADDRESS	3	DCBDVTBA	– ADDRESS OF ENTRY IN I/O DEVICE CHARACTERISTICS TABLE FOR DEVICE BEING USED
16 (10)	SIGNED	1	DCBKEYLE	– KEY LENGTH OF DATA SET
17 (11)	CHARACTER	1	DCBDEVT	– DEVICE TYPE
FOR MASKS FOR ISAM DIRECT ACCESS, SEE DCBOVDEV IN ISAM SECTION				
	0 010 0 0 01		DCBDV311	″X'21'″– 2311 DISK STORAGE
	0 010 0 010		DCBDV301	″X'22'″– 2301 PARALLEL DRUM
	0 010 0 011		DCBDV303	″X'23'″– 2303 SERIAL DRUM
	0 010 010 0		DCBDV345	″X'24'″– 9345 DISK STORAGE FACILITY
	0 010 0101		DCBDV321	″X'25'″– 2321 DATA CELL STORAGE
	0 010 0110		DCBD1305	″X'26'″– 2305 DRUM MODEL-1
	0 010 0111		DCBDV305	″X'27'″– 2305 DRUM MODEL-2
	0 010 10 0 0		DCBDV314	″X'28'″– 2314/2319 DISK STORAGE FACILITY
	0 010 10 01		DCBDV330	″X'29'″– 3330 DISK STORAGE FACILITY 3330 MODEL-1 3330 MODEL-2 3333 MODEL-1
	0 010 1010		DCBDV340	″X'2A'″– 3340/3344 DISK STORAGE FACILITY
	0 010 1011		DCBDV350	″X'2B'″– 3350 DISK STORAGE FACILITY MODELS A2, B2, AND C2
	0 010 110 0		DCBDV375	″X'2C'″– 3375 DISK STORAGE FACILITY
	0 010 1101		DCBDV331	″X'2D'″– 3330 MODEL–11 OR 3333 MODEL–11 DISK STORAGE FACILITY
	0 010 1110		DCBDV380	″X'2E'″– 3380 DISK STORAGE FACILITY
	0 010 1111		DCBDV390	″X'2F'″– 3390 DISK STORAGE FACILITY
18 (12)	UNSIGNED	2	DCBTRBAL	– TRACK BALANCE. NUMBER OF BYTES REMAINING ON CURRENT TRACK AFTER A WRITE OPERATION (THIS QUANTITY MAY BE NEGATIVE IF THERE ARE NO BYTES REMAINING ON TRACK).
	MAGNETIC TAPE			
0 (0)	CHARACTER	12		–RESERVED FOR I/O SUPERVISOR
12 (C)	SIGNED	4	DCBBLKCT	–BLOCK COUNT FOR EACH VOLUME
16 (10)	CHARACTER	1	DCBTRTCH	–TAPE RECORDING TECHNIQUE FOR 7-TRACK TAPE
	. .1. . .11		DCBMTE	″X'23'″– E – EVEN PARITY
	. .11 1.11		DCBMTT	″X'3B'″– T – BCD/EBCDIC TRANSLATION
	. . .1 . .11		DCBMTC	″X'13'″– C – DATA CONVERSION
	. .1. 1.11		DCBMTET	″X'2B'″– ET – EVEN PARITY AND TRANSLATION 9348 SERIES RECORDING TECHNIQUE
	.1. . . .1.		DCB1TRK	″X'42'″– ONE TRACK MODEL
	1. . . . .1.		DCB2TRK	″X'82'″– TWO TRACK MODEL

Offset	Type/Value	Len	Name (Dim)	Description
	11. . . .1.		DCB4TRK	″X'C2'″– FOUR TRACK MODEL 3480X RECORDING TECHNIQUE
	. . . . 1. . .		DCBCMPAC	″X'08'″ COMPACTION
	. . . . .1. .		DCBNOPAC	″X'04'″ NO COMPACTION
17 (11)	CHARACTER	1	DCBDEVT	– DEVICE TYPE
	1. 1		DCBDVMT	″X'81'″– 3490 MAGNETIC TAPE UNIT (NO LONGER FOR 2400 SERIES TAPE UNITS)
	1. 11		DCBDVMT3	″X'83'″– 3400 SERIES MAGNETIC TAPE UNIT
	1.		DCBDVMT4	″X'80'″– 3480 MAGNETIC TAPE UNIT
18 (12)	CHARACTER	1	DCBDEN	– TAPE DENSITY – 2400 SERIES MAGNETIC TAPE UNITS CODE 7-TRACK 9-TRACK
	 11		DCBMTDN0	″X'03'″– 0 200 BPI –
	.1. . . .11		DCBMTDN1	″X'43'″– 1 556 BPI –
	1. 11		DCBMTDN2	″X'83'″– 2 800 BPI 800 BPI
	11. . . .11		DCBMTDN3	″X'C3'″– 3 –1600 BPI
	11.1 . .11		DCBMTDN4	″X'D3'″– 4 –6250 BPI
19 (13)	BITSTRING	1		– RESERVED
	PAPER TAPE			
8 (8)	ADDRESS	4	DCBLCTBL	– ADDRESS OF TRANSLATE TABLE
12 (C)	BITSTRING	4		– RESERVED
16 (10)	CHARACTER	1	DCBCODE	– PAPER TAPE CODE BEING USED. THE APPROPRIATE TRANSLATE TABLE IS MADE AVAILABLE
	1.		DCBPTCDN	″X'80'″– N - NO CONVERSION
	.1.		DCBPTCDI	″X'40'″– I – IBM BCD
	. .1.		DCBPTCDF	″X'20'″– F – FRIDEN
	. . .1		DCBPTCDB	″X'10'″– B – BURROUGHS
	. . . . 1. . .		DCBPTCDC	″X'08'″– C – NATIONAL CASH REGISTER
	. . . . .1. .		DCBPTCDA	″X'04'″– A – ASCII (8-TRACK)
	. . . . . .1.		DCBPTCDT	″X'02'″– T – TELETYPE
17 (11)	CHARACTER	1	DCBDEVT	– DEVICE TYPE
	.1.1		DCBDVPTP	″X'50'″– 2671 PAPER TAPE READER
18 (12)	BITSTRING	1		– RESERVED
19 (13)	BITSTRING	1	DCBPTFLG	– PAPER TAPE FLAGS
	. . .1		DCBPTIC	″DCBBIT3″– INVALID CHARACTER IN LAST RECORD READ
	. . . . 1. . .		DCBPTECT	″DCBBIT4″– END OF RECORD CHARACTER REACHED IN TRANSLATION
	. . . . .1. .		DCBPTECR	″DCBBIT5″– END OF RECORD CHARACTER DETECTED DURING READ
	. . . . . .1.		DCBPTUCT	″DCBBIT6″– IF ONE, UPPER CASE TRANSLATE. IF ZERO, LOWER CASE TRANSLATE
	1		DCBPTERR	″DCBBIT7″– ERROR DETECTED ON READ
	PRINTER			
16 (10)	CHARACTER	1	DCBPRTSP	– NUMBER INDICATING NORMAL PRINTER SPACING
	1		DCBPRSP0	″X'01'″– 0 – NO SPACING
	. . . . 1. .1		DCBPRSP1	″X'09'″– 1 – SPACE ONE LINE
	. . .1 . . .1		DCBPRSP2	″X'11'″– 2 – SPACE TWO LINES
	. . .1 1. .1		DCBPRSP3	″X'19'″– 3 – SPACE THREE LINES
17 (11)	CHARACTER	1	DCBDEVT	– DEVICE TYPE
	.1. . 1. . .		DCBDVPR1	″X'48'″– 1403 PRINTER AND 1404 PRINTER (CONTINUOUS FORM SUPPORT ONLY)
	.1. . 1.1.		DCBDVPR2	″X'4A'″– 1443 PRINTER
	.1. . 1. .1		DCBDVPR3	″X'49'″– 3211 PRINTER

Offset	Type/Value	Len	Name (Dim)	Description
	.1. . 1.11		DCBDVPR4	″X'4B'″– 3203–4 PRINTER
	.1. . 11.1		DCBDVPRT	″X'4D'″ ANY OTHER PRINTER DEVICE TYPE. IF THIS DEVICE TYPE CODE IS SET, THE UCB DEVICE TYPE FIELD (UCBTBYT4) MUST BE USED TO DETERMINE THE ACTUAL DEVICE TYPE.
	.1. . 111.		DCBDVPR5	″X'4E'″– 3800 PRINTER
18 (12)	CHARACTER	1	DCBPRTOV	– TEST-FOR-PRINTER–OVERFLOW MASK (PRTOV MASK)
	. .1.		DCBPRC9	″X'20'″– 9 – TEST FOR CHANNEL 9 OVERFLOW
	. . .1		DCBPRC12	″X'10'″– 12 – TEST FOR CHANNEL 12 OVERFLOW
19 (13)	CHARACTER	1	DCBPRBYT	PRINTER BYTE
	11		DCBTRCID	″DCBBIT6+DCBBIT7″ 2–BIT ID OF 3800 TRANSLATE TABLE ACTIVE/LAST SELECTED
20 (14)	BITSTRING	1		– RESERVED
	CARD READER, CARD PUNCH			
16 (10)	BITSTRING	1	DCBMODE(0)	– MODE OF OPERATION FOR 1442 CARD READ PUNCH (BITS 0–3)
16 (10)	BITSTRING	1	DCBSTACK	– STACKER SELECTION (BITS 4–7)
	1.		DCBMODEC	″DCBBIT0″– COLUMN BINARY MODE
	.1.		DCBMODEE	″DCBBIT1″– EBCDIC MODE
	. .1.		DCBMODEO	″DCBBIT2″– OPTICAL MARK READ MODE
	. . .1		DCBMODER	″DCBBIT3″– READ COLUMN ELIMINATE MODE
	. . . . . .1.		DCBSTCK2	″DCBBIT6″– STACKER 2
	1		DCBSTCK1	″DCBBIT7″– STACKER 1
17 (11)	CHARACTER	1	DCBDEVT	– DEVICE TYPE
	.1. 1		DCBDVCR0	″X'41'″– 2540 CARD READER
	.1. . . .1.		DCBDVCP0	″X'42'″– 2540 CARD PUNCH
	.1. . . .11		DCBDVCRP	″X'43'″– 1442 CARD READ PUNCH
	.1. . .1. .		DCBDVCR1	″X'44'″– 2501 CARD READER
	.1. . .1.1		DCBDVCPR	″X'45'″– 2520 CARD READ PUNCH
	.1. . .11.		DCBDVCR2	″X'46'″– 3505 CARD READER
	.1. . 11. .		DCBDVCP1	″X'4C'″– 3525 CARD PUNCH
18 (12)	BITSTRING	1		– RESERVED
19 (13)	BITSTRING	1	DCBFUNC	– FUNCTION INDICATOR FOR THE 3525
	1.		DCBFNCBI	″DCBBIT0″– INTERPRET (PUNCH AND PRINT TWO LINES)
	.1.		DCBFNCBR	″DCBBIT1″– READ
	. .1.		DCBFNCBP	″DCBBIT2″– PUNCH
	. . .1		DCBFNCBW	″DCBBIT3″– PRINT
	. . . . 1. . .		DCBFNCBD	″DCBBIT4″– DATA PROTECTION
	. . . . .1. .		DCBFNCBX	″DCBBIT5″– THIS DATA SET IS TO BE PRINTED
	. . . . . .1.		DCBFNCBT	″DCBBIT6″– TWO–LINE PRINT SUPPORT REQUEST
	OPTICAL READER AND MAGNETIC CHAR READER			
0 (0)	ADDRESS	4	DCBWTOID(0)	–SAME AS DCBWTOIA BELOW
0 (0)	BITSTRING	1		– RESERVED
1 (1)	ADDRESS	3	DCBWTOIA	– A BINARY IDENTIFICATION NUMBER ASSIGNED BY COMMUNICATIONS TASK TO MESSAGE ISSUED BY WTO MACRO. THIS NUMBER IS USED BY THE DOM MACRO WHEN MESSAGE IS NO LONGER REQUIRED (MCS SUPPORT). – – –FOR MAGNETIC CHAR READER – AFTER FIRST READ HAS BEEN ISSUED, CONTAINS ADDRESS OF MAGNETIC INTERRUPT CONTROL BLOCK (MICB) BEING USED BY THE APPENDAGES.

Offset	Type/Value	Len	Name (Dim)	Description
	OPTICAL READER DEVICES 1285, 1287, 1288, 3886			
4 (4)	ADDRESS	4	DCBERRCN(0)	– SAME AS DCBERRCA BELOW
4 (4)	BITSTRING	1		– RESERVED
5 (5)	ADDRESS	3	DCBERRCA	– ADDRESS OF 32 BYTES OF DECLARED STORAGE SPECIFIED BY THE USER IN HIS PROGRAM. THIS STORAGE WILL BE USED BY THE PROGRAMMING SUPPORT AS EIGHT 4–BYTE COUNTERS IN WHICH TOTALS OF CERTAIN 1285, 1287 AND 1288 ERROR CONDITIONS ARE ACCUMULATED.
4 (4)	BITSTRING	1	DCBLNNUM	– 3886 DOCUMENT LINE NUMBER
5 (5)	BITSTRING	1	DCBLFMAT	– 3886 LINE FORMAT NUMBER
6 (6)	BITSTRING	1	DCBORFLG	– 3886 FLAGS
	1.		DCBEOPFG	″DCBBIT0″– END OF PAGE
7 (7)	BITSTRING	1		– RESERVED
8 (8)	ADDRESS	4	DCBDSPLY(0)	– SAME AS DCBDSPLA BELOW
8 (8)	CHARACTER	4	DCBFRID(0)	– 3886 FORMAT RECORD ID
8 (8)	BITSTRING	1		– RESERVED
9 (9)	ADDRESS	3	DCBDSPLA	– ADDRESS OF DSPLY (BSAM) ROUTINE USED FOR KEYBOARD ENTRY OF A COMPLETE FIELD
12 (C)	ADDRESS	4	DCBRESCN(0)	– SAME AS DCBRESCA BELOW
12 (C)	ADDRESS	4	DCBRDLNE(0)	– SAME AS DCBRDLNA BELOW
12 (C)	ADDRESS	4	DCBFRTBA(0)	– 3886 FORMAT RECORD TABLE
12 (C)	BITSTRING	1		– RESERVED
13 (D)	ADDRESS	3	DCBRESCA(0)	– ADDRESS OF RESCN (BSAM) ROUTINE USED TO FORCE ON–LINE CORRECTION OF UNREADABLE CHARACTERS
13 (D)	ADDRESS	3	DCBRDLNA	– ADDRESS OF RDLNE (QSAM) ROUTINE USED TO FORCE ON–LINE CORRECTION OF UNREADABLE CHARACTERS
16 (10)	BITSTRING	1	DCBORBYT	– OPTICAL READER BYTE USED BY BSAM/QSAM
	1.		DCBORSYN	″DCBBIT0″– SYNAD IN CONTROL
	.1.		DCBOREOF	″DCBBIT1″– END OF FILE (EOF)
	. .1.		DCBORBFP	″DCBBIT2″– BUFFERS PRIMED (QSAM)
17 (11)	CHARACTER	1	DCBDEVT	– DEVICE TYPE
	.1.1 1.1.		DCBDVOR5	″X'5A'″– 1285 OPTICAL READER
	.1.1 1.11		DCBDVOR7	″X'5B'″– 1287 OPTICAL READER
	.1.1 11. .		DCBDVOR8	″X'5C'″– 1288 OPTICAL READER
	.1.1 .111		DCBDVOR9	″X'57'″– 3886 OPTICAL READER
18 (12)	BITSTRING	1	DCBEIB	– ERROR INDICATOR BYTE
	.1.		DCBORNRM	″DCBBIT1″– THE 1287 OR 1288 SCANNER WAS UNABLE TO LOCATE THE REFERENCE MARK
	. .1.		DCBORREJ	″DCBBIT2″– FOR 1287, A STACKER SELECT COMMAND WAS GIVEN AFTER ALLOTTED TIME HAD ELAPSED AND THE DOCUMENT HAS BEEN PUT IN REJECT POCKET. FOR 1288 UNFORMATTED ONLY, END–OF–PAGE HAS OCCURRED.
	. . .1		DCBORERR	″DCBBIT3″– A NONRECOVERABLE ERROR HAS OCCURRED.
	. . . . 1. . .		DCBORECK	″DCBBIT4″– AN EQUIPMENT CHECK RESULTED IN AN INCOMPLETE READ
	. . . . .1. .		DCBORWLR	″DCBBIT5″– A WRONG–LENGTH RECORD CONDITION HAS OCCURRED

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . . .1.		DCBORHPR	″DCBBIT6″– FOR QSAM –OPERATOR ENTERED ONE OR MORE CHARACTERS FROM THE KEYBOARD. FOR BSAM – A HOPPER EMPTY CONDITION HAS OCCURRED
	1		DCBORDCK	″DCBBIT7″– A DATA CHECK HAS OCCURRED
19 (13)	BITSTRING	1		– RESERVED
	MAGNETIC CHARACTER READER DEVICES 1419 MAGNETIC CHARACTER READER 1275 OPTICAL READER SORTER 3890™ MAGNETIC CHARACTER READER 3895 DOCUMENT READER/INSCRIBER			
0 (0)	CHARACTER	8	DCBSSID	– BEFORE DCB IS OPENED – NAME OF USER'S STACKER SELECT ROUTINE.
0 (0)	ADDRESS	4	(0)	– AFTER DCB IS OPENED – DCBWTOID
0 (0)	ADDRESS	4	DCBQSMEX	– ADDRESS OF QSAM EXIT ROUTINE
4 (4)	ADDRESS	4	DCBSSAD(0)	– ADDRESS OF USER'S STACKER SELECT ROUTINE
4 (4)	ADDRESS	4	DCBIMG(0)	– 3890 ADDR OF USER'S IMAGE PROC RTN
4 (4)	BITSTRING	1		– RESERVED
5 (5)	ADDRESS	3	DCBSSADA	– ADDRESS OF USER'S STACKER SELECT ROUTINE
8 (8)	ADDRESS	4	DCBIMAGE(0)	– SAME AS DCBIMAGA BELOW
8 (8)	BITSTRING	1	DCBMRFG	– BUFFER INDICATOR
	11.		DCBMRBCT	″DCBBIT0+DCBBIT1″– TWO–BIT BINARY COUNTER WHICH INDICATES INTO WHICH BUFFER STATUS INFORMATION IS TO BE POSTED
9 (9)	ADDRESS	3	DCBIMAGA	– ADDRESS OF PARAMETER LIST USED TO COMMUNICATE BETWEEN USER'S PROCESSING ROUTINES AND HIS STACKER SELECT ROUTINES
12 (C)	ADDRESS	4	DCBECBLT(0)	– SAME AS DCBECBLA BELOW
12 (C)	ADDRESS	4	DCBHDR(0)	– 3890 ADDR OF USER'S HEADER DATA AREA
12 (C)	BITSTRING	1	DCBMRIND	– INDICATOR AND COUNTER BYTE
	111.		DCBMRDCT	″DCBBIT0+DCBBIT1+DCBBIT2″ THREE–BIT BINARY COUNTER OF NUMBER OF DOCUMENTS READ AFTER DISENGAGE
	. . .1		DCBMRSCU	″DCBBIT3″– DCB WAS ALTERED WHEN SYNAD ROUTINE WAS ENTERED DUE TO SECONDARY CONTROL UNIT (SCU) ERROR
	. . . . 1. . .		DCBMRPLO	″DCBBIT4″– POCKET LIGHT HAS BEEN TURNED ON
	. . . . .1. .		DCBMRPLS	″DCBBIT5″– POCKET LIGHT 0–6 IS BEING SET ON
	. . . . . .1.		DCBMRERP	″DCBBIT6″–ERROR RECOVERY PROCEDURE IS EXECUTING FOR PRIMARY CONTROL UNIT (PCU)
	1		DCBMRERS	″DCBBIT7″– ERROR RECOVERY PROCEDURE IS EXECUTING FOR SECONDARY CONTROL UNIT (SCU)
13 (D)	ADDRESS	3	DCBECBLA	– ADDRESS OF ECB LIST PASSED TO WAIT MACRO BY CHECK MACRO WHEN NO 1419/1275 IS AVAILABLE FOR PROCESSING
16 (10)	BITSTRING	1	DCBMRFLG	– FLAG BYTE
	1.		DCBMRSCC	″DCBBIT0″– FIRST OR SECOND SECONDARY CONTROL UNIT COMMAND CHAIN IS BEING USED
	.1.		DCBMRDBG	″DCBBIT1″– DEBUGGING MODE IN USE
	. .1.		DCBMRDRU	″DCBBIT2″– DISENGAGE REQUESTED BY USER
	. . .1		DCBMRDR	″DCBBIT3″– DISENGAGE REQUESTED

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . 11. .		DCBMRPCC	″DCBBIT4+DCBBIT5″– TWO–BIT BINARY COUNTER INDICATING FIRST, SECOND, OR THIRD PRIMARY CONTROL UNIT COMMAND CHAIN IS BEING USED
	. . . . . .1.		DCBMRDWT	″DCBBIT6″– WTO MESSAGE MUST BE DELETED
	1		DCBMRUE	″DCBBIT7″– UNIT EXCEPTION
17 (11)	CHARACTER	1	DCBDEVT	– DEVICE TYPE
	.1.1 11.1		DCBDVMR	″X'5D'″– 1419 MAGNETIC CHARACTER READER
	.1.1 1111		DCBDVORS	″X'5F'″– 1275 OPTICAL READER SORTER
	.1.1 .11.		DCBDVMRS	″X'56'″– 3890 MAGNETIC CHARACTER READER SORTER
	.1.1 1. .1		DCBDVDRI	″X'59'″– 3895 DOCUMENT READER/INSCRIBER
18 (12)	CHARACTER	1	DCBAPPIN	– AN INDICATOR USED BY THE APPENDAGES TO PASS INFORMATION ABOUT ONE CHANNEL CHAIN TO AN APPENDAGE ASSOCIATED WITH ANOTHER CHANNEL CHAIN
19 (13)	BITSTRING	1		– RESERVED
	ACCESS METHOD COMMON INTERFACE			
16 (10)	SIGNED	4	DCBRELB(0)	– SAME AS DCBREL BELOW
16 (10)	SIGNED	1	DCBKEYLE	– KEY LENGTH OF DATA SET
17 (11)	CHARACTER	1	DCBDEVT(0)	– DEVICE TYPE
	.1. . 1111		DCBDVTRM	″X'4F'″– TERMINAL. (DD CONTAINS TERM=TS)
17 (11)	SIGNED	3	DCBREL	– NUMBER OF RELATIVE TRACKS OR BLOCKS IN THIS DATA SET (BDAM)
20 (14)	ADDRESS	4	DCBBUFCB(0)	– ADDRESS OF BUFFER POOL CONTROL BLOCK
20 (14)	SIGNED	1	DCBBUFNO	– NUMBER OF BUFFERS REQUIRED FOR THIS DATA SET. MAY RANGE FROM 0 TO 255. IF UNBLOCKED SPANNED RECORDS ARE USED, NUMBER OF SEGMENT WORK AREAS REQUIRED FOR THIS DATA SET.
21 (15)	ADDRESS	3	DCBBUFCA	– ADDRESS OF BUFFER POOL CONTROL BLOCK
24 (18)	SIGNED	2	DCBBUFL	– LENGTH OF BUFFER. MAY RANGE FROM 0 TO 32,767.
26 (1A)	BITSTRING	2	DCBDSORG(0)	– DATA SET ORGANIZATION BEING USED
26 (1A)	BITSTRING	1	DCBDSRG1	– FIRST BYTE OF DCBDSORG
	1.		DCBDSGIS	″DCBBIT0″– IS – INDEXED SEQUENTIAL ORGANIZATION
	.1.		DCBDSGPS	″DCBBIT1″– PS – PHYSICAL SEQUENTIAL ORGANIZATION
	. .1.		DCBDSGDA	″DCBBIT2″– DA – DIRECT ORGANIZATION
	. . .1		DCBDSGCX	″DCBBIT3″– CX – BTAM OR QTAM LINE GROUP
	. . . . . .1.		DCBDSGPO	″DCBBIT6″– PO – PARTITIONED ORGANIZATION
	1		DCBDSGU	″DCBBIT7″– U – UNMOVABLE, THE DATA CONTAINS LOCATION DEPENDENT INFORMATION
27 (1B)	BITSTRING	1	DCBDSRG2	– SECOND BYTE OF DCBDSORG
	1.		DCBDSGGS	″DCBBIT0″– GS – GRAPHICS ORGANIZATION
	.1.		DCBDSGTX	″DCBBIT1″– TX – TCAM LINE GROUP
	. .1.		DCBDSGTQ	″DCBBIT2″ – TQ – TCAM MESSAGE QUEUE
	. . . . 1. . .		DCBACBM	″DCBBIT4″– ACCESS METHOD CONTROL BLOCK
	. . . . .1. .		DCBDSGTR	″DCBBIT5″– TR – TCAM 3705
28 (1C)	ADDRESS	4	DCBIOBAD(0)	– ADDRESS OF IOB WHEN CHAINED SCHEDULING IS USED OR FOR 1419/1275
28 (1C)	ADDRESS	4	DCBICQE(0)	– ADDRESS OF ICQE
28 (1C)	ADDRESS	4	DCBODEB(0)	– ADDRESS OF OLD DEB
28 (1C)	SIGNED	1	DCBLNP(0)	– 3525 PRINTER LINE POSITION COUNTER

Offset	Type/Value	Len	Name (Dim)	Description
28 (1C)	BITSTRING	1	DCBQSLM	– QSAM LOCATE MODE LOGICAL RECORD INTERFACE INDICATOR BYTE FOR UPDAT PROCESSING OF SPANNED RECORDS
	1.		DCB1DVDS	″DCBBIT0″– ONLY ONE DEVICE IS ALLOCATED TO THIS DATA SET
	.1.		DCBUPDCM	″DCBBIT1″– UPDATE COMPLETE, FREE OLD DEB
	. .11 . . .		DCBUPDBT	″DCBBIT2+DCBBIT3″– UPDATE BITS
	. .1.		DCBUPDT	″DCBBIT2″– UPDATE TO TAKE PLACE
	. .11		DCBNUPD	″DCBBIT2+DCBBIT3″– NO UPDATE TO TAKE PLACE
	. . .1		DCBSVDEB	″DCBBIT3″– OLD DEB ADDRESS MUST BE SAVED
29 (1D)	ADDRESS	3	DCBIOBAA(0)	– SAME AS DCBIOBAD ABOVE
29 (1D)	ADDRESS	3	DCBICQEA(0)	– SAME AS DCBICQE ABOVE
29 (1D)	ADDRESS	3	DCBODEBA	– ADDRESS OF OLD DEB
28 (1C)	ADDRESS	4	DCBSVCXL(0)	– SAME AS DCBSVCXA BELOW
28 (1C)	BITSTRING	1		– RESERVED
29 (1D)	ADDRESS	3	DCBSVCXA	– POINTER TO EXIT LIST OF JES C.I. INTERFACE CONTROL SVC
	FOUNDATION EXTENSION			
32 (20)	ADDRESS	4	DCBEODAD(0)	– SAME AS DCBEODA BELOW
32 (20)	BITSTRING	1	DCBHIARC(0)	– HIERARCHY BITS
32 (20)	BITSTRING	1	DCBBFTEK(0)	– BUFFERING TECHNIQUE BITS
32 (20)	BITSTRING	1	DCBBFALN	– BUFFER ALIGNMENT BITS
	1.		DCBH1	″DCBBIT0″– HIERARCHY 1 MAIN STORAGE IF BIT 5 IS ZERO. DCBE EXISTS IF BIT 5 IS ON. INTERPRETATION OF BITS DCBH1 AND DCBH0:
				DCBH1 DCBH0
				0 0 NO HIARCHY, NO DCBE.
				1 0 HIARCHY=1, NO DCBE.
				0 1 HIARCHY=0, NO DCBE.
				1 1 NO HIARCHY, DCBE EXISTS, POINTED TO BY DCBDCBE
	.111		DCBBFT	″DCBBIT1+DCBBIT2+DCBBIT3″ BUFFERING TECHNIQUE
	.11.		DCBBFTA	″DCBBIT1+DCBBIT2″– QSAM LOCATE MODE PROCESSING OF SPANNED RECORDS – OPEN IS TO CONSTRUCT A RECORD AREA IF IT AUTOMATICALLY CONSTRUCTS BUFFERS
	. .1.		DCBBFTR	″DCBBIT2″– FOR BSAM CREATE BDAM PROCESSING OF UNBLOCKED SPANNED RECORDS – SOFTWARE TRACK OVERFLOW. FOR BSAM INPUT PROCESSING OF UNBLOCKED SPANNED RECORDS WITH KEYS – RECORD OFFSET PROCESSING.
	.1.		DCBBFTS	″DCBBIT1″– SIMPLE BUFFERING –BIT 3 IS ZERO
	. .1.		DCBBFTKR	″DCBBIT2″– UNBLOCKED SPANNED RECORDS – SOFTWARE TRACK OVERFLOW (BDAM)
	. . .1		DCBBFTE	″DCBBIT3″– EXCHANGE BUFFERING– BIT 1 IS ZERO
	. . . . 1. . .		DCBBFTKD	″DCBBIT4″– DYNAMIC BUFFERING (BTAM)
	. . . . 1. . .		DCBBFTK	″DCBBIT4″– LRECL IN ’K’ UNITS FOR XLRI

Offset	Type/Value	Len	Name (Dim)	Description
	.11. 1. . .		DCBBXLRI	″DCBBIT1+DCBBIT2+DCBBIT4″ – EXTENDED LOGICAL RECORD INTERFACE MODE FOR SPANNED RECORDS GREATER THAN 32K – ALL RECORDS TRANSFERRED TO OR FROM THE RECORD AREA.
	. . . . .1. .		DCBH0	″DCBBIT5″– HIERARCHY 0 MAIN STORAGE IF BIT 0 IS ZERO. DCBE EXISTS IF BIT 0 IS ON. SEE DCBH1
	11		DCBBFA	″DCBBIT6+DCBBIT7″– BUFFER ALIGNMENT
	. . . . . .1.		DCBBFAD	″DCBBIT6″– DOUBLEWORD BOUNDARY
	1		DCBBFAF1	″DCBBIT7″– FULLWORD NOT A DOUBLEWORD BOUNDARY, CODED IN DCB MACRO INSTRUCTION
	11		DCBBFAF2	″DCBBIT6+DCBBIT7″– FULLWORD NOT A DOUBLEWORD BOUNDARY, CODED IN DCB MACRO INSTRUCTION
33 (21)	ADDRESS	3	DCBEODA	– ADDRESS OF A USER– PROVIDED ROUTINE TO HANDLE END-OF-DATA CONDITIONS
36 (24)	ADDRESS	4	DCBEXLST(0)	– ADDRESS OF USER-PROVIDED LIST OF EXITS
36 (24)	BITSTRING	1	DCBRECFM	– RECORD FORMAT
	111.		DCBRECLA	″DCBBIT0+DCBBIT1+DCBBIT2″ RECORD FORMAT INDICATOR – ASCII
	. .1.		DCBRECD	″DCBBIT2″– ASCII VARIABLE FORMAT
	11.		DCBRECL	″DCBBIT0+DCBBIT1″– RECORD FORMAT INDICATOR
	1.		DCBRECF	″DCBBIT0″– FIXED RECORD FORMAT
	.1.		DCBRECV	″DCBBIT1″– VARIABLE RECORD FORMAT
	11.		DCBRECU	″DCBBIT0+DCBBIT1″– UNDEFINED RECORD FORMAT
	. .1.		DCBRECTO	″DCBBIT2″– TRACK OVERFLOW
	. . .1		DCBRECBR	″DCBBIT3″– BLOCKED RECORDS
	. . . . 1. . .		DCBRECSB	″DCBBIT4″– FOR FIXED LENGTH RECORD FORMAT –STANDARD BLOCKS. FOR VARIABLE LENGTH RECORD FORMAT – SPANNED RECORDS
	. . . . .11.		DCBRECCC	″DCBBIT5+DCBBIT6″– CONTROL CHARACTER INDICATOR
	. . . . .1. .		DCBRECCA	″DCBBIT5″– ASA CONTROL CHARACTER
	. . . . . .1.		DCBRECCM	″DCBBIT6″– MACHINE CONTROL CHARACTER
	. . . . .		DCBRECC	″X’00’″– NO CONTROL CHARACTER
	1		DCBRECKL	″DCBBIT7″– KEY LENGTH (KEYLEN) WAS SPECIFIED IN DCB MACRO INSTRUCTION
37 (25)	ADDRESS	3	DCBEXLSA	– ADDRESS OF USER– PROVIDED LIST OF EXITS
	FOUNDATION BEFORE OPEN			
40 (28)	CHARACTER	8	DCBDDNAM	– NAME ON THE DD STATEMENT WHICH DEFINES THE DATA SET ASSOCIATED WITH THIS DCB
48 (30)	BITSTRING	1	DCBOFLGS	– FLAGS USED BY OPEN ROUTINE
	1.		DCBOFLWR	″DCBBIT0″– IF ZERO, LAST I/O OPERATION WAS READ OR POINT. IF ONE, LAST I/O OPERATION WAS WRITE.
	1.		DCBOFIOD	″DCBBIT0″– DATA SET IS BEING OPENED FOR INPUT OR OUTPUT (BDAM)
	.1.		DCBOFLRB	″DCBBIT1″– LAST I/O OPERATION WAS IN READ BACKWARD MODE
	. .1.		DCBOFEOV	″DCBBIT2″– SET TO 1 BY EOV WHEN IT CALLS CLOSE ROUTINE FOR CONCATENATION OF DATA SETS WITH UNLIKE ATTRIBUTES
	. . .1		DCBOFOPN	″DCBBIT3″– AN OPEN HAS BEEN SUCCESSFULLY COMPLETED

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . 1. . .		DCBOFPPC	″DCBBIT4″– SET TO 1 BY PROBLEM PROGRAM TO INDICATE A CONCATENATION OF UNLIKE ATTRIBUTES
	. . . . .1. .		DCBOFTM	″DCBBIT5″– TAPE MARK HAS BEEN READ
	. . . . . .1.		DCBOFUEX	″DCBBIT6″– SET TO 0 BY AN I/O SUPPORT FUNCTION WHEN THAT FUNCTION TAKES A USER EXIT. SET TO 1 ON RETURN FROM USER EXIT TO THE I/O SUPPORT FUNCTION WHICH TOOK THE EXIT.
	1		DCBOFIOF	″DCBBIT7″– SET TO 1 BY AN I/O SUPPORT FUNCTION IF DCB IS TO BE PROCESSED BY THAT FUNCTION
49 (31)	BITSTRING	1	DCBIFLG	– FLAGS USED BY IOS IN COMMUNICATING ERROR CONDITIONS AND IN DETERMINING CORRECTIVE PROCEDURES
	11.		DCBIBEC	″DCBBIT0+DCBBIT1″– ERROR CORRECTION INDICATOR
	. . . . .		DCBIFNEP	″X’00’″– NOT IN ERROR PROCEDURE
	.1.		DCBEX	″DCBBIT1″– ERROR CORRECTION OR IOS PAGE FIX IN PROCESS
	11.		DCBIFPEC	″DCBBIT0+DCBBIT1″– PERMANENT ERROR CORRECTION
	. .11		DCBIBPCT	″DCBBIT2+DCBBIT3″– PRINTER CARRIAGE TAPE PUNCH INDICATOR
	. .1.		DCBIFC9	″DCBBIT2″– CHANNEL 9 PRINTER CARRIAGE TAPE PUNCH SENSED
	. . .1		DCBIFC12	″DCBBIT3″–L 12 PRINTER CARRIAGE TAPE PUNCH SENSED
	. . . . 11. .		DCBIBIOE	″DCBBIT4+DCBBIT5″– IOS ERROR ROUTINE USE INDICATOR
	. . . . .		DCBIFER	″X'00'″– ALWAYS USE I/O SUPERVISOR ERROR ROUTINE
	. . . . .1. .		DCBIFNE1	″DCBBIT5″– NEVER USE I/O SUPERVISOR ERROR ROUTINE
	. . . . .1. .		DCBIFTIM	″DCBBIT5″– TEST IOS MASK (IMSK) FOR ERROR PROCEDURE (BTAM)
	. . . . 1. . .		DCBIFNE2	″DCBBIT4″– NEVER USE I/O SUPERVISOR ERROR ROUTINE
	. . . . 11. .		DCBIFNE3	″DCBBIT4+DCBBIT5″– NEVER USE I/O SUPERVISOR ERROR ROUTINE
50 (32)	BITSTRING	2	DCBMACR(0)	– MACRO INSTRUCTION REFERENCE
50 (32)	BITSTRING	1	DCBMACR1	– FIRST BYTE OF DCBMACR
	1.		DCBMRECP	″DCBBIT0″– EXECUTE CHANNEL PROGRAM (EXCP). ALWAYS ZERO (BSAM, QSAM, BPAM, BISAM, QISAM, BDAM). RESERVED (QTAM, BTAM)
	.1.		DCBMRFE	″DCBBIT1″– FOUNDATION EXTENSION IS PRESENT (EXCP)
	.1.		DCBMRGET	″DCBBIT1″– GET (QSAM, QISAM, TCAM)
	.1.		DCBMRPTQ	″DCBBIT1″– PUT FOR MESSAGE GROUP (QTAM) – – – ALWAYS ZERO (BSAM, BPAM, BISAM, BDAM) – – – RESERVED (BTAM)
	. .1.		DCBMRAPG	″DCBBIT2″– APPENDAGES ARE REQUIRED (EXCP)
	. .1.		DCBMRRD	″DCBBIT2″– READ (BSAM, BPAM, BISAM, BDAM, BTAM)
	. .1.		DCBMRWRQ	″DCBBIT2″– WRITE FOR LINE GROUP (QTAM). h. ALWAYS ZERO (QSAM, QISAM)

Offset	Type/Value	Len	Name (Dim)	Description
	. . .1		DCBMRCI	″DCBBIT3″– COMMON INTERFACE (EXCP)
	. . .1		DCBMRMVG	″DCBBIT3″– MOVE MODE OF GET (QSAM, QISAM)
	. . .1		DCBMRRDK	″DCBBIT3″– KEY SEGMENT WITH READ (BDAM) – – – ALWAYS ZERO (BISAM) – – –RESERVED (BSAM, BPAM, QTAM, BTAM)
	. . . . 1. . .		DCBMRLCG	″DCBBIT4″– LOCATE MODE OF GET (QSAM, QISAM)
	. . . . 1. . .		DCBMRRDI	″DCBBIT4″– ID ARGUMENT WITH READ (BDAM) – – – ALWAYS ZERO (BISAM) – – –RESERVED (EXCP, BSAM, BPAM, QTAM, BTAM)
	. . . . .1. .		DCBMRABC	″DCBBIT5″– USER'S PROGRAM MAINTAINS ACCURATE BLOCK COUNT (EXCP)
	. . . . .1 . .		DCBMRPT1	″DCBBIT5″– POINT (WHICH IMPLIES NOTE) (BSAM, BPAM)
	. . . . .1. .		DCBMRSBG	″DCBBIT5″– SUBSTITUTE MODE OF GET (QSAM)
	. . . . .1. .		DCBMRDBF	″DCBBIT5″– DYNAMIC BUFFERING (BISAM, BDAM) – – – ALWAYS ZERO (QISAM) – – –RESERVED (QTAM, BTAM)
	. . . . . .1.		DCBPGFXA	″DCBBIT6″– PAGE FIX APPENDAGE IS SPECIFIED (EXCP)
	. . . . . .1.		DCBMRCRL	″DCBBIT6″– CNTRL (BSAM, QSAM)
	. . . . . .1.		DCBMRCHK	″DCBBIT6″– CHECK (BISAM)
	. . . . . .1.		DCBMRRDX	″DCBBIT6″– READ EXCLUSIVE (BDAM) – – –RESERVED (BPAM, QISAM, QTAM, BTAM)
	1		DCBMRDMG	″DCBBIT7″– DATA MODE OF GET (QSAM)
	1		DCBMRCK	″DCBBIT7″– CHECK (BDAM) – – – RESERVED (EXCP, BSAM, BPAM, BISAM, QISAM, QTAM, BTAM)
51 (33)	BITSTRING	1	DCBMACR2	– SECOND BYTE OF DCBMACR
	1.		DCBMRSTL	″DCBBIT0″– SETL (QISAM) – – –ALWAYS ZERO (BSAM, QSAM, BPAM, BISAM, BDAM) – – –RESERVED (EXCP, QTAM, BTAM)
	.1.		DCBMRPUT	″DCBBIT1″– PUT (QSAM, TCAM) –PUT OR PUTX (QISAM)
	.1.		DCBMRGTQ	″DCBBIT1″– GET FOR MESSAGE GROUP (QTAM) – – –ALWAYS ZERO (BSAM, BPAM, BISAM, BDAM) – – – RESERVED (EXCP, BTAM)
	. .1.		DCBMRWRT	″DCBBIT2″– WRITE (BSAM, BPAM, BISAM, BDAM, BTAM)
	. .1.		DCBMRRDQ	″DCBBIT2″– READ FOR LINE GROUP (QTAM) – – – ALWAYS ZERO (QSAM, QISAM) – – – RESERVED (EXCP)
	. . .1		DCBMRMVP	″DCBBIT3″– MOVE MODE OF PUT (QSAM, QISAM)
	. . .1		DCBMRWRK	″DCBBIT3″– KEY SEGMENT WITH WRITE (BDAM) – – –ALWAYS ZERO (BISAM) – – –RESERVED (EXCP, BSAM, BPAM, QTAM, BTAM)
	. . . . 1. . .		DCBMR5WD	″DCBBIT4″– FIVE–WORD DEVICE INTERFACE (EXCP)
	. . . . 1. . .		DCBMRLDM	″DCBBIT4″– LOAD MODE BSAM (CREATE BDAM DATA SET) (BSAM)
	. . . . 1. . .		DCBMRLCP	″DCBBIT4″– LOCATE MODE OF PUT (QSAM, QISAM)
	. . . . 1. . .		DCBMRIDW	″DCBBIT4″– ID ARGUMENT WITH WRITE (BDAM) – – – ALWAYS ZERO (BISAM) – – – RESERVED (BPAM, QTAM, BTAM)
	. . . . .1. .		DCBMR4WD	″DCBBIT5″– FOUR–WORD DEVICE INTERFACE (EXCP)

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . .1. .		DCBMRPT2	″DCBBIT5″– POINT (WHICH IMPLIES NOTE) (BSAM, BPAM)
	. . . . .1. .		DCBMRTMD	″DCBBIT5″– SUBSTITUTE MODE (QSAM)
	. . . . .1. .		DCBMRUIP	″DCBBIT5″–UPDATE IN PLACE (PUTX) (QISAM) – – – ALWAYS ZERO (BISAM) – – –RESERVED (BDAM, QTAM, BTAM)
	. . . . . .1.		DCBMR3WD	″DCBBIT6″– THREE-WORD DEVICE INTERFACE (EXCP)
	. . . . . .1.		DCBMRCTL	″DCBBIT6″– CNTRL (BSAM, QSAM)
	. . . . . .1.		DCBMRSTK	″DCBBIT6″– SETL BY KEY (QISAM)
	. . . . . .1.		DCBMRAWR	″DCBBIT6″– ADD TYPE OF WRITE (BDAM) – – – ALWAYS ZERO (BISAM) – – – RESERVED (BPAM, QTAM, BTAM)
	1		DCBMR1WD	″DCBBIT7″– ONE–WORD DEVICE INTERFACE (EXCP)
	1		DCBMRSWA	″DCBBIT7″– USER'S PROGRAM HAS PROVIDED A SEGMENT WORK AREA POOL (BSAM CREATE BDAM, BDAM)
	1		DCBMRDMD	″DCBBIT7″– DATA MODE (QSAM)
	1		DCBMRSTI	″DCBBIT7″– SETL BY ID (QISAM) – – – ALWAYS ZERO (BISAM) – – –RESERVED (BPAM, QTAM, BTAM)
	FOUNDATION AFTER OPEN			
40 (28)	ADDRESS	2	DCBTIOT	OFFSET FROM TIOT ORIGIN TO TIOELNGH FIELD IN TIOT ENTRY FOR DD STATEMENT ASSOCIATED WITH THIS DCB
42 (2A)	BITSTRING	2	DCBMACRF(0)	– SAME AS DCBMACR BEFORE OPEN
42 (2A)	BITSTRING	1	DCBMACF1	– FIRST BYTE OF DCBMACRF
43 (2B)	BITSTRING	1	DCBMACF2	– SECOND BYTE OF DCBMACRF
44 (2C)	ADDRESS	4	DCBDEBAD(0)	– ADDRESS OF ASSOCIATED DEB
44 (2C)	BITSTRING	1	DCBIFLGS	– SAME AS DCBIFLG BEFORE OPEN
	11.		DCBIFEC	″DCBBIT0+DCBBIT1″– ERROR CORRECTION INDICATOR
	. .11		DCBIFPCT	″DCBBIT2+DCBBIT3″– PRINTER CARRIAGE TAPE PUNCH INDICATOR
	. . . . 11. .		DCBIFIOE	″DCBBIT4+DCBBIT5″– IOS ERROR ROUTINE USE INDICATOR
	. . . . . .1.		DCBIFLDT	″DCBBIT6″– POSSIBLE LOST DATA CONDITION SUCH AS FOR A PRINTER
45 (2D)	ADDRESS	3	DCBDEBA	– ADDRESS OF ASSOCIATED DEB
48 (30)	ADDRESS	4	DCBREAD(0)	– ADDRESS OF READ MODULE
48 (30)	ADDRESS	4	DCBWRITE(0)	– ADDRESS OF WRITE MODULE
48 (30)	BITSTRING	1	DCBOFLG	SAME AS DCBOFLGS BEFORE OPEN
49 (31)	ADDRESS	3	DCBREADA(0)	ADDRESS OF READ MODULE
49 (31)	ADDRESS	3	DCBWRITA	ADDRESS OF WRITE MODULE
48 (30)	ADDRESS	4	DCBGET(0)	– ADDRESS OF GET MODULE
48 (30)	ADDRESS	4	DCBPUT(0)	– ADDRESS OF PUT MODULE
48 (30)	BITSTRING	1	DCBOFLG1	SAME AS DCBOFLGS BEFORE OPEN
49 (31)	ADDRESS	3	DCBGETA(0)	ADDRESS OF GET MODULE
49 (31)	ADDRESS	3	DCBPUTA	ADDRESS OF PUT MODULE
	EXCP WITH APPENDAGES			
52 (34)	BITSTRING	1	DCBOPTCD	– OPTION CODES
53 (35)	BITSTRING	7		– RESERVED
	EXCP APPENDAGE LIST			
60 (3C)	CHARACTER	2	DCBEOEA	– END OF EXTENT APPENDAGE ID

Offset	Type/Value	Len	Name (Dim)	Description
62 (3E)	CHARACTER	2	DCBPCIA	– PROGRAM CONTROLLED INTERRUPTION APPENDAGE ID
64 (40)	CHARACTER	2	DCBSIOA	– START I/O APPENDAGE ID
66 (42)	CHARACTER	2	DCBCENDA	– CHANNEL END APPENDAGE ID
68 (44)	CHARACTER	2	DCBXENDA	– ABNORMAL END APPENDAGE ID
70 (46)	BITSTRING	2		– RESERVED
	QSAM-BSAM-BPAM	COMMON	INTERFACE	
52 (34)	ADDRESS	4	DCBGERR(0)	– ADDRESS OF SYNCHRONIZING ROUTINE FOR GET
52 (34)	ADDRESS	4	DCBPERR(0)	– ADDRESS OF SYNCHRONIZING ROUTINE FOR PUT
52 (34)	ADDRESS	4	DCBCHECK(0)	– ADDRESS OF CHECK MODULE
52 (34)	BITSTRING	1	DCBOPTCD	– OPTION CODES
	1.		DCBOPTW	″DCBBIT0″– WRITE VALIDITY CHECK (DASD) (BSAM, BPAM, QSAM, ISAM, BDAM)
	.1.		DCBOPTU	″DCBBIT1″– ALLOW DATA CHECK CAUSED BY INVALID CHARACTER (PRINTER WITH UCS FEATURE) (BSAM, BPAM, QSAM) MSS WINDOW PROCESSING REQUESTED (BSAM, QSAM)
	. .1.		DCBOPTC	″DCBBIT2″– CHAINED SCHEDULING (BSAM, BPAM, QSAM)
	. . .1		DCBOPTH	″DCBBIT3″– 1287/1288 OPTICAL READER – HOPPER EMPTY EXIT (BSAM, BPAM)
	. . .1		DCBOPTO	″DCBBIT3″– 1285/1287 OPTICAL READER – ONLINE CORRECTION (QSAM)
	. . .1		DCBBCKPT	″DCBBIT3″– CHANNEL–END APPENDAGE IS TO BYPASS DOS EMBEDDED CHECKPOINT RECORDS ON TAPE (BSAM, QSAM)
	. . . . 1. . .		DCBOPTQ	″DCBBIT4″– TRANSLATION TO OR FROM ASCII (BSAM, BPAM, QSAM)
	. . . . .1. .		DCBOPTZ	″DCBBIT5″– MAGNETIC TAPE DEVICES – USE REDUCED ERROR RECOVERY PROCEDURE (EXCP, BSAM, BPAM, QSAM)
	. . . . .1. .		DCBSRCHD	″DCBBIT5″– USE SEARCH DIRECT, INSTEAD OF SEARCH PREVIOUS, ON RECORD POSITION SENSING DEVICE (EXCP, BSAM, BPAM, QSAM)
	. . . . . .1.		DCBOPTT	″DCBBIT6″– USER TOTALING (BSAM, QSAM)
	1		DCBOPTJ	″DCBBIT7″ 3800 PRINTER, OPTCD=J; (DYNAMIC SELECT OF TRANSLATE TABLES)
53 (35)	ADDRESS	3	DCBGERRA(0)	– ADDRESS OF SYNCHRONIZING ROUTINE FOR GET
53 (35)	ADDRESS	3	DCBPERRA(0)	– ADDRESS OF SYNCHRONIZING ROUTINE FOR PUT
53 (35)	ADDRESS	3	DCBCHCKA	– ADDRESS OF CHECK MODULE
56 (38)	ADDRESS	4	DCBSYNAD(0)	– ADDRESS OF USER–PROVIDED SYNAD ROUTINE
56 (38)	SIGNED	1	DCBIOBL	– IOB LENGTH IN DOUBLEWORDS
57 (39)	ADDRESS	3	DCBSYNA	– ADDRESS OF USER–PROVIDED SYNAD ROUTINE
60 (3C)	BITSTRING	1	DCBFLAG1(0)	– TCAM APPLICATION PROGRAM FLAGS (BSAM, BPAM, QSAM)
60 (3C)	BITSTRING	1	DCBCIND1	– CONDITION INDICATORS
	1.		DCBCNTOV	″DCBBIT0″– DIRECT ACCESS – TRACK OVERFLOW IN USE (BSAM, BPAM, QSAM) 2540 CARD PUNCH – DATA SET WAS OPENED BUT NO DATA WAS WRITTEN (QSAM)

Offset	Type/Value	Len	Name (Dim)	Description
	1.		DCBSTQCK	″DCBBIT0″– STOP EQUAL QUICK WAS SPECIFIED FOR APPLICATION PROG. DCBS (TCAM)
	.1.		DCBSTFLS	″DCBBIT1″– STOP EQUAL FLUSH WAS SPECIFIED FOR APPLICATION PROG. DCBS (TCAM)
	.1.		DCBCNSRD	″DCBBIT1″– SEARCH DIRECT (BSAM, BPAM, QSAM)
	. .1.		DCBCNEVB	″DCBBIT2″– END OF VOLUME –USED BY EOB ROUTINES (BSAM, BPAM, QSAM)
	. . .1		DCBCNEVA	″DCBBIT3″– END OF VOLUME –USED BY CHANNEL– END APPENDAGE ROUTINES (BSAM, BPAM, QSAM)
	. . . . 1. . .		DCBCNCI	″DCBBIT4″– SAM–SI COMPATIBILITY INTERFACE (CI) PROCESSING HAS BEEN PERFORMED (BSAM, QSAM)
	. . . . .1. .		DCBCNBRM	″DCBBIT5″– BLOCKED RECORD BIT MODIFIED (BSAM, BPAM, QSAM)
	. . . . . .1.		DCBCBNDF	OPEN DEFAULTED BUFNO (QSAM)
	1		DCBCNEXB	″DCBBIT7″– EXCHANGE BUFFERING SUPPORTED (QSAM)
61 (3D)	BITSTRING	1	DCBCIND2	– CONDITION INDICATORS
	1.		DCBCNSTO	″DCBBIT0″– PARTITIONED DATA SET – STOW HAS BEEN PERFORMED (BSAM, BPAM, QSAM) SEQUENTIAL DATA SET – UPDATE (BSAM, BPAM)
	.1.		DCBCNWR0	″DCBBIT1″– DIRECT ORGANIZATION DATA SET – LAST I/O WAS A WRITE RECORD ZERO (BSAM, BPAM, QSAM) SEQUENTIAL DATA SET – UPDATE EOF IS INDICATED (BSAM, BPAM)
	. .1.		DCBCNCLO	″DCBBIT2″– CLOSE IN PROCESS (QSAM)
	. . .1		DCBCNIOE	″DCBBIT3″– PERMANENT I/O ERROR (BSAM, BPAM, QSAM)
	. . . . 1. . .		DCBCNBFP	″DCBBIT4″– OPEN ACQUIRED BUFFER POOL (BSAM, BPAM, QSAM)
	. . . . .1. .		DCBCNCHS	″DCBBIT5″– CHAINED SCHEDULING BEING SUPPORTED (BSAM, BPAM, QSAM)
	. . . . . .1.		DCBCNFEO	″DCBBIT6″– FEOV BIT (BSAM, BPAM, QSAM)
	1		DCBCNQSM	″DCBBIT7″– ALWAYS ZERO (BSAM, BPAM) THIS IS A QSAM DCB (QSAM)
62 (3E)	SIGNED	2	DCBBLKSI	– MAXIMUM BLOCK SIZE
64 (40)	ADDRESS	1	DCBWCPO	– OFFSET OF WRITE CHANNEL PROGRAM FROM THE START OF IOB
65 (41)	SIGNED	1	DCBWCPL	– LENGTH OF WRITE CHANNEL PROGRAM
66 (42)	ADDRESS	1	DCBOFFSR	– OFFSET OF READ CCW FROM BSAM/BPAM PREFIX OF IOB
67 (43)	ADDRESS	1	DCBOFFSW	– OFFSET OF WRITE CCW FROM BSAM/BPAM PREFIX OF IOB
68 (44)	ADDRESS	4	DCBIOBA	– FOR NORMAL SCHEDULING, ADDRESS OF QSAM OR BSAM/BPAM PREFIX OF IOB. FOR CHAINED SCHEDULING, ADDRESS OF ICB. FOR 1419/1275, ADDRESS OF MAGNETIC INTERRUPT CONTROL BLOCK (MICB) CURRENTLY BEING PROCESSED BY READ ROUTINE. FOR TSO/E TERMINAL DATA SET OPENED FOR INPUT AND FORMAT U, SIMULATED LOW– ORDER FOUR BYTES OF IOBCSW.
68 (44)	BITSTRING	1		– DCBNCP (BSAM, BPAM)
69 (45)	ADDRESS	3	DCBIOBB	– SAME AS DCBIOBA ABOVE

Offset	Type/Value	Len	Name (Dim)	Description
68 (44)	ADDRESS	4	DCBCICB(0)	– SAME AS DCBCICBA BELOW
68 (44)	BITSTRING	1		– DCBNCP (BSAM, BPAM)
69 (45)	ADDRESS	3	DCBCICBA	– POINTER TO JES C.I. CONTROL BLOCK (CICB)
80 (50)	SIGNED	2	DCBDIRCT(0)	– NUMBER OF BYTES USED IN LAST DIRECTORY BLOCK (RANGE 0–254) (BSAM, BPAM)
80 (50)	BITSTRING	1	DCBQSWS(0)	– FLAG BYTE
	. . . . .1. .		DCBPOPEN	″DCBBIT5″– QSAM PARALLEL INPUT PROCESSING
80 (50)	BITSTRING	1	DCBUSASI	– FLAG BYTE FOR ASCII TAPES
	.1.		DCBBLBP	″DCBBIT1″– BLOCK PREFIX IS FOUR BYTE FIELD CONTAINING BLOCK LENGTH IN UNPACKED DECIMAL (SPECIFIED BY BUFFER=L).
	. .11 1. . .		DCBQADFS	″DCBBIT2+DCBBIT3+DCBBIT4″USED TO PERFORM SEQUENCE CHECKING WITH MULTIPLE FUNCTION SUPPORT FOR 3525 (BSAM, QSAM)
	. .1.		DCBQADF1	″DCBBIT2″– FIRST BIT OF DCBQADFS
	. . .1		DCBQADF2	″DCBBIT3″– SECOND BIT OF DCBQADFS
	. . . . 1. . .		DCBQADF3	″DCBBIT4″– THIRD BIT OF DCBQADFS
	. . . . . .1.		DCB3525A	″DCBBIT6″– DCB IS 3525 – ASSOCIATED DATA SETS EXIST
	1		DCBQSTRU	″DCBBIT7″– TRUNC ENTRY POINT ENTERED (QSAM)
81 (51)	SIGNED	1	DCBBUFOF(0)	– BLOCK PREFIX LENGTH (0–99), SPECIFIED BY BUFOFF=N OR BUFOFF=L
81 (51)	SIGNED	1	DCBDIRCQ	– NUMBER OF BYTES USED IN LAST DIRECTORY BLOCK (RANGE 0–254) (QSAM)
	BSAM–BPAM INTERFACE			
72 (48)	ADDRESS	4	DCBEOBR(0)	– ADDRESS OF END-OF-BLOCK MODULE FOR READ
72 (48)	SIGNED	1	DCBNCP	– NUMBER OF CHANNEL PROGRAMS. NUMBER OF READ OR WRITE REQUESTS WHICH MAY BE ISSUED PRIOR TO A CHECK, NUMBER OF IOB'S GENERATED. (99 MAXIMUM)
73 (49)	ADDRESS	3	DCBEOBRA	– ADDRESS OF END-OF-BLOCK MODULE FOR READ
76 (4C)	ADDRESS	4	DCBEOBW	– ADDRESS OF END-OF-BLOCK MODULE FOR WRITE. FOR BSAM CREATE BDAM PROCESSING OF UNBLOCKED SPANNED RECORDS WITH BKTEK=R SPECIFIED, ADDRESS OF SEGMENT WORK AREA CONTROL BLOCK.
80 (50)	SIGNED	2	DCBDIRCT	–NUMBER OF BYTES USED IN LAST DIRECTORY BLOCK (RANGE 0–254)
82 (52)	SIGNED	2	DCBLRECL	– LOGICAL RECORD LENGTH
84 (54)	ADDRESS	4	DCBCNTRL(0)	– ADDRESS OF CNTRL MODULE
84 (54)	ADDRESS	4	DCBNOTE(0)	– ADDRESS OF NOTE/POINT MODULE
84 (54)	ADDRESS	4	DCBPOINT	– ADDRESS OF NOTE/POINT MODULE
	QSAM INTERFACE			
72 (48)	ADDRESS	4	DCBLCCW(0)	– FOR EXCHANGE BUFFERING, ADDRESS OF LAST CCW IN LIST
72 (48)	ADDRESS	4	DCBEOBAD	– FOR SIMPLE BUFFERING, ADDRESS OF LAST BYTE OF CURRENT BUFFER
76 (4C)	ADDRESS	4	DCBCCCW(0)	– FOR EXCHANGE BUFFERING, ADDRESS OF CURRENT OR NEXT CCW
76 (4C)	ADDRESS	4	DCBRECAD(0)	– ADDRESS OF CURRENT OR NEXT LOGICAL RECORD
77 (4D)	ADDRESS	3	DCBRECA	– ADDRESS OF CURRENT OR NEXT LOGICAL RECORD

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Offset	Type/Value	Len	Name (Dim)	Description
80 (50)	BITSTRING	1	DCBQSWS	– FLAG BYTE
81 (51)	SIGNED	1	DCBDIRCQ	– NUMBER OF BYTES USED IN LAST DIRECTORY BLOCK (RANGE 0–254)
82 (52)	SIGNED	2	DCBLRECL	– LOGICAL RECORD LENGTH
84 (54)	ADDRESS	4	DCBCNTRL(0)	– ADDRESS OF CNTRL MODULE
84 (54)	BITSTRING	1	DCBEROPT	– ERROR OPTION
	1.		DCBERACC	″DCBBIT0″– ACCEPT PERMANENT ERROR
	.1.		DCBERSKP	″DCBBIT1″– SKIP PERMANENT ERROR
	. .1.		DCBERABE	″DCBBIT2″– ABNORMAL END OF TASK
	1		DCBERBAM	″DCBBIT7″– OPEN TURNED OFF BFTEK=A
85 (55)	ADDRESS	3	DCBXLREC(0)	LRECL WHEN IN XLRI MODE
85 (55)	ADDRESS	3		DCBCNTRA – ADDRESS OF CNTRL MODULE
88 (58)	BITSTRING	2		– RESERVED
90 (5A)	SIGNED	2	DCBPRECL	– FORMAT F RECORDS: BLOCK LENGTH FORMAT. U RECORDS: MAXIMUM BLOCK LENGTH FORMAT. V RECORDS: UNSPANNED RECORDS: MAXIMUM BLOCK LENGTH. SPANNED RECORDS: PUT, NOT DATA MODE: MAXIMUM BLOCK LENGTH; PUT, DATA MODE: DATA LENGTH; GET: SEGMENT CONTROL CODE OF PREVIOUS SEGMENT.
92 (5C)	ADDRESS	4	DCBEOB	– ADDRESS OF END OF BLOCK MODULE

Cross-Reference

Name	Hex Offset	Hex Value	Level
DCBACBM	1B	8	2
DCBAPPIN	12		2
DCBBCKPT	34	10	2
DCBBFA	20	3	2
DCBBFAD	20	2	2
DCBBFAF1	20	1	2
DCBBFAF2	20	3	2
DCBBFALN	20		2
DCBBFT	20	70	2
DCBBFTA	20	60	2
DCBBFTE	20	10	2
DCBBFTEK	20		2
DCBBFTK	20	8	2
DCBBFTKD	20	8	2
DCBBFTKR	20	20	2
DCBBFTR	20	20	2
DCBBFTS	20	40	2
DCBBIT0	0	80	2
DCBBIT1	0	40	2
DCBBIT2	0	20	2
DCBBIT3	0	10	2
DCBBIT4	0	8	2
DCBBIT5	0	4	2
DCBBIT6	0	2	2
DCBBIT7	0	1	2
DCBBLBP	50	40	2
DCBBLKCT	C		2
DCBBLKSI	3E		2

Name	Hex Offset	Hex Value	Level
DCBBUFCA	15		2
DCBBUFCB	14		2
DCBBUFL	18		2
DCBBUFNO	14		2
DCBBUFOF	51		2
DCBBXLRI	20	68	2
DCBCCCW	4C		2
DCBCENDA	42		2
DCBCHCKA	35		2
DCBCHECK	34		2
DCBCICB	44		2
DCBCICBA	45		2
DCBCIND1	3C		2
DCBCIND2	3D		2
DCBCMPAC	10	8	2
DCBCNBFP	3D	8	2
DCBCNBRM	3C	4	2
DCBCNCHS	3D	4	2
DCBCNCI	3C	8	2
DCBCNCLO	3D	20	2
DCBCNEVA	3C	10	2
DCBCNEVB	3C	20	2
DCBCNEXB	3C	1	2
DCBCNFEO	3D	2	2
DCBCNIOE	3D	10	2
DCBCNQSM	3D	1	2
DCBCNSRD	3C	40	2
DCBCNSTO	3D	80	2
DCBCNTOV	3C	80	2
DCBCNTRL	54		2
DCBCNWR0	3D	40	2
DCBCODE	10		2
DCBDCBE	0		2
DCBDDNAM	28		2
DCBDEBA	2D		2
DCBDEBAD	2C		2
DCBDEN	12		2
DCBDEVT	11		2
DCBDIRCQ	51		2
DCBDIRCT	50		2
DCBDSGCX	1A	10	2
DCBDSGDA	1A	20	2
DCBDSGGS	1B	80	2
DCBDSGIS	1A	80	2
DCBDSGPO	1A	2	2
DCBDSGPS	1A	40	2
DCBDSGTQ	1B	20	2
DCBDSGTR	1B	4	2
DCBDSGTX	1B	40	2
DCBDSGU	1A	1	2
DCBDSORG	1A		2
DCBDSPLA	9		2
DCBDSPLY	8		2
DCBDSRG1	1A		2

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Name	Hex Offset	Hex Value	Level
DCBDSRG2	1B		2
DCBDVCPR	11	45	2
DCBDVCP0	11	42	2
DCBDVCP1	11	4C	2
DCBDVCRP	11	43	2
DCBDVCR0	11	41	2
DCBDVCR1	11	44	2
DCBDVCR2	11	46	2
DCBDVDRI	11	59	2
DCBDVMR	11	5D	2
DCBDVMRS	11	56	2
DCBDVMT	11	81	2
DCBDVMT3	11	83	2
DCBDVMT4	11	80	2
DCBDVORS	11	5F	2
DCBDVOR5	11	5A	2
DCBDVOR7	11	5B	2
DCBDVOR8	11	5C	2
DCBDVOR9	11	57	2
DCBDVPRT	11	4D	2
DCBDVPR1	11	48	2
DCBDVPR2	11	4A	2
DCBDVPR3	11	49	2
DCBDVPR4	11	4B	2
DCBDVPR5	11	4E	2
DCBDVPTP	11	50	2
DCBDVTBA	D		2
DCBDVTBL	C		2
DCBDVTRM	11	4F	2
DCBDV301	11	22	2
DCBDV302	11	24	2
DCBDV303	11	23	2
DCBDV305	11	27	2
DCBDV311	11	21	2
DCBDV314	11	28	2
DCBDV321	11	25	2
DCBDV330	11	29	2
DCBDV331	11	2D	2
DCBDV340	11	2A	2
DCBDV350	11	2B	2
DCBDV375	11	2C	2
DCBDV380	11	2E	2
DCBDV390	11	2F	2
DCBD1305	11	26	2
DCBECBLA	D		2
DCBECBLT	C		2
DCBEIB	12		2
DCBEOB	5C		2
DCBEOBAD	48		2
DCBEOBR	48		2
DCBEOBRA	49		2
DCBEOBW	4C		2
DCBEODA	21		2
DCBEODAD	20		2

Name	Hex Offset	Hex Value	Level
DCBEOEA	3C		2
DCBEOPFG	6	80	2
DCBERABE	54	20	2
DCBERACC	54	80	2
DCBERBAM	54	1	2
DCBEROPT	54		2
DCBERRCA	5		2
DCBERRCN	4		2
DCBERSKP	54	40	2
DCBEX	31	40	2
DCBEXLSA	25		2
DCBEXLST	24		2
DCBFDAD	5		2
DCBFLAG1	3C		2
DCBFNCBD	13	8	2
DCBFNCBI	13	80	2
DCBFNCBP	13	20	2
DCBFNCBR	13	40	2
DCBFNCBT	13	2	2
DCBFNCBW	13	10	2
DCBFNCBX	13	4	2
DCBFRID	8		2
DCBFRTBA	C		2
DCBFUNC	13		2
DCBGERR	34		2
DCBGERRA	35		2
DCBGET	30		2
DCBGETA	31		2
DCBHDR	C		2
DCBHIARC	20		2
DCBH0	20	4	2
DCBH1	20	80	2
DCBIBEC	31	C0	2
DCBIBIOE	31	C	2
DCBIBPCT	31	30	2
DCBICQE	1C		2
DCBICQEA	1D		2
DCBIFC12	31	10	2
DCBIFC9	31	20	2
DCBIFEC	2C	C0	2
DCBIFER	31	0	2
DCBIFIOE	2C	C	2
DCBIFLDT	2C	2	2
DCBIFLG	31		2
DCBIFLGS	2C		2
DCBIFNEP	31	0	2
DCBIFNE1	31	4	2
DCBIFNE2	31	8	2
DCBIFNE3	31	C	2
DCBIFPCT	2C	30	2
DCBIFPEC	31	C0	2
DCBIFTIM	31	4	2
DCBIMAGA	9		2
DCBIMAGE	8		2

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Name	Hex Offset	Hex Value	Level
DCBIMG	4		2
DCBIOBA	44		2
DCBIOBAA	1D		2
DCBIOBAD	1C		2
DCBIOBB	45		2
DCBIOBL	38		2
DCBKEYCN	4		2
DCBKEYLE	10		2
DCBLCCW	48		2
DCBLCTBL	8		2
DCBLFMAT	5		2
DCBLNGBS	54	58	2
DCBLNGPO	54	58	2
DCBLNGPS	5C	60	2
DCBLNGQS	5C	60	2
DCBLNGXA	46	48	2
DCBLNGXE	33	34	2
DCBLNNUM	4		2
DCBLNP	1C		2
DCBLRECL	52		2
DCBMACF1	2A		2
DCBMACF2	2B		2
DCBMACR	32		2
DCBMACRF	2A		2
DCBMACR1	32		2
DCBMACR2	33		2
DCBMODE	10		2
DCBMODEC	10	80	2
DCBMODEE	10	40	2
DCBMODEO	10	20	2
DCBMODER	10	10	2
DCBMRABC	32	4	2
DCBMRAPG	32	20	2
DCBMRAWR	33	2	2
DCBMRBCT	8	C0	2
DCBMRCHK	32	2	2
DCBMRCI	32	10	2
DCBMRCK	32	1	2
DCBMRCRL	32	2	2
DCBMRCTL	33	2	2
DCBMRDBF	32	4	2
DCBMRDBG	10	40	2
DCBMRDCT	C	E0	2
DCBMRDMD	33	1	2
DCBMRDMG	32	1	2
DCBMRDR	10	10	2
DCBMRDRU	10	20	2
DCBMRDWT	10	2	2
DCBMRECP	32	80	2
DCBMRERP	C	2	2
DCBMRERS	C	1	2
DCBMRFE	32	40	2
DCBMRFG	8		2
DCBMRFLG	10		2

Name	Hex Offset	Hex Value	Level
DCBMRGET	32	40	2
DCBMRGTQ	33	40	2
DCBMRIDW	33	8	2
DCBMRIND	C		2
DCBMRLCG	32	8	2
DCBMRLCP	33	8	2
DCBMRLDM	33	8	2
DCBMRMVG	32	10	2
DCBMRMVP	33	10	2
DCBMRPCC	10	C	2
DCBMRPLO	C	8	2
DCBMRPLS	C	4	2
DCBMRPTQ	32	40	2
DCBMRPT1	32	4	2
DCBMRPT2	33	4	2
DCBMRPUT	33	40	2
DCBMRRD	32	20	2
DCBMRRDI	32	8	2
DCBMRRDK	32	10	2
DCBMRRDQ	32	20	2
DCBMRRDX	32	2	2
DCBMRSBG	32	4	2
DCBMRSCC	10	80	2
DCBMRSCU	C	10	2
DCBMRSTI	33	1	2
DCBMRSTK	33	2	2
DCBMRSTL	33	80	2
DCBMRSWA	33	1	2
DCBMRTMD	33	4	2
DCBMRUE	10	1	2
DCBMRUIP	33	4	2
DCBMRWRK	33	10	2
DCBMRWRQ	32	20	2
DCBMRWRT	33	20	2
DCBMR1WD	33	1	2
DCBMR3WD	33	2	2
DCBMR4WD	33	4	2
DCBMR5WD	33	8	2
DCBMTC	10	13	2
DCBMTDN0	12	3	2
DCBMTDN1	12	43	2
DCBMTDN2	12	83	2
DCBMTDN3	12	C3	2
DCBMTDN4	12	D3	2
DCBMTE	10	23	2
DCBMTET	10	2B	2
DCBMTT	10	3B	2
DCBNCP	48		2
DCBNOPAC	10	4	2
DCBNOTE	54		2
DCBNUPD	1C	30	2
DCBODEB	1C		2
DCBODEBA	1D		2
DCBOFEOV	30	20	2

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Name	Hex Offset	Hex Value	Level
DCBOFFSR	42		2
DCBOFFSW	43		2
DCBOFIOD	30	80	2
DCBOFIOF	30	1	2
DCBOFLG	30		2
DCBOFLGS	30		2
DCBOFLG1	30		2
DCBOFLRB	30	40	2
DCBOFLWR	30	80	2
DCBOFOPN	30	10	2
DCBOFPPC	30	8	2
DCBOFTM	30	4	2
DCBOFUEX	30	2	2
DCBOPTC	34	20	2
DCBOPTCD	34		2
DCBOPTH	34	10	2
DCBOPTJ	34	1	2
DCBOPTO	34	10	2
DCBOPTQ	34	8	2
DCBOPTT	34	2	2
DCBOPTU	34	40	2
DCBOPTW	34	80	2
DCBOPTZ	34	4	2
DCBORBFP	10	20	2
DCBORBYT	10		2
DCBORDCK	12	1	2
DCBORECK	12	8	2
DCBOREOF	10	40	2
DCBORERR	12	10	2
DCBORFLG	6		2
DCBORHPR	12	2	2
DCBORNRM	12	40	2
DCBORREJ	12	20	2
DCBORSYN	10	80	2
DCBORWLR	12	4	2
DCBPCIA	3E		2
DCBPERR	34		2
DCBPERRA	35		2
DCBPGFXA	32	2	2
DCBPOINT	54		2
DCBPOPEN	50	4	2
DCBPRBYT	13		2
DCBPRC12	12	10	2
DCBPRC9	12	20	2
DCBPRECL	5A		2
DCBPRSP0	10	1	2
DCBPRSP1	10	9	2
DCBPRSP2	10	11	2
DCBPRSP3	10	19	2
DCBPRTOV	12		2
DCBPRTSP	10		2
DCBPTCDA	10	4	2
DCBPTCDB	10	10	2
DCBPTCDC	10	8	2

Name	Hex Offset	Hex Value	Level
DCBPTCDF	10	20	2
DCBPTCDI	10	40	2
DCBPTCDN	10	80	2
DCBPTCDT	10	2	2
DCBPTECR	13	4	2
DCBPTECT	13	8	2
DCBPTERR	13	1	2
DCBPTFLG	13		2
DCBPTIC	13	10	2
DCBPTUCT	13	2	2
DCBPUT	30		2
DCBPUTA	31		2
DCBQADFS	50	38	2
DCBQADF1	50	20	2
DCBQADF2	50	10	2
DCBQADF3	50	8	2
DCBQSLM	1C		2
DCBQSMEX	0		2
DCBQSTRU	50	1	2
DCBQSWS	50		2
DCBRCFGT	4C	40	2
DCBRCREL	4C	F0	2
DCBRCTRU	4C	80	2
DCBRDLNA	D		2
DCBRDLNE	C		2
DCBREAD	30		2
DCBREADA	31		2
DCBRECA	4D		2
DCBRECAD	4C		2
DCBRECBR	24	10	2
DCBRECBT	4C		2
DCBRECC	24	0	2
DCBRECCA	24	4	2
DCBRECCC	24	6	2
DCBRECCM	24	2	2
DCBRECD	24	20	2
DCBRECF	24	80	2
DCBRECFM	24		2
DCBRECKL	24	1	2
DCBRECL	24	C0	2
DCBRECLA	24	E0	2
DCBRECSB	24	8	2
DCBRECTO	24	20	2
DCBRECU	24	C0	2
DCBRECV	24	40	2
DCBREL	11		2
DCBRELAD	0		2
DCBRELB	10		2
DCBRESCA	D		2
DCBRESCN	C		2
DCBSIOA	40		2
DCBSRCHD	34	4	2
DCBSSAD	4		2
DCBSSADA	5		2

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Name	Hex Offset	Hex Value	Level
DCBSSID	0		2
DCBSTACK	10		2
DCBSTCK1	10	1	2
DCBSTCK2	10	2	2
DCBSTFLS	3C	40	2
DCBSTQCK	3C	80	2
DCBSVCXA	D		2
DCBSVCXL	1C		2
DCBSVDEB	1C	10	2
DCBSYNA	39		2
DCBSYNAD	38		2
DCBTIOT	28		2
DCBTRBAL	12		2
DCBTRCID	13	3	2
DCBTRTCH	10		2
DCBUPDBT	1C	30	2
DCBUPDCM	1C	40	2
DCBUPDT	1C	20	2
DCBUSASI	50		2
DCBWCPL	41		2
DCBWCPO	40		2
DCBWRITA	31		2
DCBWRITE	30		2
DCBWTOIA	1		2
DCBWTOID	0		2
DCBXENDA	44		2
DCBXLREC	55		2
DCB1DVDS	1C	80	2
DCB1TRK	10	42	2
DCB2TRK	10	82	2
DCB3525A	50	2	2
DCB4TRK	10	C2	2

DCB for ISAM

Common Name: Data Control Block for ISAM
Macro ID: DCBD
DSECT Name: IHADCB
Owning Component: Data Management, subcomponent OPEN/CLOSE/EOV
Eye-Catcher ID: None
Subpool and Key: Problem program subpool and key
Size: 252 bytes
Created by: Problem program
Pointed to by:

- DEBDCBAD field of the DEB data area
- IOBDCBPT field of the IOB data area
- DECDCBAD field of the DECB data area

Serialization: None — DEB validity check ensures the contents of this DCB.
Function: This data control block (DCB) is used by the indexed sequential access method (ISAM) routines and holds data pertinent to the use of a data set that is maintained by the ISAM routines. The common interface and the foundation sections serve the same purpose in all DCBs although the format might vary slightly for each access method routine.

Offset	Type/Value	Len	Name (Dim)	Description
DATA CONTROL BLOCK DEFINITION				
	X'80'		DCBBIT0	″128″
	X'40'		DCBBIT1	″64″
	X'20'		DCBBIT2	″32″
	X'10'		DCBBIT3	″16″
	X'8'		DCBBIT4	″8″
	X'4'		DCBBIT5	″4″
	X'2'		DCBBIT6	″2″
	X'1'		DCBBIT7	″1″
	ACCESS METHOD COMMON INTERFACE			
16 (10)	SIGNED	4	DCBRELB(0)	– SAME AS DCBREL BELOW
16 (10)	SIGNED	1	DCBKEYLE	– KEY LENGTH OF DATA SET
17 (11)	CHARACTER	1	DCBDEVT(0)	– DEVICE TYPE
	.1. . 1111		DCBDVTRM	″X'4F'″– TERMINAL. (DD CONTAINS TERM=TS)
17 (11)	SIGNED	3	DCBREL	– NUMBER OF RELATIVE TRACKS OR BLOCKS IN THIS DATA SET (BDAM)
20 (14)	ADDRESS	4	DCBBUFCB(0)	– ADDRESS OF BUFFER POOL CONTROL BLOCK
20 (14)	SIGNED	1	DCBBUFNO	– NUMBER OF BUFFERS REQUIRED FOR THIS DATA SET. MAY RANGE FROM 0 TO 255. IF UNBLOCKED SPANNED RECORDS ARE USED, NUMBER OF SEGMENT WORK AREAS REQUIRED FOR THIS DATA SET.
21 (15)	ADDRESS	3	DCBBUFCA	– ADDRESS OF BUFFER POOL CONTROL BLOCK
24 (18)	SIGNED	2	DCBBUFL	– LENGTH OF BUFFER. MAY RANGE FROM 0 TO 32,767.
26 (1A)	BITSTRING	2	DCBDSORG(0)	– DATA SET ORGANIZATION BEING USED
26 (1A)	BITSTRING	1	DCBDSRG1	– FIRST BYTE OF DCBDSORG

Offset	Type/Value	Len	Name (Dim)	Description
	1.		DCBDSGIS	″DCBBIT0″– IS – INDEXED SEQUENTIAL ORGANIZATION
	.1.		DCBDSGPS	″DCBBIT1″– PS – PHYSICAL SEQUENTIAL ORGANIZATION
	. .1.		DCBDSGDA	″DCBBIT2″– DA – DIRECT ORGANIZATION
	. . .1		DCBDSGCX	″DCBBIT3″– CX – BTAM OR QTAM LINE GROUP
	. . . . . .1.		DCBDSGPO	″DCBBIT6″– PO – PARTITIONED ORGANIZATION
	1		DCBDSGU	″DCBBIT7″– U – UNMOVABLE, THE DATA CONTAINS LOCATION DEPENDENT INFORMATION
27 (1B)	BITSTRING	1	DCBDSRG2	– SECOND BYTE OF DCBDSORG
	1.		DCBDSGGS	″DCBBIT0″– GS – GRAPHICS ORGANIZATION
	.1.		DCBDSGTX	″DCBBIT1″– TX – TCAM LINE GROUP
	. .1.		DCBDSGTQ	″DCBBIT2″– TQ – TCAM MESSAGE QUEUE
	. . . . 1. . .		DCBACBM	″DCBBIT4″– ACCESS METHOD CONTROL BLOCK
	. . . . .1. .		DCBDSGTR	″DCBBIT5″– TR – TCAM 3705
28 (1C)	ADDRESS	4	DCBIOBAD(0)	– ADDRESS OF IOB WHEN CHAINED SCHEDULING IS USED OR FOR 1419/1275
28 (1C)	ADDRESS	4	DCBICQE(0)	– ADDRESS OF ICQE
28 (1C)	ADDRESS	4	DCBODEB(0)	– ADDRESS OF OLD DEB
28 (1C)	SIGNED	1	DCBLNP(0)	– 3525 PRINTER LINE POSITION COUNTER
28 (1C)	BITSTRING	1	DCBQSLM	– QSAM LOCATE MODE LOGICAL RECORD INTERFACE INDICATOR BYTE FOR UPDATE PROCESSING OF SPANNED RECORDS
	1.		DCB1DVDS	″DCBBIT0″– ONLY ONE DEVICE IS ALLOCATED TO THIS DATA SET
	.1.		DCBUPDCM	″DCBBIT1″– UPDATE COMPLETE, FREE OLD DEB
	. .11		DCBUPDBT	″DCBBIT2+DCBBIT3″– UPDATE BITS
	. .1.		DCBUPDT	″DCBBIT2″– UPDATE TO TAKE PLACE
	. .11		DCBNUPD	″DCBBIT2+DCBBIT3″– NO UPDATE TO TAKE PLACE
	. . .1		DCBSVDEB	″DCBBIT3″– OLD DEB ADDRESS MUST BE SAVED
29 (1D)	ADDRESS	3	DCBIOBAA(0)	– SAME AS DCBIOBAD ABOVE
29 (1D)	ADDRESS	3	DCBICQEA(0)	– SAME AS DCBICQE ABOVE
29 (1D)	ADDRESS	3	DCBODEBA	–ADDRESS OF OLD DEB
28 (1C)	ADDRESS	4	DCBSVCXL(0)	–SAME AS DCBSVCXA BELOW
28 (1C)	BITSTRING	1		– RESERVED
29 (1D)	ADDRESS	3	DCBSVCXA	– POINTER TO EXIT LIST OF JES C.I. INTERFACE CONTROL SVC
	FOUNDATION EXTENSION			
32 (20)	ADDRESS	4	DCBEODAD(0)	– SAME AS DCBEODA BELOW
32 (20)	BITSTRING	1	DCBHIARC(0)	– HIERARCHY BITS
32 (20)	BITSTRING	1	DCBBFTEK(0)	– BUFFERING TECHNIQUE BITS
32 (20)	BITSTRING	1	DCBBFALN	– BUFFER ALIGNMENT BITS
	1.		DCBH1	″DCBBIT0″– HIERARCHY 1 MAIN STORAGE – BIT 5 IS ZERO
	.111		DCBBFT	″DCBBIT1+DCBBIT2+DCBBIT3″ BUFFERING TECHNIQUE
	.11.		DCBBFTA	″DCBBIT1+DCBBIT2″– QSAM LOCATE MODE PROCESSING OF SPANNED RECORDS – OPEN IS TO CONSTRUCT A RECORD AREA IF IT AUTOMATICALLY CONSTRUCTS BUFFERS

Offset	Type/Value	Len	Name (Dim)	Description
	. .1.		DCBBFTR	″DCBBIT2″– FOR BSAM CREATE BDAM PROCESSING OF UNBLOCKED SPANNED RECORDS – SOFTWARE TRACK OVERFLOW. FOR BSAM INPUT PROCESSING OF UNBLOCKED SPANNED RECORDS WITH KEYS – RECORD OFFSET PROCESSING.
	.1.		DCBBFTS	″DCBBIT1″– SIMPLE BUFFERING – BIT 3 IS ZERO
	. .1.		DCBBFTKR	″DCBBIT2″– UNBLOCKED SPANNED RECORDS – SOFTWARE TRACK OVERFLOW (BDAM)
	. . .1		DCBBFTE	″DCBBIT3″– EXCHANGE BUFFERING – BIT 1 IS ZERO
	. . . . 1. . .		DCBBFTKD	″DCBBIT4″– DYNAMIC BUFFERING (BTAM)
	. . . . 1. . .		DCBBFTK	″DCBBIT4″– LRECL IN ’K’ UNITS FOR XLRI
	.11. 1. . .		DCBBXLRI	″DCBBIT1+DCBBIT2+DCBBIT4″– EXTENDED LOGICAL RECORD INTERFACE MODE FOR SPANNED RECORDS GREATER THAN 32K – ALL RECORDS TRANSFERRED TO OR FROM THE RECORD AREA.
	. . . . .1. .		DCBH0	″DCBBIT5″– HIERARCHY 0 MAIN STORAGE – BIT 0 IS ZERO
	11		DCBBFA	″DCBBIT6+DCBBIT7″– BUFFER ALIGNMENT
	. . . . . .1.		DCBBFAD	″DCBBIT6″– DOUBLEWORD BOUNDARY
	1		DCBBFAF1	″DCBBIT7″– FULLWORD NOT A DOUBLEWORD BOUNDARY, CODED IN DCB MACRO INSTRUCTION
	11		DCBBFAF2	″DCBBIT6+DCBBIT7″– FULLWORD NOT A DOUBLEWORD BOUNDARY, CODED IN DCB MACRO INSTRUCTION
33 (21)	ADDRESS	3	DCBEODA	– ADDRESS OF A USER– PROVIDED ROUTINE TO HANDLE END-OF-DATA CONDITIONS
36 (24)	ADDRESS	4	DCBEXLST(0)	– ADDRESS OF USER– PROVIDED LIST OF EXITS
36 (24)	BITSTRING	1	DCBRECFM	– RECORD FORMAT
	111.		DCBRECLA	″DCBBIT0+DCBBIT1+DCBBIT2″ RECORD LENGTH INDICATOR –ASCII
	. .1.		DCBRECD	″DCBBIT2″– ASCII VARIABLE RECORD LENGTH
	11.		DCBRECL	″DCBBIT0+DCBBIT1″– RECORD LENGTH INDICATOR
	1.		DCBRECF	″DCBBIT0″–FIXED RECORD LENGTH
	.1.		DCBRECV	″DCBBIT1″– VARIABLE RECORD LENGTH
	11.		DCBRECU	″DCBBIT0+DCBBIT1″– UNDEFINED RECORD LENGTH
	. .1.		DCBRECTO	″DCBBIT2″– TRACK OVERFLOW
	. . .1		DCBRECBR	″DCBBIT3″– BLOCKED RECORDS
	. . . . 1. . .		DCBRECSB	″DCBBIT4″– FOR FIXED LENGTH RECORD FORMAT – STANDARD BLOCKS. FOR VARIABLE LENGTH RECORD FORMAT – SPANNED RECORDS
	. . . . .11.		DCBRECCC	″DCBBIT5+DCBBIT6″– CONTROL CHARACTER INDICATOR
	. . . . .1. .		DCBRECCA	″DCBBIT5″– ASA CONTROL CHARACTER
	. . . . . .1.		DCBRECCM	″DCBBIT6″– MACHINE CONTROL CHARACTER
	. . . . .		DCBRECC	″X'00'″– NO CONTROL CHARACTER
	1		DCBRECKL	″DCBBIT7″– KEY LENGTH (KEYLEN) WAS SPECIFIED IN DCB MACRO INSTRUCTION
37 (25)	ADDRESS	3	DCBEXLSA	– ADDRESS OF USER–PROVIDED LIST OF EXITS
	FOUNDATION BEFORE OPEN			

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Offset	Type/Value	Len	Name (Dim)	Description
40 (28)	CHARACTER	8	DCBDDNAM	– NAME ON THE DD STATEMENT THAT DEFINES THE DATA SET ASSOCIATED WITH THIS DCB
48 (30)	BITSTRING	1	DCBOFLGS	– FLAGS USED BY OPEN ROUTINE
	1.		DCBOFLWR	″DCBBIT0″– IF ZERO, LAST I/O OPERATION WAS READ OR POINT. IF ONE, LAST I/O OPERATION WAS WRITE.
	1.		DCBOFIOD	″DCBBIT0″– DATA SET IS BEING OPENED FOR INPUT OR OUTPUT (BDAM)
	.1.		DCBOFLRB	″DCBBIT1″– LAST I/O OPERATION WAS IN READ BACKWARD MODE
	. .1.		DCBOFEOV	″DCBBIT2″– SET TO 1 BY EOV WHEN IT CALLS CLOSE ROUTINE FOR CONCATENATION OF DATA SETS WITH UNLIKE ATTRIBUTES
	. . .1		DCBOFOPN	″DCBBIT3″– AN OPEN HAS BEEN SUCCESSFULLY COMPLETED
	. . . . 1. . .		DCBOFPPC	″DCBBIT4″– SET TO 1 BY PROBLEM PROGRAM TO INDICATE A CONCATENATION OF UNLIKE ATTRIBUTES
	. . . . .1. .		DCBOFTM	″DCBBIT5″– TAPE MARK HAS BEEN READ
	. . . . . .1.		DCBOFUEX	″DCBBIT6″– SET TO 0 BY AN I/O SUPPORT FUNCTION WHEN THAT FUNCTION TAKES A USER EXIT. SET TO 1 ON RETURN FROM USER EXIT TO THE I/O SUPPORT FUNCTION WHICH TOOK THE EXIT.
	1		DCBOFIOF	″DCBBIT7″– 1 BY AN I/O SUPPORT FUNCTION IF DCB IS TO BE PROCESSED BY THAT FUNCTION
49 (31)	BITSTRING	1	DCBIFLG	– FLAGS USED BY IOS IN COMMUNICATING ERROR CONDITIONS AND IN DETERMINING CORRECTIVE PROCEDURES
	11.		DCBIBEC	″DCBBIT0+DCBBIT1″– ERROR CORRECTION INDICATOR
	. . . . .		DCBIFNEP	″X'00'″– NOT IN ERROR PROCEDURE
	.1.		DCBEX	″DCBBIT1″– ERROR CORRECTION OR IOS PAGE FIX IN PROCESS
	11.		DCBIFPEC	″DCBBIT0+DCBBIT1″– PERMANENT ERROR CORRECTION
	. .11		DCBIBPCT	″DCBBIT2+DCBBIT3″– PRINTER CARRIAGE TAPE PUNCH INDICATOR
	. .1.		DCBIFC9	″DCBBIT2″– CHANNEL 9 PRINTER CARRIAGE TAPE PUNCH SENSED
	. . .1		DCBIFC12	″DCBBIT3″ – CHANNEL 12 PRINTER CARRIAGE TAPE PUNCH SENSED
	. . . . 11. .		DCBIBIOE	″DCBBIT4+DCBBIT5″– IOS ERROR ROUTINE USE INDICATOR
	. . . . .		DCBIFER	″X'00'″– ALWAYS USE I/O SUPERVISOR ERROR ROUTINE
	. . . . .1. .		DCBIFNE1	″DCBBIT5″– NEVER USE I/O SUPERVISOR ERROR ROUTINE
	. . . . .1. .		DCBIFTIM	″DCBBIT5″– TEST IOS MASK (IMSK) FOR ERROR PROCEDURE (BTAM)
	. . . . 1. . .		DCBIFNE2	″DCBBIT4″– NEVER USE I/O SUPERVISOR ERROR ROUTINE
	. . . . 11. .		DCBIFNE3	″DCBBIT4+DCBBIT5″– NEVER USE I/O SUPERVISOR ERROR ROUTINE
50 (32)	BITSTRING	2	DCBMACR(0)	– MACRO INSTRUCTION REFERENCE
50 (32)	BITSTRING	1	DCBMACR1	– FIRST BYTE OF DCBMACR

Offset	Type/Value	Len	Name (Dim)	Description
	1.		DCBMRECP	″DCBBIT0″– EXECUTE CHANNEL PROGRAM (EXCP) – – – ALWAYS ZERO (BSAM, QSAM, BPAM, BISAM, QISAM, BDAM) – – – RESERVED (QTAM, BTAM)
	.1.		DCBMRFE	″DCBBIT1″– FOUNDATION EXTENSION IS PRESENT (EXCP)
	.1.		DCBMRGET	″DCBBIT1″– GET (QSAM, QISAM, TCAM)
	.1.		DCBMRPTQ	″DCBBIT1″– PUT FOR MESSAGE GROUP (QTAM) – – – ALWAYS ZERO (BSAM, BPAM, BISAM, BDAM) – – – RESERVED (BTAM)
	. .1.		DCBMRAPG	″DCBBIT2″– APPENDAGES ARE REQUIRED (EXCP)
	. .1.		DCBMRRD	″DCBBIT2″– READ (BSAM, BPAM, BISAM, BDAM, BTAM)
	. .1.		DCBMRWRQ	″DCBBIT2″– WRITE FOR LINE GROUP (QTAM) – – – ALWAYS ZERO (QSAM, QISAM)
	. . .1		DCBMRCI	″DCBBIT3″– COMMON INTERFACE (EXCP)
	. . .1		DCBMRMVG	″DCBBIT3″– MOVE MODE OF GET (QSAM, QISAM)
	. . .1		DCBMRRDK	″DCBBIT3″– KEY SEGMENT WITH READ (BDAM) – – – ALWAYS ZERO (BISAM) – – – RESERVED (BSAM, BPAM, QTAM, BTAM)
	. . . . 1. . .		DCBMRLCG	″DCBBIT4″– LOCATE MODE OF GET (QSAM, QISAM)
	. . . . 1. . .		DCBMRRDI	″DCBBIT4″– ID ARGUMENT WITH READ (BDAM) – – – ALWAYS ZERO (BISAM) – – – RESERVED (EXCP, BSAM, BPAM, QTAM, BTAM)
	. . . . .1. .		DCBMRABC	″DCBBIT5″– USER'S PROGRAM MAINTAINS ACCURATE BLOCK COUNT (EXCP)
	. . . . .1. .		DCBMRPT1	″DCBBIT5″– POINT (WHICH IMPLIES NOTE) (BSAM, BPAM)
	. . . . .1. .		DCBMRSBG	″DCBBIT5″– SUBSTITUTE MODE OF GET (QSAM)
	. . . . .1. .		DCBMRDBF	″DCBBIT5″–DYNAMIC BUFFERING (BISAM, BDAM)– – – ALWAYS ZERO (QISAM) – – – RESERVED (QTAM, BTAM)
	. . . . . .1.		DCBPGFXA	″DCBBIT6″– PAGE FIX APPENDAGE IS SPECIFIED (EXCP)
	. . . . . .1.		DCBMRCRL	″DCBBIT6″– CNTRL (BSAM, QSAM)
	. . . . . .1.		DCBMRCHK	″DCBBIT6″– CHECK (BISAM)
	. . . . . .1.		DCBMRRDX	″DCBBIT6″– READ EXCLUSIVE (BDAM) – – – RESERVED (BPAM, QISAM, QTAM, BTAM)
	1		DCBMRDMG	″DCBBIT7″–DATA MODE OF GET (QSAM)
	1		DCBMRCK	″DCBBIT7″– CHECK (BDAM) – – – RESERVED (EXCP, BSAM, BPAM, BISAM, QISAM, QTAM, BTAM)
51 (33)	BITSTRING	1	DCBMACR2	– SECOND BYTE OF DCBMACR
	1.		DCBMRSTL	″DCBBIT0″– SETL (QISAM) – – – ALWAYS ZERO (BSAM, QSAM, BPAM, BISAM, BDAM) – – – RESERVED (EXCP, QTAM, BTAM)
	.1.		DCBMRPUT	″DCBBIT1″– PUT (QSAM, TCAM) – PUT OR PUTX (QISAM)
	.1.		DCBMRGTQ	″DCBBIT1″– GET FOR MESSAGE GROUP (QTAM) – – –ALWAYS ZERO (BSAM, BPAM, BISAM, BDAM) – – – RESERVED (EXCP, BTAM)
	. .1.		DCBMRWRT	″DCBBIT2″– WRITE (BSAM, BPAM, BISAM, BDAM, BTAM)
	. .1.		DCBMRRDQ	″DCBBIT2″– READ FOR LINE GROUP (QTAM) – – – ALWAYS ZERO (QSAM, QISAM) – – – RESERVED (EXCP)

Offset	Type/Value	Len	Name (Dim)	Description
	. . .1		DCBMRMVP	″DCBBIT3″– MOVE MODE OF PUT (QSAM, QISAM)
	. . .1		DCBMRWRK	″DCBBIT3″– KEY SEGMENT WITH WRITE (BDAM) – – – ALWAYS ZERO (BISAM) – – – RESERVED (EXCP, BSAM, BPAM, QTAM, BTAM)
	. . . . 1. . .		DCBMR5WD	″DCBBIT4″– FIVE-WORD DEVICE INTERFACE (EXCP)
	. . . . 1. . .		DCBMRLDM	″DCBBIT4″– LOAD MODE BSAM (CREATE BDAM DATA SET) (BSAM)
	. . . . 1. . .		DCBMRLCP	″DCBBIT4″– LOCATE MODE OF PUT (QSAM, QISAM)
	. . . . 1. . .		DCBMRIDW	″DCBBIT4″– ID ARGUMENT WITH WRITE (BDAM) – – ALWAYS ZERO (BISAM) – – – RESERVED (BPAM, QTAM, BTAM)
	. . . . .1. .		DCBMR4WD	″DCBBIT5″– FOUR-WORD DEVICE INTERFACE (EXCP)
	. . . . .1. .		DCBMRPT2	″DCBBIT5″– POINT (WHICH IMPLIES NOTE) (BSAM, BPAM)
	. . . . .1. .		DCBMRTMD	″DCBBIT5″– SUBSTITUTE MODE (QSAM)
	. . . . .1. .		DCBMRUIP	″DCBBIT5″– UPDATE IN PLACE (PUTX) (QISAM) – – –ALWAYS ZERO (BISAM) – – – RESERVED (BDAM, QTAM, BTAM)
	. . . . . .1.		DCBMR3WD	″DCBBIT6″– THREE–WORD DEVICE INTERFACE (EXCP)
	. . . . . .1.		DCBMRCTL	″DCBBIT6″– CNTRL (BSAM, QSAM)
	. . . . . .1.		DCBMRSTK	″DCBBIT6″– SETL BY KEY (QISAM)
	. . . . . .1.		DCBMRAWR	″DCBBIT6″– ADD TYPE OF WRITE (BDAM) – – – ALWAYS ZERO (BISAM) – – – RESERVED (BPAM, QTAM, BTAM)
	1		DCBMR1WD	″DCBBIT7″– ONE–WORD DEVICE INTERFACE (EXCP)
	1		DCBMRSWA	″DCBBIT7″– USER'S PROGRAM HAS PROVIDED A SEGMENT WORK AREA POOL (BSAM CREATE BDAM, BDAM)
	1		DCBMRDMD	″DCBBIT7″– DATA MODE (QSAM)
	1		DCBMRSTI	″DCBBIT7″– SETL BY ID (QISAM) – – – ALWAYS ZERO (BISAM) – – – RESERVED (BPAM, QTAM, BTAM)
	FOUNDATION AFTER OPEN			
40 (28)	ADDRESS	2	DCBTIOT	OFFSET FROM TIOT ORIGIN TO TIOELNGH FIELD IN TIOT ENTRY FOR DD STATEMENT ASSOCIATED WITH THIS DCB
42 (2A)	BITSTRING	2	DCBMACRF(0)	– SAME AS DCBMACR BEFORE OPEN
42 (2A)	BITSTRING	1	DCBMACF1	– FIRST BYTE OF DCBMACRF
43 (2B)	BITSTRING	1	DCBMACF2	– SECOND BYTE OF DCBMACRF
44 (2C)	ADDRESS	4	DCBDEBAD(0)	– ADDRESS OF ASSOCIATED DEB
44 (2C)	BITSTRING	1	DCBIFLGS	– SAME AS DCBIFLG BEFORE OPEN
	11.		DCBIFEC	″DCBBIT0+DCBBIT1″– ERROR CORRECTION INDICATOR
	. .11		DCBIFPCT	″DCBBIT2+DCBBIT3″– PRINTER CARRIAGE TAPE PUNCH INDICATOR
	. . . . 11. .		DCBIFIOE	″DCBBIT4+DCBBIT5″– IOS ERROR ROUTINE USE INDICATOR
	. . . . . .1.		DCBIFLDT	″DCBBIT6″– POSSIBLE LOST DATA CONDITION SUCH AS FOR A PRINTER
45 (2D)	ADDRESS	3	DCBDEBA	– ADDRESS OF ASSOCIATED DEB
48 (30)	ADDRESS	4	DCBGET(0)	– ADDRESS OF GET MODULE

Offset	Type/Value	Len	Name (Dim)	Description
48 (30)	ADDRESS	4	DCBPUT(0)	– ADDRESS OF PUT MODULE
48 (30)	BITSTRING	1	DCBOFLG1	SAME AS DCBOFLGS BEFORE OPEN
49 (31)	ADDRESS	3	DCBGETA(0)	ADDRESS OF GET MODULE
49 (31)	ADDRESS	3	DCBPUTA	ADDRESS OF PUT MODULE
	BISAM–QISAM INTERFACE			
52 (34)	BITSTRING	1	DCBOPTCD	– OPTION CODES
	1.		DCBOPTW	″DCBBIT0″– WRITE VALIDITY CHECK (DASD) (BSAM, BPAM, QSAM, ISAM, BDAM)
	.1.		DCBOPTUF	″DCBBIT1″– FULL–TRACK INDEX WRITE
	. .1.		DCBOPTM	″DCBBIT2″– MASTER INDEXES
	. . .1		DCBOPTI	″DCBBIT3″– INDEPENDENT OVERFLOW AREA
	. . . . 1. . .		DCBOPTY	″DCBBIT4″– CYLINDER OVERFLOW AREA
	. . . . . .1.		DCBOPTL	″DCBBIT6″– DELETE OPTION
	1		DCBOPTR	″DCBBIT7″– REORGANIZATION CRITERIA
53 (35)	BITSTRING	1	DCBMAC	– EXTENSION OF DCBMACRF FIELD FOR ISAM
	. . . . 1. . .		DCBMACUR	″DCBBIT4″– UPDATE FOR READ
	. . . . .1. .		DCBMACUW	″DCBBIT5″– UPDATE TYPE OF WRITE
	. . . . . .1.		DCBMACAW	″DCBBIT6″– ADD TYPE OF WRITE
	1		DCBMACRE	″DCBBIT7″– READ EXCLUSIVE
54 (36)	SIGNED	1	DCBNTM	– NUMBER OF TRACKS THAT DETERMINE THE DEVELOPMENT OF A MASTER INDEX MAXIMUM PERMISSIBLE VALUE – 99
55 (37)	SIGNED	1	DCBCYLOF	– NUMBER OF TRACKS TO BE RESERVED ON EACH PRIME DATA CYLINDER FOR RECORDS THAT OVERFLOW FROM OTHER TRACKS ON THAT CYLINDER
56 (38)	ADDRESS	4	DCBSYNAD	– ADDRESS OF USER'S SYNAD ROUTINE
60 (3C)	SIGNED	2	DCBRKP	– RELATIVE POSITION OF FIRST BYTE OF KEY WITHIN EACH LOGICAL RECORD
62 (3E)	SIGNED	2	DCBBLKSI	– BLOCK SIZE
64 (40)	BITSTRING	8	DCBLPDT(0)	– FOR RESUME LOAD, THE LAST PRIME DATA TRACK ON THE LAST PRIME DATA CYLINDER IN THE FORM MBBCCHHR.
64 (40)	ADDRESS	4	DCBMSWA	– ADDRESS OF MAIN STORAGE WORK AREA FOR USE BY CONTROL PROGRAM WHEN NEW RECORDS ARE BEING ADDED TO AN EXISTING DATA SET
68 (44)	SIGNED	2	DCBSMSI	– NUMBER OF BYTES IN AREA RESERVED TO HOLD HIGHEST LEVEL INDEX
70 (46)	SIGNED	2	DCBSMSW	– NUMBER OF BYTES IN WORK AREA USED BY CONTROL PROGRAM WHEN NEW RECORDS ARE BEING ADDED TO DATA SET
72 (48)	ADDRESS	4	DCBMSHI(0)	– ADDRESS OF MAIN STORAGE AREA TO HOLD HIGHEST LEVEL INDEX
72 (48)	SIGNED	1	DCBNCP	– NUMBER OF COPIES OF READ–WRITE (TYPE K) CHANNEL PROGRAMS THAT ARE TO BE ESTABLISHED FOR THIS DCB. (99 MAXIMUM)
73 (49)	ADDRESS	3	DCBMSHIA	– SAME AS DCBMSHI ABOVE
76 (4C)	ADDRESS	4	DCBSETL	– ADDRESS OF SETL MODULE FOR QISAM. ADDRESS OF CHECK MODULE FOR BISAM
80 (50)	BITSTRING	1	DCBEXCD1	– FIRST BYTE IN WHICH EXCEPTIONAL CONDITIONS DETECTED IN PROCESSING DATA RECORDS ARE REPORTED TO THE USER
	1.		DCBEXNKY	″DCBBIT0″– LOWER KEY LIMIT NOT FOUND
	.1.		DCBEXIDA	″DCBBIT1″– INVALID DEVICE ADDRESS FOR LOWER LIMIT

Offset	Type/Value	Len	Name (Dim)	Description
	. .1.		DCBEXNSP	″DCBBIT2″– SPACE NOT FOUND
	. . .1		DCBEXINV	″DCBBIT3″– INVALID REQUEST
	. . . . 1. . .		DCBEXIER	″DCBBIT4″– UNCORRECTABLE INPUT ERROR
	. . . . .1. .		DCBEXOER	″DCBBIT5″– UNCORRECTABLE OUTPUT ERROR
	. . . . . .1.		DCBEXBLI	″DCBBIT6″– BLOCK COULD NOT BE REACHED (INPUT)
	1		DCBEXBLU	″DCBBIT7″– BLOCK COULD NOT BE REACHED (UPDATE)
81 (51)	BITSTRING	1	DCBEXCD2	– SECOND BYTE IN WHICH EXCEPTIONAL CONDITIONS DETECTED IN PROCESSING DATA RECORDS ARE REPORTED TO THE USER
	1.		DCBEXSEQ	″DCBBIT0″– SEQUENCE CHECK
	.1.		DCBEXDUP	″DCBBIT1″– DUPLICATE RECORD
	. .1.		DCBEXCLD	″DCBBIT2″– DCB CLOSED WHEN ERROR WAS DETECTED
	. . .1		DCBEXOFL	″DCBBIT3″– OVERFLOW RECORD
	. . . . 1. . .		DCBEXLTH	″DCBBIT4″– FOR PUT – LENGTH FIELD OF RECORD LARGER THAN LENGTH INDICATED IN DCBLRECL
	. . . . 1. . .		DCBEXRDE	″DCBBIT4″– READ EXCLUSIVE
82 (52)	SIGNED	2	DCBLRECL	– FOR FIXED–LENGTH RECORD FORMATS, LOGICAL RECORD LENGTH. FOR VARIABLE–LENGTH RECORD FORMATS, MAXIMUM LOGICAL RECORD LENGTH OR AN ACTUAL LOGICAL RECORD LENGTH CHANGED DYNAMICALLY BY USER WHEN CREATING THE DATA SET
84 (54)	ADDRESS	4	DCBESETL	– ADDRESS OF ESETL ROUTINE IN GET MODULE
88 (58)	ADDRESS	4	DCBLRAN	– ADDRESS OF READ–WRITE K MODULE OR EXCLUSIVE MODULE
92 (5C)	ADDRESS	4	DCBLWKN	– ADDRESS OF WRITE KN MODULE
96 (60)	ADDRESS	4	DCBRELSE	– WORK AREA FOR TEMPORARY STORAGE OF REGISTER CONTENTS
100 (64)	ADDRESS	4	DCBPUTX	– WORK AREA FOR TEMPORARY STORAGE OF REGISTER CONTENTS
104 (68)	ADDRESS	4	DCBRELEX	– ADDRESS OF READ EXCLUSIVE MODULE
108 (6C)	ADDRESS	4	DCBFREED	– ADDRESS OF DYNAMIC BUFFERING MODULE
112 (70)	SIGNED	1	DCBHIRTI	– NUMBER OF INDEX ENTRIES THAT FIT ON A PRIME DATA TRACK
113 (71)	CHARACTER	7	DCBFTMI2	– DIRECT ACCESS DEVICE ADDRESS OF FIRST TRACK OF SECOND LEVEL MASTER INDEX (IN THE FORM MBBCCHH)
120 (78)	CHARACTER	5	DCBLEMI2	– DIRECT ACCESS DEVICE ADDRESS OF LAST ACTIVE ENTRY IN SECOND LEVEL MASTER INDEX (IN THE FORM CCHHR)
125 (7D)	CHARACTER	7	DCBFTMI3	– DIRECT ACCESS DEVICE ADDRESS OF FIRST TRACK OF THIRD LEVEL MASTER INDEX (IN THE FORM MBBCCHH)
132 (84)	CHARACTER	5	DCBLEMI3	– DIRECT ACCESS DEVICE ADDRESS OF LAST ACTIVE ENTRY IN THIRD LEVEL MASTER INDEX (IN THE FORM CCHHR)
137 (89)	SIGNED	1	DCBNLEV	– NUMBER OF LEVELS OF INDEX
138 (8A)	CHARACTER	3	DCBFIRSH	– HHR OF FIRST DATA RECORD ON EACH CYLINDER. FOR VARIABLE LENGTH RECORD PROCESSING, R PORTION OF THIS FIELD IS ALWAYS X’01’.

Offset	Type/Value	Len	Name (Dim)	Description
141 (8D)	CHARACTER	1	DCBHMASK	– BYTE INDICATING 2301 OR NOT
	111		DCBHMDRM	″X'07'″– DEVICE IS 2301 DRUM
	1111 1111		DCBHMNDM	″X'FF'″– DEVICE IS OTHER THAN 2301 DRUM
142 (8E)	CHARACTER	2	DCBLDT	– HH IS THE LAST PRIME DATA TRACK ON EACH CYLINDER
144 (90)	CHARACTER	1	DCBHIRCM	– HIGHEST POSSIBLE R FOR TRACKS OF THE CYLINDER AND MASTER INDICES
145 (91)	CHARACTER	1	DCBHIRPD	– HIGHEST R ON ANY PRIME TRACK IN DATA SET. FOR VARIABLE –LENGTH RECORDS, THIS REPRESENTS THE GREATEST NUMBER OF PHYSICAL RECORDS ON ANY PRIME TRACK IN THE DATA SET
146 (92)	CHARACTER	1	DCBHIROV	– FOR FIXED–LENGTH RECORD FORMAT, HIGHEST POSSIBLE R FOR OVERFLOW DATA TRACKS. FOR VARIABLE-LENGTH RECORD FORMAT, UNUSED.
147 (93)	CHARACTER	1	DCBHIRSH	– FOR FIXED–LENGTH RECORD FORMAT, R OF LAST DATA RECORD ON A SHARED TRACK, IF APPLICABLE. FOR VARIABLE–LENGTH RECORD FORMAT, UNUSED.
148 (94)	SIGNED	2	DCBTDC	– USER–SUPPLIED NUMBER OF RECORDS TAGGED FOR DELETION.
150 (96)	SIGNED	2	DCBNCRHI	– NUMBER OF STORAGE LOCATIONS NEEDED TO HOLD THE HIGHEST LEVEL INDEX
152 (98)	SIGNED	4	DCBRORG3	– FOR EACH USE OF DATA SET, NUMBER OF READ OR WRITE ACCESSES TO AN OVER FLOW RECORD WHICH IS NOT FIRST IN A CHAIN OF SUCH RECORDS
156 (9C)	SIGNED	4	DCBNREC	– NUMBER OF LOGICAL RECORDS IN PRIME DATA AREA
160 (A0)	BITSTRING	1	DCBST	– STATUS INDICATORS
	1.		DCBSTSSM	″DCBBIT0″– SINGLE SCHEDULE MODE
	.1.		DCBSTKSQ	″DCBBIT1″– KEY SEQUENCE CHECKING IS TO BE PERFORMED
	. .1.		DCBSTLOD	″DCBBIT2″– LOADING HAS COMPLETED. SET TO 1 BY CLOSE ROUTINE AND TO 0 BY FIRST EXECUTION OF PUT ROUTINE.
	. . .1		DCBSTNCY	″DCBBIT3″– EXTENSION OF DATA SET WILL BEGIN ON NEW CYLINDER
	. . . . .1. .		DCBSTNMC	″DCBBIT5″– FIRST MACRO INSTRUCTION NOT YET RECEIVED
	. . . . . .1.		DCBSTLBF	″DCBBIT6″– LAST BLOCK FULL
	1		DCBSTLTF	″DCBBIT7″– LAST TRACK FULL
161 (A1)	CHARACTER	7	DCBFTCI	– DIRECT ACCESS DEVICE ADDRESS OF FIRST TRACK OF CYLINDER INDEX (IN THE FORM MBBCCHH).
168 (A8)	CHARACTER	1	DCBHIIOV	– FOR FIXED LENGTH RECORD FORMAT, HIGHEST POSSIBLE R FOR INDEPENDENT OVERFLOW DATA TRACKS. FOR VARIABLE LENGTH RECORD FORMAT, UNUSED
169 (A9)	CHARACTER	7	DCBFTMI1	– DIRECT ACCESS DEVICE ADDRESS OF FIRST TRACK OF FIRST LEVEL MASTER INDEX (IN THE FORM MBBCCHH).
176 (B0)	SIGNED	1	DCBNTHI	– NUMBER OF TRACKS OF HIGH–LEVEL INDEX

Offset	Type/Value	Len	Name (Dim)	Description
177 (B1)	CHARACTER	7	DCBFTHI	– DIRECT ACCESS DEVICE ADDRESS OF FIRST TRACK OF HIGHEST LEVEL INDEX (IN THE FORM MBBCCHH).
184 (B8)	CHARACTER	8	DCBLPDA	– DIRECT ACCESS DEVICE ADDRESS OF LAST PRIME DATA RECORD IN PRIME DATA AREA (IN THE FORM MBBCCHHR).
192 (C0)	CHARACTER	5	DCBLETI	– DIRECT ACCESS DEVICE ADDRESS OF LAST ACTIVE NORMAL ENTRY OF TRACK INDEX ON LAST ACTIVE CYLINDER (IN THE FORM CCHHR).
197 (C5)	CHARACTER	1	DCBOVDEV	– DEVICE TYPE FOR INDEPENDENT OVERFLOW
	THESE SAME MASKS APPLY TO DCBDEVT FOR ISAM DIRECT ACCESS			
	. . . . . .1.		DCBDVI01	″X'02'″– 2301 PARALLEL DRUM
	11		DCBDVI03	″X'03'″– 2303 SERIAL DRUM
	. . . . .1. .		DCBDVI02	″X'04'″– 2302 DISK STORAGE
	. . . . .1.1		DCBDVI05	″X'05'″– 2305 DRUM
	. . . . 1. . .		DCBDVI14	″X'08'″– 2314 DISK STORAGE FACILITY
	. . . . 1. .1		DCBDVI30	″X'09'″– 3330 DISK STORAGE FACILITY
198 (C6)	SIGNED	2	DCBNBOV	– FOR FIXED LENGTH RECORD FORMAT, RESERVED. FOR VARIABLE LENGTH RECORD FORMAT, IF THE INDEPENDENT OVERFLOW OPTION IS SELECTED, CONTAINS, IN BINARY, NUMBER OF BYTES LEFT ON CURRENT TRACK OF INDEPENDENT OVERFLOW AREA
200 (C8)	CHARACTER	5	DCBLECI	– DIRECT ACCESS DEVICE ADDRESS OF LAST ACTIVE ENTRY IN CYLINDER INDEX (IN THE FORM CCHHR).
205 (CD)	BITSTRING	1		– RESERVED
206 (CE)	SIGNED	2	DCBRORG2	– NUMBER OF TRACKS (PARTIALLY OR WHOLLY) REMAINING IN INDEPENDENT OVERFLOW AREA
208 (D0)	CHARACTER	5	DCBLEMI1	– DIRECT ACCESS DEVICE ADDRESS OF LAST ACTIVE ENTRY IN FIRST LEVEL MASTER INDEX (IN THE FORM CCHHR).
213 (D5)	BITSTRING	1		– RESERVED
214 (D6)	SIGNED	2	DCBNOREC	– NUMBER OF LOGICAL RECORDS IN AN OVERFLOW AREA
216 (D8)	CHARACTER	8	DCBLIOV	– DIRECT ACCESS DEVICE ADDRESS OF LAST AREA (IN THE FORM MBBCCHHR).
224 (E0)	SIGNED	2	DCBRORG1	– NUMBER OF CYLINDER OVERFLOW AREAS THAT ARE FULL
226 (E2)	BITSTRING	2		– RESERVED
228 (E4)	ADDRESS	4	DCBWKPT1	– POINTER TO WORK AREA OR TO CONSTRUCTED CHANNEL PROGRAM FOR WHICH SPACE IS OBTAINED BY GETMAIN MACRO INSTRUCTIONS ISSUED BY OPEN EXECUTORS
232 (E8)	ADDRESS	4	DCBWKPT2	– ADDITIONAL POINTER AS IN DCBWKPT1
236 (EC)	ADDRESS	4	DCBWKPT3	– ADDITIONAL POINTER AS IN DCBWKPT1
240 (F0)	ADDRESS	4	DCBWKPT4	– ADDITIONAL POINTER AS IN DCBWKPT1
244 (F4)	ADDRESS	4	DCBWKPT5	– ADDITIONAL POINTER AS IN DCBWKPT1
248 (F8)	ADDRESS	4	DCBWKPT6	– ADDITIONAL POINTER AS IN DCBWKPT1

Cross-Reference

Name	Hex Offset	Hex Value	Level
DCBACBM	1B	8	2
DCBBFA	20	3	2

Name	Hex Offset	Hex Value	Level
DCBBFAD	20	2	2
DCBBFAF1	20	1	2
DCBBFAF2	20	3	2
DCBBFALN	20		2
DCBBFT	20	70	2
DCBBFTA	20	60	2
DCBBFTE	20	10	2
DCBBFTEK	20		2
DCBBFTK	20	8	2
DCBBFTKD	20	8	2
DCBBFTKR	20	20	2
DCBBFTR	20	20	2
DCBBFTS	20	40	2
DCBBIT0	0	80	2
DCBBIT1	0	40	2
DCBBIT2	0	20	2
DCBBIT3	0	10	2
DCBBIT4	0	8	2
DCBBIT5	0	4	2
DCBBIT6	0	2	2
DCBBIT7	0	1	2
DCBBLKSI	3E		2
DCBBUFCA	15		2
DCBBUFCB	14		2
DCBBUFL	18		2
DCBBUFNO	14		2
DCBBXLRI	20	68	2
DCBCYLOF	37		2
DCBDDNAM	28		2
DCBDEBA	2D		2
DCBDEBAD	2C		2
DCBDEVT	11		2
DCBDSGCX	1A	10	2
DCBDSGDA	1A	20	2
DCBDSGGS	1B	80	2
DCBDSGIS	1A	80	2
DCBDSGPO	1A	2	2
DCBDSGPS	1A	40	2
DCBDSGTQ	1B	20	2
DCBDSGTR	1B	4	2
DCBDSGTX	1B	40	2
DCBDSGU	1A	1	2
DCBDSORG	1A		2
DCBDSRG1	1A		2
DCBDSRG2	1B		2
DCBDVI01	C5	2	2
DCBDVI02	C5	4	2
DCBDVI03	C5	3	2
DCBDVI05	C5	5	2
DCBDVI14	C5	8	2
DCBDVI30	C5	9	2
DCBDVTRM	11	4F	2
DCBEODA	21		2
DCBEODAD	20		2

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Name	Hex Offset	Hex Value	Level
DCBESETL	54		2
DCBEX	31	40	2
DCBEXBLI	50	2	2
DCBEXBLU	50	1	2
DCBEXCD1	50		2
DCBEXCD2	51		2
DCBEXCLD	51	20	2
DCBEXDUP	51	40	2
DCBEXIDA	50	40	2
DCBEXIER	50	8	2
DCBEXINV	50	10	2
DCBEXLSA	25		2
DCBEXLST	24		2
DCBEXLTH	51	8	2
DCBEXNKY	50	80	2
DCBEXNSP	50	20	2
DCBEXOER	50	4	2
DCBEXOFL	51	10	2
DCBEXRDE	51	8	2
DCBEXSEQ	51	80	2
DCBFIRSH	8A		2
DCBFREED	6C		2
DCBFTCI	A1		2
DCBFTHI	B1		2
DCBFTMI1	A9		2
DCBFTMI2	71		2
DCBFTMI3	7D		2
DCBGET	30		2
DCBGETA	31		2
DCBHIARC	20		2
DCBHIIOV	A8		2
DCBHIRCM	90		2
DCBHIROV	92		2
DCBHIRPD	91		2
DCBHIRSH	93		2
DCBHIRTI	70		2
DCBHMASK	8D		2
DCBHMDRM	8D	7	2
DCBHMNDM	8D	FF	2
DCBH0	20	4	2
DCBH1	20	80	2
DCBIBEC	31	C0	2
DCBIBIOE	31	C	2
DCBIBPCT	31	30	2
DCBICQE	1C		2
DCBICQEA	1D		2
DCBIFC12	31	10	2
DCBIFC9	31	20	2
DCBIFEC	2C	C0	2
DCBIFER	31	0	2
DCBIFIOE	2C	C	2
DCBIFLDT	2C	2	2
DCBIFLG	31		2
DCBIFLGS	2C		2

Name	Hex Offset	Hex Value	Level
DCBIFNEP	31	0	2
DCBIFNE1	31	4	2
DCBIFNE2	31	8	2
DCBIFNE3	31	C	2
DCBIFPCT	2C	30	2
DCBIFPEC	31	C0	2
DCBIFTIM	31	4	2
DCBIOBAA	1D		2
DCBIOBAD	1C		2
DCBKEYLE	10		2
DCBLDT	8E		2
DCBLECI	C8		2
DCBLEMI1	D0		2
DCBLEMI2	78		2
DCBLEMI3	84		2
DCBLETI	C0		2
DCBLIOV	D8		2
DCBLNGIS	F8	FC	2
DCBLNGXE	33	34	2
DCBLNP	1C		2
DCBLPDA	B8		2
DCBLPDT	40		2
DCBLRAN	58		2
DCBLRECL	52		2
DCBLWKN	5C		2
DCBMAC	35		2
DCBMACAW	35	2	2
DCBMACF1	2A		2
DCBMACF2	2B		2
DCBMACR	32		2
DCBMACRE	35	1	2
DCBMACRF	2A		2
DCBMACR1	32		2
DCBMACR2	33		2
DCBMACUR	35	8	2
DCBMACUW	35	4	2
DCBMRABC	32	4	2
DCBMRAPG	32	20	2
DCBMRAWR	33	2	2
DCBMRCHK	32	2	2
DCBMRCI	32	10	2
DCBMRCK	32	1	2
DCBMRCRL	32	2	2
DCBMRCTL	33	2	2
DCBMRDBF	32	4	2
DCBMRDMD	33	1	2
DCBMRDMG	32	1	2
DCBMRECP	32	80	2
DCBMRFE	32	40	2
DCBMRGET	32	40	2
DCBMRGTQ	33	40	2
DCBMRIDW	33	8	2
DCBMRLCG	32	8	2
DCBMRLCP	33	8	2

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Name	Hex Offset	Hex Value	Level
DCBMRLDM	33	8	2
DCBMRMVG	32	10	2
DCBMRMVP	33	10	2
DCBMRPTQ	32	40	2
DCBMRPT1	32	4	2
DCBMRPT2	33	4	2
DCBMRPUT	33	40	2
DCBMRRD	32	20	2
DCBMRRDI	32	8	2
DCBMRRDK	32	10	2
DCBMRRDQ	33	20	2
DCBMRRDX	32	2	2
DCBMRSBG	32	4	2
DCBMRSTI	33	1	2
DCBMRSTK	33	2	2
DCBMRSTL	33	80	2
DCBMRSWA	33	1	2
DCBMRTMD	33	4	2
DCBMRUIP	33	4	2
DCBMRWRK	33	10	2
DCBMRWRQ	32	20	2
DCBMRWRT	33	20	2
DCBMR1WD	33	1	2
DCBMR3WD	33	2	2
DCBMR4WD	33	4	2
DCBMR5WD	33	8	2
DCBMSHI	48		2
DCBMSHIA	49		2
DCBMSWA	40		2
DCBNBOV	C6		2
DCBNCP	48		2
DCBNCRHI	96		2
DCBNLEV	89		2
DCBNOREC	D6		2
DCBNREC	9C		2
DCBNTHI	B0		2
DCBNTM	36		2
DCBNUPD	1C	30	2
DCBODEB	1C		2
DCBODEBA	1D		2
DCBOFEOV	30	20	2
DCBOFIOD	30	80	2
DCBOFIOF	30	1	2
DCBOFLGS	30		2
DCBOFLG1	30		2
DCBOFLRB	30	40	2
DCBOFLWR	30	80	2
DCBOFOPN	30	10	2
DCBOFPPC	30	8	2
DCBOFTM	30	4	2
DCBOFUEX	30	2	2
DCBOPTCD	34		2
DCBOPTI	34	10	2
DCBOPTL	34	2	2

Name	Hex Offset	Hex Value	Level
DCBOPTM	34	20	2
DCBOPTR	34	1	2
DCBOPTUF	34	40	2
DCBOPTW	34	80	2
DCBOPTY	34	8	2
DCBOVDEV	C5		2
DCBPGFXA	32	2	2
DCBPUT	30		2
DCBPUTA	31		2
DCBPUTX	64		2
DCBQSLM	1C		2
DCBRECBR	24	10	2
DCBRECC	24	0	2
DCBRECCA	24	4	2
DCBRECCC	24	6	2
DCBRECCM	24	2	2
DCBRECD	24	20	2
DCBRECF	24	80	2
DCBRECFM	24		2
DCBRECKL	24	1	2
DCBRECL	24	C0	2
DCBRECLA	24	E0	2
DCBRECSB	24	8	2
DCBRECTO	24	20	2
DCBRECU	24	C0	2
DCBRECV	24	40	2
DCBREL	11		2
DCBRELB	10		2
DCBRELEX	68		2
DCBRELSE	60		2
DCBRKP	3C		2
DCBRORG1	E0		2
DCBRORG2	CE		2
DCBRORG3	98		2
DCBSETL	4C		2
DCBSMSI	44		2
DCBSMSW	46		2
DCBST	A0		2
DCBSTKSQ	A0	40	2
DCBSTLBF	A0	2	2
DCBSTLOD	A0	20	2
DCBSTLTF	A0	1	2
DCBSTNCY	A0	10	2
DCBSTNMC	A0	4	2
DCBSTSSM	A0	80	2
DCBSVCXA	1D		2
DCBSVCXL	1C		2
DCBSVDEB	1C	10	2
DCBSYNAD	38		2
DCBTDC	94		2
DCBTIOT	28		2
DCBUPDBT	1C	30	2
DCBUPDCM	1C	40	2
DCBUPDT	1C	20	2

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Name	Hex Offset	Hex Value	Level
DCBWKPT1	E4		2
DCBWKPT2	E8		2
DCBWKPT3	EC		2
DCBWKPT4	F0		2
DCBWKPT5	F4		2
DCBWKPT6	F8		2
DCB1DVDS	1C	80	2

DCB for BDAM

Common Name: Data Control Block for BDAM
Macro Name: DCBD
DSECT Name: IHADCB
Owning Component: Data Management, subcomponent OPEN/CLOSE/EOV
Eye-Catcher ID: None
Subpool and Key: Problem program subpool and key
Size: 104 bytes
Created by: Problem program
Pointed to by:

- DEBDCBAD field of the DEB data area
- IOBDCBPT field of the IOB data area
- DECDCBAD field of the DECB data area

Serialization: The user is responsible for serialization. While OPEN/CLOSE processes the DCB, a temporary protected copy of the DCB is made to serialize processing. Use the DEBCHK macro to check for the validity of the DCB.

Function: This data control block (DCB) contains information pertaining to the data sets being processed by the basic direct access method (BDAM) routines. All DCB formats share the same interface and foundation sections. The direct access storage device (DASD) section and the BDAM interface section complete the description of the block.

Offset	Type/Value	Len	Name (Dim)	Description
DATA CONTROL BLOCK DEFINITION				
	X'80'		DCBBIT0	″128″
	X'40'		DCBBIT1	″64″
	X'20'		DCBBIT2	″32″
	X'10'		DCBBIT3	″16″
	X'8'		DCBBIT4	″8″
	X'4'		DCBBIT5	″4″
	X'2'		DCBBIT6	″2″
	X'1'		DCBBIT7	″1″
	ACCESS METHOD COMMON INTERFACE			
16 (10)	SIGNED	4	DCBRELB (0)	–SAME AS DCBREL BELOW
16 (10)	SIGNED	1	DCBKEYLE	–KEY LENGTH OF DATA SET
17 (11)	CHARACTER	1	DCBDEVT (0)	– DEVICE TYPE
	.1. . 1111		DCBDVTRM	″X'4F'″– TERMINAL. (DD CONTAINS TERM=TS)
17 (11)	SIGNED	3	DCBREL	–NUMBER OF RELATIVE TRACKS OR BLOCKS IN THIS DATA SET (BDAM)

Offset	Type/Value	Len	Name (Dim)	Description
20 (14)	ADDRESS	4	DCBBUFCB (0)	–ADDRESS OF BUFFER POOL CONTROL BLOCK
20 (14)	SIGNED	1	DCBBUFNO	– NUMBER OF BUFFERS REQUIRED FOR THIS DATA SET. MAY RANGE FROM 0 TO 255. IF UNBLOCKED SPANNED RECORDS ARE USED, NUMBER OF SEGMENT WORK AREAS REQUIRED FOR THIS DATA SET.
21 (15)	ADDRESS	3	DCBBUFCA	–ADDRESS OF BUFFER POOL CONTROL BLOCK
24 (18)	SIGNED	2	DCBBUFL	–LENGTH OF BUFFER. MAY RANGE FROM 0 TO 32,767.
26 (1A)	BITSTRING	2	DCBDSORG (0)	–DATA SET ORGANIZATION BEING USED
26 (1A)	BITSTRING	1	DCBDSRG1	–FIRST BYTE OF DCBDSORG
	1.		DCBDSGIS	″DCBBIT0″– IS – INDEXED SEQUENTIAL ORGANIZATION
	.1.		DCBDSGPS	″DCBBIT1″– PS – PHYSICAL SEQUENTIAL ORGANIZATION
	. .1.		DCBDSGDA	″DCBBIT2″– DA – DIRECT ORGANIZATION
	. . .1		DCBDSGCX	″DCBBIT3″– CX – BTAM OR QTAM LINE GROUP
	. . . . . .1.		DCBDSGPO	″DCBBIT6″– PO – PARTITIONED ORGANIZATION
	1		DCBDSGU	″DCBBIT7″– U – UNMOVABLE, THE DATA CONTAINS LOCATION DEPENDENT INFORMATION
27 (1B)	BITSTRING	1	DCBDSRG2	–BYTE OF DCBDSORG
	1.		DCBDSGGS	″DCBBIT0″– GS – GRAPHICS ORGANIZATION
	.1.		DCBDSGTX	″DCBBIT1″– TX – TCAM LINE GROUP
	. .1.		DCBDSGTQ	″DCBBIT2″– TQ – TCAM MESSAGE QUEUE
	. . . . 1. . .		DCBACBM	″DCBBIT4″– ACCESS METHOD CONTROL BLOCK
	. . . . .1. .		DCBDSGTR	″DCBBIT5″– TR – TCAM 3705
28 (1C)	ADDRESS	4	DCBIOBAD (0)	–ADDRESS OF IOB WHEN CHAINED SCHEDULING IS USED OR FOR 1419/1275
28 (1C)	ADDRESS	4	DCBICQE (0)	–ADDRESS OF ICQE
28 (1C)	ADDRESS	4	DCBODEB (0)	–ADDRESS OF OLD DEB
28 (1C)	SIGNED	1	DCBLNP (0)	–3525 PRINTER LINE POSITION COUNTER
28 (1C)	BITSTRING	1	DCBQSLM	–QSAM LOCATE MODE LOGICAL RECORD INTERFACE INDICATOR BYTE FOR UPDAT PROCESSING OF SPANNED RECORDS
	1.		DCB1DVDS	″DCBBIT0″– ONLY ONE DEVICE IS ALLOCATED TO THIS DATA SET
	.1.		DCBUPDCM	″DCBBIT1″– UPDATE COMPLETE, FREE OLD DEB
	. .11		DCBUPDBT	″DCBBIT2+DCBBIT3″– UPDATE BITS
	. .1.		DCBUPDT	″DCBBIT2″– UPDATE TO TAKE PLACE
	. .11		DCBNUPD	″DCBBIT2+DCBBIT3″– NO UPDATE TO TAKE PLACE
	. . .1		DCBSVDEB	″DCBBIT3″– OLD DEB ADDRESS MUST BE SAVED
29 (1D)	ADDRESS	3	DCBIOBAA (0)	– SAME AS DCBIOBAD ABOVE
29 (1D)	ADDRESS	3	DCBICQEA (0)	– SAME AS DCBICQE ABOVE
29 (1D)	ADDRESS	3	DCBODEBA	– ADDRESS OF OLD DEB
28 (1C)	ADDRESS	4	DCBSVCXL (0)	– SAME AS DCBSVCXA BELOW
28 (1C)	BITSTRING	1		– RESERVED
29 (1D)	ADDRESS	3	DCBSVCXA	– POINTER TO EXIT LIST OF JES C.I. INTERFACE CONTROL SVC
	FOUNDATION EXTENSION			
32 (20)	ADDRESS	4	DCBEODAD (0)	– SAME AS DCBEODA BELOW
32 (20)	BITSTRING	1	DCBHIARC (0)	– HIERARCHY BITS
32 (20)	BITSTRING	1	DCBBFTEK (0)	– BUFFERING TECHNIQUE BITS
32 (20)	BITSTRING	1	DCBBFALN	– BUFFER ALIGNMENT BITS

Offset	Type/Value	Len	Name (Dim)	Description
	1.		DCBH1	″DCBBIT0″– HIERARCHY 1 MAIN STORAGE – BIT 5 IS ZERO
	.111		DCBBFT	″DCBBIT1+DCBBIT2+DCBBIT3″ BUFFERING TECHNIQUE
	.11.		DCBBFTA	″DCBBIT1+DCBBIT2″ – QSAM LOCATE MODE PROCESSING OF SPANNED RECORDS – OPEN IS TO CONSTRUCT A RECORD AREA IF IT AUTOMATICALLY CONSTRUCTS BUFFERS
	. .1.		DCBBFTR	″DCBBIT2″– FOR BSAM CREATE BDAM PROCESSING OF UNBLOCKED SPANNED RECORDS – SOFTWARE TRACK OVERFLOW. FOR BSAM INPUT PROCESSING OF UNBLOCKED SPANNED RECORDS WITH KEYS – RECORD OFFSET PROCESSING.
	.1.		DCBBFTS	″DCBBIT1″– SIMPLE BUFFERING – BIT 3 IS ZERO
	. .1.		DCBBFTKR	″DCBBIT2″– UNBLOCKED SPANNED RECORDS – SOFTWARE TRACK OVERFLOW (BDAM)
	. . .1		DCBBFTE	″DCBBIT3″– EXCHANGE BUFFERING – BIT 1 IS ZERO
	. . . . 1. . .		DCBBFTKD	″DCBBIT4″– DYNAMIC BUFFERING (BTAM)
	. . . . 1. . .		DCBBFTK	″DCBBIT4″– LRECL IN ’K’ UNITS FOR XLRI
	.11. 1. . .		DCBBXLRI	″DCBBIT1+DCBBIT2+DCBBIT4″ – EXTENDED LOGICAL RECORD INTERFACE MODE FOR SPANNED RECORDS GREATER THAN 32K – ALL RECORDS TRANSFERRED TO OR FROM THE RECORD AREA.
	. . . . .1. .		DCBH0	″DCBBIT5″– HIERARCHY 0 MAIN STORAGE – BIT 0 IS ZERO
	11		DCBBFA	″DCBBIT6+DCBBIT7″– BUFFER ALIGNMENT
	. . . . . .1.		DCBBFAD	″DCBBIT6″– DOUBLEWORD BOUNDARY
	1		DCBBFAF1	″DCBBIT7″– FULLWORD NOT A DOUBLEWORD BOUNDARY, CODED IN DCB MACRO INSTRUCTION
	11		DCBBFAF2	″DCBBIT6+DCBBIT7″– FULLWORD NOT A DOUBLEWORD BOUNDARY, CODED IN DCB MACRO INSTRUCTION
33 (21)	ADDRESS	3	DCBEODA	– ADDRESS OF A USER–PROVIDED ROUTINE TO HANDLE END–OF7–DATA CONDITIONS
36 (24)	ADDRESS	4	DCBEXLST(0)	– ADDRESS OF USER-PROVIDED LIST OF EXITS
36 (24)	BITSTRING	1	DCBRECFM	– RECORD FORMAT
	111.		DCBRECLA	″DCBBIT0+DCBBIT1+DCBBIT2″ RECORD LENGTH INDICATOR – ASCII
	. .1.		DCBRECD	″DCBBIT2″– ASCII VARIABLE RECORD LENGTH
	11.		DCBRECL	″DCBBIT0+DCBBIT1″– RECORD LENGTH INDICATOR
	1.		DCBRECF	″DCBBIT0″– FIXED RECORD LENGTH
	.1.		DCBRECV	″DCBBIT1″– VARIABLE RECORD LENGTH
	11.		DCBRECU	″DCBBIT0+DCBBIT1″– UNDEFINED RECORD LENGTH
	. .1.		DCBRECTO	″DCBBIT2″– TRACK OVERFLOW
	. . .1		DCBRECBR	″DCBBIT3″– BLOCKED RECORDS
	. . . . 1. . .		DCBRECSB	″DCBBIT4″–FOR FIXED LENGTH RECORD FORMAT – STANDARD BLOCKS. FOR VARIABLE LENGTH RECORD FORMAT – SPANNED RECORDS
	. . . . .11.		DCBRECCC	″DCBBIT5+DCBBIT6″– CONTROL CHARACTER INDICATOR

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . .1. .		DCBRECCA	″DCBBIT5″– ASA CONTROL CHARACTER
	. . . . . .1.		DCBRECCM	″DCBBIT6″– MACHINE CONTROL CHARACTER
	. . . . .		DCBRECC	″X'00'″– NO CONTROL CHARACTER
	1		DCBRECKL	″DCBBIT7″– KEY LENGTH (KEYLEN) WAS SPECIFIED IN DCB MACRO INSTRUCTION
37 (25)	ADDRESS	3	DCBEXLSA	– ADDRESS OF USER-PROVIDED LIST OF EXITS
	FOUNDATION BEFORE OPEN			
40 (28)	CHARACTER	8	DCBDDNAM	– NAME ON THE DD STATEMENT WHICH DEFINES THE DATA SET ASSOCIATED WITH THIS DCB
48 (30)	BITSTRING	1	DCBOFLGS	– FLAGS USED BY OPEN ROUTINE
	1.		DCBOFLWR	″DCBBIT0″– IF ZERO, LAST I/O OPERATION WAS READ OR POINT. IF ONE, LAST I/O OPERATION WAS WRITE.
	1.		DCBOFIOD	″DCBBIT0″– DATA SET IS BEING OPENED FOR INPUT OR OUTPUT (BDAM)
	.1.		DCBOFLRB	″DCBBIT1″– LAST I/O OPERATION WAS IN READ BACKWARD MODE
	. .1.		DCBOFEOV	″DCBBIT2″– SET TO 1 BY EOV WHEN IT CALLS CLOSE ROUTINE FOR CONCATENATION OF DATA SETS WITH UNLIKE ATTRIBUTES
	. . .1		DCBOFOPN	″DCBBIT3″– AN OPEN HAS BEEN SUCCESSFULLY COMPLETED
	. . . . 1. . .		DCBOFPPC	″DCBBIT4″– SET TO 1 BY PROBLEM PROGRAM TO INDICATE A CONCATENATION OF UNLIKE ATTRIBUTES
	. . . . .1. .		DCBOFTM	″DCBBIT5″– TAPE MARK HAS BEEN READ
	. . . . . .1.		DCBOFUEX	″DCBBIT6″– SET TO 0 BY AN I/O SUPPORT FUNCTION WHEN THAT FUNCTION TAKES A USER EXIT. SET TO 1 ON RETURN FROM USER EXIT TO THE I/O SUPPORT FUNCTION WHICH TOOK THE EXIT.
	1		DCBOFIOF	″DCBBIT7″– SET TO 1 BY AN I/O SUPPORT FUNCTION IF DCB IS TO BE PROCESSED BY THAT FUNCTION
49 (31)	BITSTRING	1	DCBIFLG	– FLAGS USED BY IOS IN COMMUNICATING ERROR CONDITIONS AND IN DETERMINING CORRECTIVE PROCEDURES
	11.		DCBIBEC	″DCBBIT0+DCBBIT1″– ERROR CORRECTION INDICATOR
	. . . . .		DCBIFNEP	″X'00'″– NOT IN ERROR PROCEDURE
	.1.		DCBEX	″DCBBIT1″– ERROR CORRECTION OR IOS PAGE FIX IN PROCESS
	11.		DCBIFPEC	″DCBBIT0+DCBBIT1″– PERMANENT ERROR CORRECTION
	. .11		DCBIBPCT	″DCBBIT2+DCBBIT3″– PRINTER CARRIAGE TAPE PUNCH INDICATOR
	. .1.		DCBIFC9	″DCBBIT2″– CHANNEL 9 PRINTER CARRIAGE TAPE PUNCH SENSED
	. . .1		DCBIFC12	″DCBBIT3″– CHANNEL 12 PRINTER CARRIAGE TAPE PUNCH SENSED
	. . . . 11. .		DCBIBIOE	″DCBBIT4+DCBBIT5″– IOS ERROR ROUTINE USE INDICATOR
	. . . . .		DCBIFER	″X'00'″– ALWAYS USE I/O SUPERVISOR ERROR ROUTINE
	. . . . .1. .		DCBIFNE1	″DCBBIT5″– NEVER USE I/O SUPERVISOR ERROR ROUTINE

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . .1. .		DCBIFTIM	″DCBBIT5″– TEST IOS MASK (IMSK) FOR ERROR PROCEDURE (BTAM)
	. . . . 1. . .		DCBIFNE2	″DCBBIT4″– NEVER USE I/O SUPERVISOR ERROR ROUTINE
	. . . . 11. .		DCBIFNE3	″DCBBIT4+DCBBIT5″– NEVER USE I/O SUPERVISOR ERROR ROUTINE
50 (32)	BITSTRING	2	DCBMACR (0)	– MACRO INSTRUCTION REFERENCE
50 (32)	BITSTRING	1	DCBMACR1	– FIRST BYTE OF DCBMACR
	1.		DCBMRECP	″DCBBIT0″– EXECUTE CHANNEL PROGRAM (EXCP) --- ALWAYS ZERO (BSAM, QSAM, BPAM, BISAM, QISAM, BDAM) --- RESERVED (QTAM, BTAM)
	.1.		DCBMRFE	″DCBBIT1″– FOUNDATION EXTENSION IS PRESENT (EXCP)
	.1.		DCBMRGET	″DCBBIT1″– GET (QSAM, QISAM, TCAM)
	.1.		DCBMRPTQ	″DCBBIT1″– PUT FOR MESSAGE GROUP (QTAM) --- ALWAYS ZERO (BSAM, BPAM, BISAM, BDAM) --- RESERVED (BTAM)
	. .1.		DCBMRAPG	″DCBBIT2″– APPENDAGES ARE REQUIRED (EXCP)
	. .1.		DCBMRRD	″DCBBIT2″– READ (BSAM, BPAM, BISAM, BDAM, BTAM)
	. .1.		DCBMRWRQ	″DCBBIT2″– WRITE FOR LINE GROUP (QTAM) --- ALWAYS ZERO (QSAM, QISAM)
	. .1		DCBMRCI	″DCBBIT3″– COMMON INTERFACE (EXCP)
	. . .1		DCBMRMVG	″DCBBIT3″– MOVE MODE OF GET (QSAM, QISAM)
	. . .1		DCBMRRDK	″DCBBIT3″– KEY SEGMENT WITH READ (BDAM) --- ALWAYS ZERO (BISAM) --- RESERVED (BSAM, BPAM, QTAM, BTAM)
	. . . . 1. . .		DCBMRLCG	″DCBBIT4″– LOCATE MODE OF GET (QSAM, QISAM)
	. . . . 1. . .		DCBMRRDI	″DCBBIT4″– ID ARGUMENT WITH READ (BDAM) --- ALWAYS ZERO (BISAM) --- RESERVED (EXCP, BSAM, BPAM, QTAM, BTAM)
	. . . . .1. .		DCBMRABC	″DCBBIT5″– USER’S PROGRAM MAINTAINS ACCURATE BLOCK COUNT (EXCP)
	. . . . .1. .		DCBMRPT1	″DCBBIT5″– POINT (WHICH IMPLIES NOTE) (BSAM, BPAM)
	. . . . .1. .		DCBMRSBG	″DCBBIT5″– SUBSTITUTE MODE OF GET (QSAM)
	. . . . .1. .		DCBMRDBF	″DCBBIT5″– DYNAMIC BUFFERING (BISAM, BDAM) --- ALWAYS ZERO (QISAM) --- RESERVED (QTAM, BTAM)
	. . . . . .1.		DCBPGFXA	″DCBBIT6″– PAGE FIX APPENDAGE IS SPECIFIED (EXCP)
	. . . . . .1.		DCBMRCRL	″DCBBIT6″– CNTRL (BSAM, QSAM)
	. . . . . .1.		DCBMRCHK	″DCBBIT6″– CHECK (BISAM)
	. . . . . .1.		DCBMRRDX	″DCBBIT6″– READ EXCLUSIVE (BDAM) --- RESERVED (BPAM, QISAM, QTAM, BTAM)
	1		DCBMRDMG	″DCBBIT7″– DATA MODE OF GET (QSAM)
	1		DCBMRCK	″DCBBIT7″– CHECK (BDAM) --- RESERVED (EXCP, BSAM, BPAM, BISAM, QISAM, QTAM, BTAM)
51 (33)	BITSTRING	1	DCBMACR2	– SECOND BYTE OF DCBMACR
	1.		DCBMRSTL	″DCBBIT0″– SETL (QISAM) --- ALWAYS ZERO (BSAM, QSAM, BPAM, BISAM, BDAM) --- RESERVED (EXCP, QTAM, BTAM)
	.1.		DCBMRPUT	″DCBBIT1″– PUT (QSAM, TCAM) – PUT OR PUTX (QISAM)

Offset	Type/Value	Len	Name (Dim)	Description
	.1.		DCBMRGTQ	″DCBBIT1″– GET FOR MESSAGE GROUP (QTAM) --- ALWAYS ZERO (BSAM, BPAM, BISAM, BDAM) --- RESERVED (EXCP, BTAM)
	. .1.		DCBMRWRT	″DCBBIT2″– WRITE (BSAM, BPAM, BISAM, BDAM, BTAM)
	. .1.		DCBMRRDQ	″DCBBIT2″– READ FOR LINE GROUP (QTAM) --- ALWAYS ZERO (QSAM, QISAM) --- RESERVED (EXCP)
	. . .1		DCBMRMVP	″DCBBIT3″– MOVE MODE OF PUT (QSAM, QISAM)
	. . .1		DCBMRWRK	″DCBBIT3″– KEY SEGMENT WITH WRITE (BDAM) --- ALWAYS ZERO (BISAM) --- RESERVED (EXCP, BSAM, BPAM, QTAM, BTAM)
	. . . . 1. . .		DCBMR5WD	″DCBBIT4″– FIVE-WORD DEVICE INTERFACE (EXCP)
	. . . . 1. . .		DCBMRLDM	″DCBBIT4″– LOAD MODE BSAM (CREATE BDAM DATA SET) (BSAM)
	. . . . 1. . .		DCBMRLCP	″DCBBIT4″– LOCATE MODE OF PUT (QSAM, QISAM)
	. . . . 1. . .		DCBMRIDW	″DCBBIT4″– ID ARGUMENT WITH WRITE (BDAM) --- ALWAYS ZERO (BISAM) --- RESERVED (BPAM, QTAM, BTAM)
	. . . . .1. .		DCBMR4WD	″DCBBIT5″– FOUR–WORD DEVICE INTERFACE (EXCP)
	. . . . .1. .		DCBMRPT2	″DCBBIT5″– POINT (WHICH IMPLIES NOTE) (BSAM, BPAM)
	. . . . .1. .		DCBMRTMD	″DCBBIT5″– SUBSTITUTE MODE (QSAM)
	. . . . .1. .		DCBMRUIP	″DCBBIT5″– UPDATE IN PLACE (PUTX) (QISAM) --- ALWAYS ZERO (BISAM) --- RESERVED (BDAM, QTAM, BTAM)
	. . . . . .1.		DCBMR3WD	″DCBBIT6″– THREE–WORD DEVICE INTERFACE (EXCP)
	. . . . . .1.		DCBMRCTL	″DCBBIT6″– CNTRL (BSAM, QSAM)
	. . . . . .1.		DCBMRSTK	″DCBBIT6″– SETL BY KEY (QISAM)
	. . . . . .1.		DCBMRAWR	″DCBBIT6″– ADD TYPE OF WRITE (BDAM) --- ALWAYS ZERO (BISAM) --- RESERVED (BPAM, QTAM, BTAM)
	1		DCBMR1WD	″DCBBIT7″– ONE–WORD DEVICE INTERFACE (EXCP)
	1		DCBMRSWA	″DCBBIT7″– USER'S PROGRAM HAS PROVIDED A SEGMENT WORK AREA POOL (BSAM CREATE BDAM, BDAM)
	1		DCBMRDMD	″DCBBIT7″- DATA MODE (QSAM)
	1		DCBMRSTI	″DCBBIT7″– SETL BY ID (QISAM) — ALWAYS ZERO (BISAM) — RESERVED (BPAM, QTAM, BTAM)
		FOUNDATION	AFTER OPEN	
40 (28)	ADDRESS	2	DCBTIOT	OFFSET FROM TIOT ORIGIN TO TIOELNGH FIELD IN TIOT ENTRY FOR DD STATEMENT ASSOCIATED WITH THIS DCB
42 (2A)	BITSTRING	2	DCBMACRF (0)	– SAME AS DCBMACR BEFORE OPEN
42 (2A)	BITSTRING	1	DCBMACF1	– FIRST BYTE OF DCBMACRF
43 (2B)	BITSTRING	1	DCBMACF2	– SECOND BYTE OF DCBMACRF
44 (2C)	ADDRESS	4	DCBDEBAD (0)	– ADDRESS OF ASSOCIATED DEB
44 (2C)	BITSTRING	1	DCBIFLGS	– SAME AS DCBIFLG BEFORE OPEN
	11.		DCBIFEC	″DCBBIT0+DCBBIT1″– ERROR CORRECTION INDICATOR

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Offset	Type/Value	Len	Name (Dim)	Description
	. .11		DCBIFPCT	″DCBBIT2+DCBBIT3″– PRINTER CARRIAGE TAPE PUNCH INDICATOR
	. . . . 11. .		DCBIFIOE	″DCBBIT4+DCBBIT5″– RROR ROUTINE USE INDICATOR
	. . . . . .1.		DCBIFLDT	″DCBBIT6″– POSSIBLE LOST DATA CONDITION SUCH AS FOR A PRINTER
45 (2D)	ADDRESS	3	DCBDEBA	– ADDRESS OF ASSOCIATED DEB
48 (30)	ADDRESS	4	DCBREAD (0)	– ADDRESS OF READ MODULE
48 (30)	ADDRESS	4	DCBWRITE (0)	– ADDRESS OF WRITE MODULE
48 (30)	BITSTRING	1	DCBOFLG	SAME AS DCBOFLGS BEFORE OPEN
49 (31)	ADDRESS	3	DCBREADA (0)	ADDRESS OF READ MODULE
49 (31)	ADDRESS	3	DCBWRITA	ADDRESS OF WRITE MODULE
	BDAM INTERFACE			
52 (34)	ADDRESS	4	DCBCHECK (0)	– ADDRESS OF CHECK MODULE
52 (34)	BITSTRING	1	DCBOPTCD	– OPTION CODES
	1.		DCBOPTW	″DCBBIT0″– WRITE VALIDITY CHECK (DASD) (BSAM, BPAM, QSAM, ISAM, BDAM)
	.1.		DCBOPTTO	″DCBBIT1″– TRACK OVERFLOW
	. .1.		DCBOPTE	″DCBBIT2″– EXTENDED SEARCH
	. . .1		DCBOPTF	″DCBBIT3″– FEEDBACK
	. . . . 1. . .		DCBOPTA	″DCBBIT4″– ACTUAL ADDRESSING
	. . . . .1. .		DCBOPTDB	″DCBBIT5″– DYNAMIC BUFFERING
	. . . . . .1.		DCBOPTRE	″DCBBIT6″– READ EXCLUSIVE
	1		DCBOPTRB	″DCBBIT7″– RELATIVE BLOCK ADDRESSING
53 (35)	ADDRESS	3	DCBCHCKA	– ADDRESS OF CHECK MODULE
56 (38)	ADDRESS	4	DCBSYNAD	– ADDRESS OF SYNAD ROUTINE
60 (3C)	BITSTRING	2		– RESERVED
62 (3E)	SIGNED	2	DCBBLKSI	– MAXIMUM BLOCK SIZE
64 (40)	ADDRESS	4	DCBIOBSQ	– ADDRESS OF FIRST IOB ON UNSCHEDULED QUEUE FOR EITHER A WRITE–ADD REQUEST WHEN ANOTHER WRITE– ADD IS IN PROGRESS OR A READ–EXCLUSIVE REQUEST WHEN THE READ–EXCLUSIVE LIST IS FULL
68 (44)	ADDRESS	4	DCBSQND	– ADDRESS OF LAST IOB ON UNSCHEDULED QUEUE
72 (48)	ADDRESS	4	DCBIOBUQ	– ADDRESS OF FIRST IOB ON UNPOSTED QUEUE
76 (4C)	ADDRESS	4	DCBUQND	– ADDRESS OF LAST JOB ON UNPOSTED QUEUE THAT IS MAINTAINED BY THE READ EXCLUSIVE MODULE
80 (50)	BITSTRING	1		– RESERVED
81 (51)	SIGNED	3	DCBLIMCT	– NUMBER OF TRACKS OR NUMBER OF RELATIVE BLOCKS TO BE SEARCHED (EXTENDED SEARCH OPTION)
84 (54)	ADDRESS	4	DCBXARG (0)	– ADDRESS OF READ EXCLUSIVE LIST
84 (54)	SIGNED	1	DCBXCNT	– NUMBER OF ENTRIES IN READ EXCLUSIVE LIST
85 (55)	ADDRESS	3	DCBXARGA	– ADDRESS OF READ EXCLUSIVE LIST
88 (58)	ADDRESS	4	DCBDRDX (0)	– ADDRESS OF READ EXCLUSIVE MODULE
88 (58)	SIGNED	1	DCBMVXNO	– TOTAL NUMBER OF EXTENTS IN MULTIVOLUME DATA SET
89 (59)	ADDRESS	3	DCBDRDXA	– ADDRESS OF READ EXCLUSIVE MODULE
92 (5C)	ADDRESS	4	DCBDFOR	– ADDRESS OF A FORMAT MODULE
96 (60)	ADDRESS	4	DCBDFBK	– ADDRESS OF A FEEDBACK MODULE

Offset	Type/Value	Len	Name (Dim)	Description
100 (64)	ADDRESS	4	DCBDYNB	– FOR DYNAMIC BUFFERING, ADDRESS OF DYNAMIC BUFFER MODULE. FOR UNBLOCKED SPANNED RECORDS WITH BFTEK=R SPECIFIED AND NO DYNAMIC BUFFERING, ADDRESS OF SEGMENT WORK AREA CONTROL BLOCK

Cross-Reference

Name	Hex Offset	Hex Value	Level
DCBACBM	1B	8	2
DCBBFA	20	3	2
DCBBFAD	20	2	2
DCBBFAF1	20	1	2
DCBBFAF2	20	3	2
DCBBFALN	20		2
DCBBFT	20	70	2
DCBBFTA	20	60	2
DCBBFTE	20	10	2
DCBBFTEK	20		2
DCBBFTK	20	8	2
DCBBFTKD	20	8	2
DCBBFTKR	20	20	2
DCBBFTR	20	20	2
DCBBFTS	20	40	2
DCBBIT0	0	80	2
DCBBIT1	0	40	2
DCBBIT2	0	20	2
DCBBIT3	0	10	2
DCBBIT4	0	8	2
DCBBIT5	0	4	2
DCBBIT6	0	2	2
DCBBIT7	0	1	2
DCBBLKSI	3E		2
DCBBUFCA	15		2
DCBBUFCB	14		2
DCBBUFL	18		2
DCBBUFNO	14		2
DCBBXLRI	20	68	2
DCBCHCKA	35		2
DCBCHECK	34		2
DCBDDNAM	28		2
DCBDEBA	2D		2
DCBDEBAD	2C		2
DCBDEVT	11		2
DCBDFBK	60		2
DCBDFOR	5C		2
DCBDRDX	58		2
DCBDRDXA	59		2
DCBDSGCX	1A	10	2
DCBDSGDA	1A	20	2
DCBDSGGS	1B	80	2
DCBDSGIS	1A	80	2
DCBDSGPO	1A	2	2

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Name	Hex Offset	Hex Value	Level
DCBDSGPS	1A	40	2
DCBDSGTQ	1B	20	2
DCBDSGTR	1B	4	2
DCBDSGTX	1B	40	2
DCBDSGU	1A	1	2
DCBDSORG	1A		2
DCBDSRG1	1A		2
DCBDSRG2	1B		2
DCBDVTRM	11	4F	2
DCBDYNB	64		2
DCBEODA	21		2
DCBEODAD	20		2
DCBEX	31	40	2
DCBEXLSA	25		2
DCBEXLST	24		2
DCBHIARC	20		2
DCBH0	20	4	2
DCBH1	20	80	2
DCBIBEC	31	C0	2
DCBIBIOE	31	C	2
DCBIBPCT	31	30	2
DCBICQE	1C		2
DCBICQEA	1D		2
DCBIFC12	31	10	2
DCBIFC9	31	20	2
DCBIFEC	2C	C0	2
DCBIFER	31	0	2
DCBIFIOE	2C	C	2
DCBIFLDT	2C	2	2
DCBIFLG	31		2
DCBIFLGS	2C		2
DCBIFNEP	31	0	2
DCBIFNE1	31	4	2
DCBIFNE2	31	8	2
DCBIFNE3	31	C	2
DCBIFPCT	2C	30	2
DCBIFPEC	31	C0	2
DCBIFTIM	31	4	2
DCBIOBAA	1D		2
DCBIOBAD	1C		2
DCBIOBSQ	40		2
DCBIOBUQ	48		2
DCBKEYLE	10		2
DCBLIMCT	51		2
DCBLNGDA	64	68	2
DCBLNGXE	33	34	2
DCBLNP	1C		2
DCBMACF1	2A		2
DCBMACF2	2B		2
DCBMACR	32		2
DCBMACRF	2A		2
DCBMACR1	32		2
DCBMACR2	33		2
DCBMRABC	32	4	2

Name	Hex Offset	Hex Value	Level
DCBMRAPG	32	20	2
DCBMRAWR	33	2	2
DCBMRCHK	32	2	2
DCBMRCI	32	10	2
DCBMRCK	32	1	2
DCBMRCRL	32	2	2
DCBMRCTL	33	2	2
DCBMRDBF	32	4	2
DCBMRDMD	33	1	2
DCBMRDMG	32	1	2
DCBMRECP	32	80	2
DCBMRFE	32	40	2
DCBMRGET	32	40	2
DCBMRGTQ	33	40	2
DCBMRIDW	33	8	2
DCBMRLCG	32	8	2
DCBMRLCP	33	8	2
DCBMRLDM	33	8	2
DCBMRMVG	32	10	2
DCBMRMVP	33	10	2
DCBMRPTQ	32	40	2
DCBMRPT1	32	4	2
DCBMRPT2	33	4	2
DCBMRPUT	33	40	2
DCBMRRD	32	20	2
DCBMRRDI	32	8	2
DCBMRRDK	32	10	2
DCBMRRDQ	33	20	2
DCBMRRDX	32	2	2
DCBMRSBG	32	4	2
DCBMRSTI	33	1	2
DCBMRSTK	33	2	2
DCBMRSTL	33	80	2
DCBMRSWA	33	1	2
DCBMRTMD	33	4	2
DCBMRUIP	33	4	2
DCBMRWRK	33	10	2
DCBMRWRQ	32	20	2
DCBMRWRT	33	20	2
DCBMR1WD	33	1	2
DCBMR3WD	33	2	2
DCBMR4WD	33	4	2
DCBMR5WD	33	8	2
DCBMVXNO	58		2
DCBNUPD	1C	30	2
DCBODEB	1C		2
DCBODEBA	1D		2
DCBOFEOV	30	20	2
DCBOFIOD	30	80	2
DCBOFIOF	30	1	2
DCBOFLG	30		2
DCBOFLGS	30		2
DCBOFLRB	30	40	2
DCBOFLWR	30	80	2

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Name	Hex Offset	Hex Value	Level
DCBOFOPN	30	10	2
DCBOFPPC	30	8	2
DCBOFTM	30	4	2
DCBOFUEX	30	2	2
DCBOPTA	34	8	2
DCBOPTCD	34		2
DCBOPTDB	34	4	2
DCBOPTE	34	20	2
DCBOPTF	34	10	2
DCBOPTRB	34	1	2
DCBOPTRE	34	2	2
DCBOPTTO	34	40	2
DCBOPTW	34	80	2
DCBPGFXA	32	2	2
DCBQSLM	1C		2
DCBREAD	30		2
DCBREADA	31		2
DCBRECBR	24	10	2
DCBRECC	24	0	2
DCBRECCA	24	4	2
DCBRECCC	24	6	2
DCBRECCM	24	2	2
DCBRECD	24	20	2
DCBRECF	24	80	2
DCBRECFM	24		2
DCBRECKL	24	1	2
DCBRECL	24	C0	2
DCBRECLA	24	E0	2
DCBRECSB	24	8	2
DCBRECTO	24	20	2
DCBRECU	24	C0	2
DCBRECV	24	40	2
DCBREL	11		2
DCBRELB	10		2
DCBSQND	44		2
DCBSVCXA	1D		2
DCBSVCXL	1C		2
DCBSVDEB	1C	10	2
DCBSYNAD	38		2
DCBTIOT	28		2
DCBUPDBT	1C	30	2
DCBUPDCM	1C	40	2
DCBUPDT	1C	20	2
DCBUQND	4C		2
DCBWRITA	31		2
DCBWRITE	30		2
DCBXARG	54		2
DCBXARGA	55		2
DCBXCNT	54		2
DCB1DVDS	1C	80	2

DCB for BTAM and QTAM

Common Name: Data Control Block for BTAM and QTAM
Macro Name: DCBD
DSECT Name: IHADCB
Owning Component: Data Management, subcomponent OPEN/CLOSE/EOV
Eye-Catcher ID: None
Subpool and Key: Problem program subpool and key
Size: Variable up to 100 bytes
Created by: Problem program
Pointed to by:

- DEBDCBAD field of the DEB data area
- IOBDCBPT field of the IOB data area
- DECDCBAD field of the DECB data area (BDAM, BSAM, and BTAM DCBs)

Serialization: None
Function: This data control block (DCB) describes data sets being processed by the basic telecommunications access method (BTAM) and data sets being processed by the queued telecommunications access method (QTAM). All DCB formats share the same interface and foundation extension.

Offset	Type/Value	Len	Name (Dim)	Description
DATA CONTROL BLOCK DEFINITION				
	X'80'		DCBBIT0	″128″
	X'40'		DCBBIT1	″64″
	X'20'		DCBBIT2	″32″
	X'10'		DCBBIT3	″16″
	X'8'		DCBBIT4	″8″
	X'4'		DCBBIT5	″4″
	X'2'		DCBBIT6	″2″
	X'1'		DCBBIT7	″1″
	DEVICE INTERFACES			
	WORLD TRADE TELEGRAPH			
16 (10)	BITSTRING	1	DCBBQFLG	– WTTA FLAG BYTE
	.1.		DCBBQWRU	″DCBBIT1″– WRU FEATURE IS TO BE USED
	. .1.		DCBBQIAM	″DCBBIT2″– IAM FEATURE IS TO BE USED
	. . .1		DCBBQWRS	″DCBBIT3″– WRU FEATURE TO BE USED IN SEND HEADER SUBGROUP
	. . . . 1. . .		DCBBQWRE	″DCBBIT4″– WRU FEATURE TO BE USED IN END SEND SUBGROUP
17 (11)	CHARACTER	1	DCBWTEOM	– EOM CHARACTER
18 (12)	CHARACTER	1	DCBWTEOT	– EOT CHARACTER
19 (13)	SIGNED	1	DCBWTPAD	– NUMBER OF PAD (LTRS) CHARACTERS REQUIRED FOR MOTOR-ON DELAY
	BTAM LINE GROUP INTERFACE			
20 (14)	ADDRESS	4	DCBBUFCB (0)	– ADDRESS OF BUFFER POOL CONTROL BLOCK
20 (14)	SIGNED	1	DCBBUFNO	– NUMBER OF BUFFERS OBTAINED BY OPEN
21 (15)	ADDRESS	3	DCBBUFCA	– ADDRESS OF BUFFER POOL CONTROL BLOCK
24 (18)	SIGNED	2	DCBBUFL	– BUFFER LENGTH
26 (1A)	BITSTRING	2	DCBDSORG (0)	– DATA SET ORGANIZATION BEING USED
26 (1A)	BITSTRING	1	DCBDSRG1	– FIRST BYTE OF DCBDSORG

Offset	Type/Value	Len	Name (Dim)	Description
	1.		DCBDSGIS	″DCBBIT0″– IS – INDEXED SEQUENTIAL ORGANIZATION
	.1.		DCBDSGPS	″DCBBIT1″– PS – PHYSICAL SEQUENTIAL ORGANIZATION
	. .1.		DCBDSGDA	″DCBBIT2″– DA – DIRECT ORGANIZATION
	. . .1		DCBDSGCX	″DCBBIT3″– CX – BTAM OR QTAM LINE GROUP
	. . . . . .1.		DCBDSGPO	″DCBBIT6″– PO – PARTITIONED ORGANIZATION
	1		DCBDSGU	″DCBBIT7″– U – UNMOVABLE, THE DATA CONTAINS LOCATION DEPENDENT INFORMATION
27 (1B)	BITSTRING	1	DCBDSRG2	– SECOND BYTE OF DCBDSORG
1	1.		DCBDSGGS	″DCBBIT0″– GS – GRAPHICS ORGANIZATION
	.1.		DCBDSGTX	″DCBBIT1″– TX – TCAM LINE GROUP
	. .1.		DCBDSGTQ	″DCBBIT2″– TQ – TCAM MESSAGE QUEUE
	. . . . 1. . .		DCBACBM	″DCBBIT4″– ACCESS METHOD CONTROL BLOCK
	. . . . .1. .		DCBDSGTR	″DCBBIT5″– TR – TCAM 3705
28 (1C)	ADDRESS	4	DCBIOBAD (0)	– BASE FOR ADDRESSING IOB'S (BASE = ADDRESS OF FIRST IOB MINUS LENGTH OF AN IOB)
28 (1C)	SIGNED	1	DCBDEVTP	– INDEX TO DEVICE ENTRY IN THE DEVICE I/O DIRECTORY
29 (1D)	ADDRESS	3	DCBIOBAA	–- SAME AS DCBIOBAD ABOVE
32 (20)	BITSTRING	1	DCBHIARC (0)	– HIERARCHY FLAG BITS
32 (20)	BITSTRING	1	DCBBFTEK	– BUFFERING TECHNIQUE FLAG BITS
	1.		DCBH1	″DCBBIT0″– HIERARCHY 1 MAIN STORAGE – BIT 5 IS ZERO
	.111		DCBBFT	″DCBBIT1+DCBBIT2+DCBBIT3″ BUFFERING TECHNIQUE
	.11.		DCBBFTA	″DCBBIT1+DCBBIT2″– QSAM LOCATE MODE PROCESSING OF SPANNED RECORDS – OPEN IS TO CONSTRUCT A RECORD AREA IF IT AUTOMATICALLY CONSTRUCTS BUFFERS
	. .1.		DCBBFTR	″DCBBIT2″– FOR BSAM CREATE BDAM PROCESSING OF UNBLOCKED SPANNED RECORDS – SOFTWARE TRACK OVERFLOW. FOR BSAM INPUT PROCESSING OF UNBLOCKED SPANNED RECORDS
	.1.		DCBBFTS	″DCBBIT1″– SIMPLE BUFFERING – BIT 3 IS ZERO
	. .1.		DCBBFTKR	″DCBBIT2″– UNBLOCKED SPANNED RECORDS – SOFTWARE TRACK OVERFLOW (BDAM)
	. . .1		DCBBFTE	″DCBBIT3″– EXCHANGE BUFFERING – BIT 1 IS ZERO
	. . . . 1. . .		DCBBFTKD	″DCBBIT4″– DYNAMIC BUFFERING (BTAM)
	. . . . 1. . .		DCBBFTK	″DCBBIT4″– LRECL IN 'K' UNITS FOR XLRI
	.11. 1. . .		DCBBXLRI	″DCBBIT1+DCBBIT2+DCBBIT4″– EXTENDED LOGICAL RECORD INTERFACE MODE FOR SPANNED RECORDS GREATER THAN 32K – ALL RECORDS TRANSFERRED TO OR FROM THE RECORD AREA.
	. . . . .1. .		DCBH0	″DCBBIT5″– HIERARCHY 0 MAIN STORAGE – BIT 0 IS ZERO
	11		DCBBFA	″DCBBIT6+DCBBIT7″– BUFFER ALIGNMENT
	. . . . . .1.		DCBBFAD	″DCBBIT6″– DOUBLEWORD BOUNDARY
	1		DCBBFAF1	″DCBBIT7″– FULLWORD NOT A DOUBLEWORD BOUNDARY, CODED IN DCB MACRO INSTRUCTION

Offset	Type/Value	Len	Name (Dim)	Description
	11		DCBBFAF2	″DCBBIT6+DCBBIT7″– FULLWORD NOT A DOUBLEWORD BOUNDARY, CODED IN DCB MACRO INSTRUCTION
33 (21)	BITSTRING	1	DCBERROP	– ERROR RECOVERY PROCEDURE BITS
34 (22)	SIGNED	1	DCBBUFCT	– MAX NUMBER OF READ BUFFERS
35 (23)	BITSTRING	1		– RESERVED
36 (24)	ADDRESS	4	DCBEXLST (0)	– ADDRESS OF USER–PROVIDED EXIT LIST
36 (24)	SIGNED	1	DCBEIOBX	– SIZE OF IOB
37 (25)	ADDRESS	3	DCBEXLSA	– ADDRESS OF USER–PROVIDED EXIT LIST
33 (21)	BITSTRING	1	DCBERROP	– ERROR RECOVERY PROCEDURE BITS
	. . .1		DCBERPT	″DCBBIT3″– ON–LINE TEST FACILITIES TO BE USED
	. . . . 1. . .		DCBERPC	″DCBBIT4″– THRESHOLD AND CUMULATIVE ERROR COUNTS TO BE MAINTAINED
	. . . . .1. .		DCBERPW	″DCBBIT5″– TEXT-WRITE ERRORS TO BE RETRIED
	. . . . . .1.		DCBERPR	″DCBBIT6″– TEXT–READ ERRORS TO BE RETRIED
	1		DCBERPN	″DCBBIT7″– IF ZERO, BASIC ERP TO BE FOLLOWED --- IF ONE, NO ERP TO BE FOLLOWED
34 (22)	SIGNED	1	DCBBUFCT	– CONTAINS MAXIMUM NUMBER OF BUFFERS TO BE OBTAINED BY BTAM FOR READ OPERATION (DYNAMIC BUFFERING ONLY)
	QTAM LINE GROUP INTERFACE			
20 (14)	ADDRESS	4	DCBCLPS (0)	– ADDRESS OF LINE PROCEDURE SPECIFICATION ROUTINE
20 (14)	SIGNED	1	DCBBUFRQ	– NUMBER OF BUFFERS REQUESTED FOR A READ OR WRITE OPERATION
21 (15)	ADDRESS	3	DCBCLPSA	– SAME AS DCBCLPS ABOVE
24 (18)	SIGNED	1	DCBINTVL	– NUMBER OF SECONDS OF INTENTIONAL DELAY BETWEEN PASSES THROUGH A POLLING LIST FOR NONSWITCHED LINES
25 (19)	BITSTRING	1		– RESERVED
26 (1A)	BITSTRING	2	DCBDSORG(0)	– DATA SET ORGANIZATION
26 (1A)	BITSTRING	1	DCBDSRG1	– FIRST BYTE OF DCBDSORG
27 (1B)	BITSTRING	1	DCBDSRG2	– SECOND BYTE OF DCBDSORG
28 (1C)	ADDRESS	1	DCBDEVTP	– DEVICE TYPE POINTER
29 (1D)	ADDRESS	3	DCBIOBAA	– ADDRESS OF FIRST IOB
32 (20)	ADDRESS	4	DCBLCBAD (0)	–BASE FOR ADDRESSING LCB'S (BASE = ADDRESS OF FIRST LCB MINUS LENGTH OF ONE LCB)
32 (20)	BITSTRING	1	DCBCPRI	– COMMUNICATION PRIORITY BITS
	. . . . .1. .		DCBCPR	″DCBBIT5″– RECEIVING HAS PRIORITY
	. . . . . .1.		DCBCPE	″DCBBIT6″– RECEIVING AND SENDING HAVE EQUAL PRIORITY
	1		DCBCPS	″DCBBIT7″– SENDING HAS PRIORITY
33 (21)	2.ADDRESS	3	DCBLCBA	– SAME AS DCBLCBAD ABOVE
36 (24)	SIGNED	1	DCBEIOBX	– EXTENDED IOB INDEX. SIZE OF LCB.
37 (25)	ADDRESS	3	DCBEXLSA	– ADDRESS OF EXIT LIST
	FOUNDATION BEFORE OPEN			
40 (28)	CHARACTER	8	DCBDDNAM	– NAME ON THE DD STATEMENT WHICH DEFINES THE DATA SET ASSOCIATED WITH THIS DCB
48 (30)	BITSTRING	1	DCBOFLGS	– FLAGS USED BY OPEN ROUTINE
	1.		DCBOFLWR	″DCBBIT0″– IF ZERO, LAST I/O OPERATION WAS READ OR POINT. IF ONE, LAST I/O OPERATION WAS WRITE.

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Offset	Type/Value	Len	Name (Dim)	Description
	1.		DCBOFIOD	″DCBBIT0″– DATA SET IS BEING OPENED FOR INPUT OR OUTPUT (BDAM)
	.1.		DCBOFLRB	″DCBBIT1″– LAST I/O OPERATION WAS IN READ BACKWARD MODE
	. .1.		DCBOFEOV	DCBBIT2″– 1 BY EOV WHEN IT CALLS CLOSE ROUTINE FOR CONCATENATION OF DATA SETS WITH UNLIKE ATTRIBUTES
	. . .1		DCBOFOPN	″DCBBIT3″– AN OPEN HAS BEEN SUCCESSFULLY COMPLETED
	. . . . 1. . .		DCBOFPPC	″DCBBIT4″– SET TO 1 BY PROBLEM PROGRAM TO INDICATE A CONCATENATION OF UNLIKE ATTRIBUTES
	. . . . .1. .		DCBOFTM	″DCBBIT5″– TAPE MARK HAS BEEN READ
	. . . . . .1.		DCBOFUEX	″DCBBIT6″– SET TO 0 BY AN I/O SUPPORT FUNCTION WHEN THAT FUNCTION TAKES A USER EXIT. SET TO 1 ON RETURN FROM USER EXIT TO THE I/O SUPPORT FUNCTION WHICH TOOK THE EXIT.
	1		DCBOFIOF	″DCBBIT7″– SET TO 1 BY AN I/O SUPPORT FUNCTION IF DCB IS TO BE PROCESSED BY THAT FUNCTION
49 (31)	BITSTRING	1	DCBIFLG	– FLAGS USED BY IOS IN COMMUNICATING ERROR CONDITIONS AND IN DETERMINING CORRECTIVE PROCEDURES
	11.		DCBIBEC	″DCBBIT0+DCBBIT1″– ERROR CORRECTION INDICATOR
	. . . . .		DCBIFNEP	″X'00'″– NOT IN ERROR PROCEDURE
	.1.		DCBEX	″DCBBIT1″– ERROR CORRECTION OR IOS PAGE FIX IN PROCESS
	11.		DCBIFPEC	″DCBBIT0+DCBBIT1″– PERMANENT ERROR CORRECTION
1	. .11		DCBIBPCT	″DCBBIT2+DCBBIT3″– PRINTER CARRIAGE TAPE PUNCH INDICATOR
	. .1.		DCBIFC9	″DCBBIT2″– CHANNEL 9 PRINTER CARRIAGE TAPE PUNCH SENSED
	. . .1		DCBIFC12	″DCBBIT3″– CHANNEL 12 PRINTER CARRIAGE TAPE PUNCH SENSED
	. . . . 11. .		DCBIBIOE	″DCBBIT4+DCBBIT5″– IOS ERROR ROUTINE USE INDICATOR
	. . . . .		DCBIFER	″X'00'″– ALWAYS USE I/O SUPERVISOR ERROR ROUTINE
	. . . . .1. .		DCBIFNE1	″DCBBIT5″– NEVER USE I/O SUPERVISOR ERROR ROUTINE
	. . . . .1. .		DCBIFTIM	″DCBBIT5″– TEST IOS MASK (IMSK) FOR ERROR PROCEDURE (BTAM)
	. . . . 1. . .		DCBIFNE2	″DCBBIT4″– NEVER USE I/O SUPERVISOR ERROR ROUTINE
	. . . . 11. .		DCBIFNE3	″DCBBIT4+DCBBIT5″– NEVER USE I/O SUPERVISOR ERROR ROUTINE
50 (32)	BITSTRING	2	DCBMACR (0)	– MACRO INSTRUCTION REFERENCE
50 (32)	BITSTRING	1	DCBMACR1	– FIRST BYTE OF DCBMACR
	1.		DCBMRECP	″DCBBIT0″– EXECUTE CHANNEL PROGRAM (EXCP) --- ALWAYS ZERO (BSAM, QSAM, BPAM, BISAM, QISAM, BDAM) --- RESERVED (QTAM, BTAM)

Offset	Type/Value	Len	Name (Dim)	Description
	.1.		DCBMRFE	″DCBBIT1″– FOUNDATION EXTENSION IS PRESENT (EXCP)
	.1.		DCBMRGET	″DCBBIT1″– GET (QSAM, QISAM, TCAM)
	.1.		DCBMRPTQ	″DCBBIT1″– PUT FOR MESSAGE GROUP (QTAM) --- ALWAYS ZERO (BSAM, BPAM, BISAM, BDAM) --- RESERVED (BTAM)
	. .1.		DCBMRAPG	″DCBBIT2″– APPENDAGES ARE REQUIRED (EXCP)
	. .1.		DCBMRRD	″DCBBIT2″– READ (BSAM, BPAM, BISAM, BDAM, BTAM)
	. .1.		DCBMRWRQ	″DCBBIT2″– WRITE FOR LINE GROUP (QTAM) --- ALWAYS ZERO (QSAM, QISAM)
	. . .1		DCBMRCI	″DCBBIT3″– COMMON INTERFACE (EXCP)
	. . .1		DCBMRMVG	″DCBBIT3″– MOVE MODE OF GET (QSAM, QISAM)
	. . .1		DCBMRRDK	″DCBBIT3″– KEY SEGMENT WITH READ (BDAM) --- ALWAYS ZERO (BISAM) --- RESERVED (BSAM, BPAM, QTAM, BTAM)
	. . . . 1. . .		DCBMRLCG	″DCBBIT4″– LOCATE MODE OF GET (QSAM, QISAM)
	. . . . 1. . .		DCBMRRDI	″DCBBIT4″– ID ARGUMENT WITH READ (BDAM) --- ALWAYS ZERO (BISAM) --- RESERVED (EXCP, BSAM, BPAM, QTAM, BTAM)
	. . . . .1. .		DCBMRABC	″DCBBIT5″– USER'S PROGRAM MAINTAINS ACCURATE BLOCK COUNT (EXCP)
	. . . . .1. .		DCBMRPT1	″DCBBIT5″– POINT (WHICH IMPLIES NOTE) (BSAM, BPAM)
	. . . . .1. .		DCBMRSBG	″DCBBIT5″– SUBSTITUTE MODE OF GET (QSAM)
	. . . . .1. .		DCBMRDBF	″DCBBIT5″– DYNAMIC BUFFERING (BISAM, BDAM) --- ALWAYS ZERO (QISAM) --- RESERVED (QTAM, BTAM)
	. . . . . .1.		DCBPGFXA	″DCBBIT6″– PAGE FIX APPENDAGE IS SPECIFIED (EXCP)
	. . . . . .1.		DCBMRCRL	″DCBBIT6″– CNTRL (BSAM, QSAM)
	. . . . . .1.		DCBMRCHK	″DCBBIT6″– CHECK (BISAM)
	. . . . . .1.		DCBMRRDX	″DCBBIT6″– READ EXCLUSIVE (BDAM) --- RESERVED (BPAM, QISAM, QTAM, BTAM)
	1		DCBMRDMG	″DCBBIT7″– DATA MODE OF GET (QSAM)
	1		DCBMRCK	″DCBBIT7″– CHECK (BDAM) --- RESERVED (EXCP, BSAM, BPAM, BISAM, QISAM, QTAM, BTAM)
51 (33)	BITSTRING	1	DCBMACR2	– SECOND BYTE OF DCBMACR
	1.		DCBMRSTL	″DCBBIT0″– SETL (QISAM) --- ALWAYS ZERO (BSAM, QSAM, BPAM, BISAM, BDAM) --- RESERVED (EXCP, QTAM, BTAM)
	.1.		DCBMRPUT	″DCBBIT1″– PUT (QSAM, TCAM) - PUT OR PUTX (QISAM)
	.1.		DCBMRGTQ	DCBBIT1″– GET FOR MESSAGE GROUP (QTAM) --- ALWAYS ZERO (BSAM, BPAM, BISAM, BDAM) --- RESERVED (EXCP, BTAM)
	. .1.		DCBMRWRT	″DCBBIT2″– WRITE (BSAM, BPAM, BISAM, BDAM, BTAM)
	. .1.		DCBMRRDQ	″DCBBIT2″– READ FOR LINE GROUP (QTAM) --- ALWAYS ZERO (QSAM, QISAM) --- RESERVED (EXCP)
	. . .1		DCBMRMVP	″DCBBIT3″– MOVE MODE OF PUT (QSAM, QISAM)
	. . .1		DCBMRWRK	″DCBBIT3″– KEY SEGMENT WITH WRITE (BDAM) --- ALWAYS ZERO (BISAM) --- RESERVED (EXCP, BSAM, BPAM, QTAM, BTAM)

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . 1. . .		DCBMR5WD	″DCBBIT4″– FIVE-WORD DEVICE INTERFACE (EXCP)
	. . . . 1. . .		DCBMRLDM	″DCBBIT4″– LOAD MODE BSAM (CREATE BDAM DATA SET) (BSAM)
	. . . . 1. . .		DCBMRLCP	″DCBBIT4″– LOCATE MODE OF PUT (QSAM, QISAM)
	. . . . 1. . .		DCBMRIDW	″DCBBIT4″– ID ARGUMENT WITH WRITE (BDAM) --- ALWAYS ZERO (BISAM) --- RESERVED (BPAM, QTAM, BTAM)
	. . . . .1. .		DCBMR4WD	″DCBBIT5″– FOUR-WORD DEVICE INTERFACE (EXCP)
	. . . . .1. .		DCBMRPT2	″DCBBIT5″– POINT (WHICH IMPLIES NOTE) (BSAM, BPAM)
	. . . . .1. .		DCBMRTMD	″DCBBIT5″– SUBSTITUTE MODE (QSAM)
	. . . . .1. .		DCBMRUIP	″DCBBIT5″– UPDATE IN PLACE (PUTX) (QISAM) --- ALWAYS ZERO (BISAM) --- RESERVED (BDAM, QTAM, BTAM)
	. . . . . .1.		DCBMR3WD	″DCBBIT6″– THREE–WORD DEVICE INTERFACE (EXCP)
	. . . . . .1.		DCBMRCTL	″DCBBIT6″– CNTRL (BSAM, QSAM)
	. . . . . .1.		DCBMRSTK	″DCBBIT6″– SETL BY KEY (QISAM)
	. . . . . .1.		DCBMRAWR	″DCBBIT6″– ADD TYPE OF WRITE (BDAM) --- ALWAYS ZERO (BISAM) --- RESERVED (BPAM, QTAM, BTAM)
	1		DCBMR1WD	″DCBBIT7″– ONE–WORD DEVICE INTERFACE (EXCP)
	1		DCBMRSWA	″DCBBIT7″– USER’S PROGRAM HAS PROVIDED A SEGMENT WORK AREA POOL (BSAM CREATE BDAM, BDAM)
	1		DCBMRDMD	″DCBBIT7″– DATA MODE (QSAM)
	1		DCBMRSTI	″DCBBIT7″– SETL BY ID (QISAM) --- ALWAYS ZERO (BISAM) --- RESERVED (BPAM, QTAM, BTAM)
	FOUNDATION AFTER OPEN			
40 (28)	ADDRESS	2	DCBTIOT	OFFSET FROM TIOT ORIGIN TO TIOELNGH FIELD IN TIOT ENTRY FOR DD STATEMENT ASSOCIATED WITH THIS DCB
42 (2A)	BITSTRING	2	DCBMACRF (0)	– SAME AS DCBMACR BEFORE OPEN
42 (2A)	BITSTRING	1	DCBMACF1	– FIRST BYTE OF DCBMACRF
43 (2B)	BITSTRING	1	DCBMACF2	– SECOND BYTE OF DCBMACRF
44 (2C)	ADDRESS	4	DCBDEBAD (0)	– ADDRESS OF ASSOCIATED DEB
44 (2C)	BITSTRING	1	DCBIFLGS	– SAME AS DCBIFLG BEFORE OPEN
	11.		DCBIFEC	″DCBBIT0+DCBBIT1″– CORRECTION INDICATOR
	. .11		DCBIFPCT	″DCBBIT2+DCBBIT3″– PRINTER CARRIAGE TAPE PUNCH INDICATOR
	. . . . 11. .		DCBIFIOE	″DCBBIT4+DCBBIT5″– IOS ERROR ROUTINE USE INDICATOR
	. . . . . .1.		DCBIFLDT	″DCBBIT6″– POSSIBLE LOST DATA CONDITION SUCH AS FOR A PRINTER
45 (2D)	ADDRESS	3	DCBDEBA	– ADDRESS OF ASSOCIATED DEB
48 (30)	ADDRESS	4	DCBREAD (0)	– ADDRESS OF READ MODULE
48 (30)	ADDRESS	4	DCBWRITE (0)	– ADDRESS OF WRITE MODULE
48 (30)	BITSTRING	1	DCBOFLG	SAME AS DCBOFLGS BEFORE OPEN
49 (31)	ADDRESS	3	DCBREADA (0)	ADDRESS OF READ MODULE
49 (31)	ADDRESS	3	DCBWRITA	ADDRESS OF WRITE MODULE
48 (30)	ADDRESS	4	DCBGET (0)	– ADDRESS OF GET MODULE

Offset	Type/Value	Len	Name (Dim)	Description
48 (30)	ADDRESS	4	DCBPUT (0)	– ADDRESS OF PUT MODULE
48 (30)	BITSTRING	1	DCBOFLG1	SAME AS DCBOFLGS BEFORE OPEN
49 (31)	ADDRESS	3	DCBGETA (0)	ADDRESS OF GET MODULE
49 (31)	ADDRESS	3	DCBPUTA	ADDRESS OF PUT MODULE
	QTAM INTERFACE			
52 (34)	CHARACTER	4	DCBKSTAT (0)	– FOUR THRESHOLD VALUES FOR ERROR COUNTS
52 (34)	SIGNED	1	DCBKSTA1	–THRESHOLD VALUE FOR NUMBER OF TRANSMISSIONS
53 (35)	SIGNED	1	DCBKSTA2	– THRESHOLD VALUE FOR NUMBER OF DATA CHECKS
54 (36)	SIGNED	1	DCBKSTA3	– THRESHOLD VALUE FOR NUMBER OF INTERVENTIONS REQUIRED
55 (37)	SIGNED	1	DCBKSTA4	–OLD VALUE FOR NUMBER OF TIMEOUTS
	QTAM POLLING LIST ORIGIN			
56 (38)	ADDRESS	4	DCBCPOLL (0)	– A 4–BYTE FIELD FOR EACH POLLING LIST
56 (38)	BITSTRING	1	DCBPLBYT	–ADAPTER TYPE
1	. . . . 1. . .		DCBCPWTT	″DCBBIT4″– WTTA
57 (39)	ADDRESS	3	DCBCPOLA	– ADDRESS OF THE POLLING LIST
	BTAM INTERFACE			
52 (34)	ADDRESS	4	DCBLERB (0)	– ADDRESS OF LINE ERROR BLOCK
52 (34)	BITSTRING	1	DCBRDYI (0)	– READYQ INDICATORS
	. . . . .1. .		DCBRDYPO	″DCBBIT5″– APPLICATION REQUESTS NOTIFICATION OF POWER–ON THAT IS AFTER INITIAL READY
	. . . . . .1.		DCBRDYIQ	″DCBBIT6″– ADDRESS IS READYQ AND NOT LERB
	1		DCBRDYIZ	″DCBBIT7″– READYQ SPECIFIED, BUT ADDRESS WAS 0, SO USING BTAM READYQ ROUTINE
52 (34)	ADDRESS	4	DCBRDYQ	– ADDRESS OF USER/BTAM ROUTINE TO PROCESS LOCAL 3270 DEVICE READY INTERRUPTS
	BSC INTERFACE			
56 (38)	BITSTRING	1	DCBXMODE	– MODE OF TRANSMISSION FOR BINARY SYNCHRONOUS COMMUNICATION (BSC)
	.1.		DCBXMIBC	″DCBBIT1″– INTERMEDIATE BLOCK CHECKING IS TO BE PERFORMED
	. .1.		DCBXMDA1	″DCBBIT2″– TRANSMISSION IS THROUGH A 2701 DATA ADAPTER UNIT DUAL COMMUNICATION INTERFACE B
	. . . . 1. . .		DCBXMDA2	″DCBBIT4″– TRANSMISSION IS IN CODE B FOR A 2701 DATA ADAPTER UNIT DUAL CODE FEATURE
57 (39)	BITSTRING	1	DCBXCODE	– BSC CONTROL STATION FLAG AND TRANSMISSION CODE
	1.		DCBXCCSF	″DCBBIT0″– BSC CONTROL STATION FLAG --- IF ZERO, THIS IS THE CONTROL STATION. IF ONE, THIS IS THE REMOTE STATION.
1	.1.		DCBXCPTP	″DCBBIT1″– IF PTOP IS SPECIFIED IN SYSGEN PROCEDURE – SCHEDULE AN ASYNCHRONOUS EXIT TO INTERFACE RESOLUTION ROUTINE
	. .1.		DCBXCTR1	″DCBBIT2″– 6–BIT TRANSCODE IS BEING USED (BIT 4 IS ALSO ON)
	. . .1		DCBXCAS1	″DCBBIT3″– USASCII TRANSMISSION CODE IS BEING USED (BIT 5 IS ALSO ON)
	. . . . 11. .		DCBXCEBC	″DCBBIT4+DCBBIT5″– IF BOTH BITS ARE ZERO, EBCDIC TRANSMISSION CODE IS BEING USED.

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . 1. . .		DCBXCTR2	″DCBBIT4″– 6–BIT TRANSCODE IS BEING USED (BIT 2 IS ALSO ON)
	. . . . .1. .		DCBXCAS2	″DCBBIT5″– USASCII TRANSMISSION CODE IS BEING USED (BIT 3 IS ALSO ON)
58 (3A)	CHARACTER	1	DCBBSRSV	– DLE CONTROL CHARACTER
59 (3B)	BITSTRING	1	DCBBSWBT	– RESERVED
60 (3C)	ADDRESS	4	DCBIRRAD (0)	– BEFORE OPEN – IF PTOP IS SPECIFIED IN THE SYSGEN PROCEDURE, ADDRESS OF INTERFACE RESOLUTION ROUTINE. AFTER OPEN, THE FOLLOWING 4 CHARACTERS OCCUPY THIS SPACE.
60 (3C)	CHARACTER	1	DCBBSTSX	– DLE CONTROL CHARACTER
61 (3D)	CHARACTER	1	DCBBSSTX	– STX CONTROL CHARACTER
62 (3E)	CHARACTER	1	DCBBSTEX	– DLE CONTROL CHARACTER
63 (3F)	CHARACTER	1	DCBBSETX	– ETX CONTROL CHARACTER
64 (40)	CHARACTER	2	DCBBSAK0	– ACK–0 CONTROL CHARACTER
66 (42)	CHARACTER	2	DCBBSAK1	– ACK–1 CONTROL CHARACTER
68 (44)	CHARACTER	1	DCBBSENQ	– ENQ CONTROL CHARACTER
69 (45)	CHARACTER	1	DCBBSNAK	– NAK CONTROL CHARACTER
70 (46)	CHARACTER	1	DCBBSETB	– ETB CONTROL CHARACTER
71 (47)	CHARACTER	1	DCBBSDLE	– DLE CONTROL CHARACTER
72 (48)	CHARACTER	1	DCBBSEOT	–NTROL CHARACTER
73 (49)	CHARACTER	3	DCBBSSYN	– SYN, SYN, SYN CONTROL CHARACTERS
76 (4C)	CHARACTER	2	DCBBSONL	–SOH % CONTROL CHARACTERS
78 (4E)	CHARACTER	2	DCBBSSAK	– WACK CONTROL CHARACTERS
80 (50)	CHARACTER	2	DCBBSRVI	– DLE @ CONTROL CHARACTERS
82 (52)	BITSTRING	18		– RESERVED

Cross-Reference

Name	Hex Offset	Hex Value	Level
DCBACBM	1B	8	2
DCBBFA	20	3	2
DCBBFAD	20	2	2
DCBBFAF1	20	1	2
DCBBFAF2	20	3	2
DCBBFT	20	70	2
DCBBFTA	20	60	2
DCBBFTE	20	10	2
DCBBFTEK	20		2
DCBBFTK	20	8	2
DCBBFTKD	20	8	2
DCBBFTKR	20	20	2
DCBBFTR	20	20	2
DCBBFTS	20	40	2
DCBBIT0	0	80	2
DCBBIT1	0	40	2
DCBBIT2	0	20	2
DCBBIT3	0	10	2
DCBBIT4	0	8	2
DCBBIT5	0	4	2
DCBBIT6	0	2	2
DCBBIT7	0	1	2

Name	Hex Offset	Hex Value	Level
DCBBQFLG	10		2
DCBBQIAM	10	20	2
DCBBQWRE	10	8	2
DCBBQWRS	10	10	2
DCBBQWRU	10	40	2
DCBBSAK0	40		2
DCBBSAK1	42		2
DCBBSDLE	47		2
DCBBSENQ	44		2
DCBBSEOT	48		2
DCBBSETB	46		2
DCBBSETX	3F		2
DCBBSNAK	45		2
DCBBSONL	4C		2
DCBBSRSV	3A		2
DCBBSRVI	50		2
DCBBSSAK	4E		2
DCBBSSTX	3D		2
DCBBSSYN	49		2
DCBBSTEX	3E		2
DCBBSTSX	3C		2
DCBBSWBT	3B		2
DCBBUFCA	15		2
DCBBUFCB	14		2
DCBBUFCT	22		2
DCBBUFL	18		2
DCBBUFNO	14		2
DCBBUFRQ	14		2
DCBBXLRI	20	68	2
DCBCLPS	14		2
DCBCLPSA	15		2
DCBCPE	20	2	2
DCBCPOLA	39		2
DCBCPOLL	38		2
DCBCPR	20	4	2
DCBCPRI	20		2
DCBCPS	20	1	2
DCBCPWTT	38	8	2
DCBDDNAM	28		2
DCBDEBA	2D		2
DCBDEBAD	2C		2
DCBDEVTP	1C		2
DCBDSGCX	1A	10	2
DCBDSGDA	1A	20	2
DCBDSGGS	1B	80	2
DCBDSGIS	1A	80	2
DCBDSGPO	1A	2	2
DCBDSGPS	1A	40	2
DCBDSGTQ	1B	20	2
DCBDSGTR	1B	4	2
DCBDSGTX	1B	40	2
DCBDSGU	1A	1	2
DCBDSORG	1A		2
DCBDSRG1	1A		2

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Name	Hex Offset	Hex Value	Level
DCBDSRG2	1B		2
DCBEIOBX	24		2
DCBERPC	21	8	2
DCBERPN	21	1	2
DCBERPR	21	2	2
DCBERPT	21	10	2
DCBERPW	21	4	2
DCBERROP	21		2
DCBEX	31	40	2
DCBEXLSA	25		2
DCBEXLST	24		2
DCBGET	30		2
DCBGETA	31		2
DCBHIARC	20		2
DCBH0	20	4	2
DCBH1	20	80	2
DCBIBEC	31	C0	2
DCBIBIOE	31	C	2
DCBIBPCT	31	30	2
DCBIFC12	31	10	2
DCBIFC9	31	20	2
DCBIFEC	2C	C0	2
DCBIFER	31	0	2
DCBIFIOE	2C	C	2
DCBIFLDT	2C	2	2
DCBIFLG	31		2
DCBIFLGS	2C		2
DCBIFNEP	31	0	2
DCBIFNE1	31	4	2
DCBIFNE2	31	8	2
DCBIFNE3	31	C	2
DCBIFPCT	2C	30	2
DCBIFPEC	31	C0	2
DCBIFTIM	31	4	2
DCBINTVL	18		2
DCBIOBAA	1D		2
DCBIOBAD	1C		2
DCBIRRAD	3C		2
DCBKSTAT	34		2
DCBKSTA1	34		2
DCBKSTA2	35		2
DCBKSTA3	36		2
DCBKSTA4	37		2
DCBLCBA	21		2
DCBLCBAD	20		2
DCBLERB	34		2
DCBLNGBX	52	64	2
DCBLNGCX	39	3C	2
DCBLNGQX	39	3C	2
DCBLNGXE	33	34	2
DCBMACF1	2A		2
DCBMACF2	2B		2
DCBMACR	32		2
DCBMACRF	2A		2

Name	Hex Offset	Hex Value	Level
DCBMACR1	32		2
DCBMACR2	33		2
DCBMRABC	32	4	2
DCBMRAPG	32	20	2
DCBMRAWR	33	2	2
DCBMRCHK	32	2	2
DCBMRCI	32	10	2
DCBMRCK	32	1	2
DCBMRCRL	32	2	2
DCBMRCTL	33	2	2
DCBMRDBF	32	4	2
DCBMRDMD	33	1	2
DCBMRDMG	32	1	2
DCBMRECP	32	80	2
DCBMRFE	32	40	2
DCBMRGET	32	40	2
DCBMRGTQ	33	40	2
DCBMRIDW	33	8	2
DCBMRLCG	32	8	2
DCBMRLCP	33	8	2
DCBMRLDM	33	8	2
DCBMRMVG	32	10	2
DCBMRMVP	33	10	2
DCBMRPTQ	32	40	2
DCBMRPT1	32	4	2
DCBMRPT2	33	4	2
DCBMRPUT	33	40	2
DCBMRRD	32	20	2
DCBMRRDI	32	8	2
DCBMRRDK	32	10	2
DCBMRRDQ	33	20	2
DCBMRRDX	32	2	2
DCBMRSBG	32	4	2
DCBMRSTI	33	1	2
DCBMRSTK	33	2	2
DCBMRSTL	33	80	2
DCBMRSWA	33	1	2
DCBMRTMD	33	4	2
DCBMRUIP	33	4	2
DCBMRWRK	33	10	2
DCBMRWRQ	32	20	2
DCBMRWRT	33	20	2
DCBMR1WD	33	1	2
DCBMR3WD	33	2	2
DCBMR4WD	33	4	2
DCBMR5WD	33	8	2
DCBOFEOV	30	20	2
DCBOFIOD	30	80	2
DCBOFIOF	30	1	2
DCBOFLG	30		2
DCBOFLGS	30		2
DCBOFLG1	30		2
DCBOFLRB	30	40	2
DCBOFLWR	30	80	2

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Name	Hex Offset	Hex Value	Level
DCBOFOPN	30	10	2
DCBOFPPC	30	8	2
DCBOFTM	30	4	2
DCBOFUEX	30	2	2
DCBPGFXA	32	2	2
DCBPLBYT	38		2
DCBPUT	30		2
DCBPUTA	31		2
DCBRDYI	34		2
DCBRDYIQ	34	2	2
DCBRDYIZ	34	1	2
DCBRDYPO	34	4	2
DCBRDYQ	34		2
DCBREAD	30		2
DCBREADA	31		2
DCBTIOT	28		2
DCBWRITA	31		2
DCBWRITE	30		2
DCBWTEOM	11		2
DCBWTEOT	12		2
DCBWTPAD	13		2
DCBXCAS1	39	10	2
DCBXCAS2	39	4	2
DCBXCCSF	39	80	2
DCBXCEBC	39	C	2
DCBXCODE	39		2
DCBXCPTP	39	40	2
DCBXCTR1	39	20	2
DCBXCTR2	39	8	2
DCBXMDA1	38	20	2
DCBXMDA2	38	8	2
DCBXMIBC	38	40	2
DCBXMODE	38		2

DCB for TCAM

Common Name: Data Control Block for TCAM
Macro Name: DCBD
DSECT Name: IHADCB
Owning Component: Data Management, subcomponent OPEN/CLOSE/EOV
Eye-Catcher ID: None
Subpool and Key: Problem program subpool and key
Size: Variable 64 bytes
Created by: Problem program
Pointed to by:

- DEBDCBAD field of the DEB data area
- IOBDCBPT field of the IOB data area

Serialization: None
Function: The format of the data control block (DCB) used by the telecommunications access method (TCAM) depends on the type of data set it represents. The five types of DCB formats used in TCAM message control programs and application programs are line

groups, message queues, checkpoint data set, message logging, and application programs.

Offset	Type/Value	Len	Name (Dim)	Description
DATA CONTROL BLOCK DEFINITION				
	X'80'		DCBBIT0	″128″
	X'40'		DCBBIT1	″64″
	X'20'		DCBBIT2	″32″
	X'10'		DCBBIT3	″16″
	X'8'		DCBBIT4	″8″
	X'4'		DCBBIT5	″4″
	X'2'		DCBBIT6	″2″
	X'1'		DCBBIT7	″1″
	DEVICE INTERFACES			
	WORLD TRADE TELEGRAPH			
16 (10)	BITSTRING	1	DCBBQFLG	– WTTA FLAG BYTE
	.1.		DCBBQWRU	″DCBBIT1″– WRU FEATURE IS TO BE USED
	. .1.		DCBBQIAM	″DCBBIT2″– IAM FEATURE IS TO BE USED
	. . .1		DCBBQWRS	″DCBBIT3″– WRU FEATURE TO BE USED IN SEND HEADER SUBGROUP
	. . . . 1. . .		DCBBQWRE	″DCBBIT4″– WRU FEATURE TO BE USED IN END SEND SUBGROUP
17 (11)	CHARACTER	1	DCBWTEOM	– EOM CHARACTER
18 (12)	CHARACTER	1	DCBWTEOT	– EOT CHARACTER
19 (13)	SIGNED	1	DCBWTPAD	– NUMBER OF PAD (LTRS) CHARACTERS REQUIRED FOR MOTOR–ON DELAY
	3705 LINE TERMINAL			
8 (8)	CHARACTER	8	DCBIPLTX	– NAME OF MODULE TO BE USED TO IPL THE 3705
16 (10)	ADDRESS	4	DCBBCKUP(0)	– FULLWORD LABEL
16 (10)	BITSTRING	1		– RESERVED
17 (11)	ADDRESS	3	DCBBCKUA	– ADDRESS OF THE DCB FOR THE BACKUP 3705.
	TCAM LINE GROUP INTERFACE			
20 (14)	ADDRESS	4	DCBMHA(0)	– SAME AS DCBMH BELOW
20 (14)	BITSTRING	1	DCBBUFIN(0)	– NUMBER OF INPUT BUFFERS (BITS 0-3)
20 (14)	BITSTRING	1	DCBBUFOU	– NUMBER OF OUTPUT BUFFERS (BITS 4–7)
	1111		DCBBFIN	″DCBBIT0+DCBBIT1+DCBBIT2+ DCBBIT3″ NUMBER OF BUFFERS ASSIGNED INITIALLY FOR RECEIVING OPERATIONS, FOR EACH LINE IN LINE GROUP
	 1111		DCBBFOUT	″DCBBIT4+DCBBIT5+DCBBIT6+ DCBBIT7″ NUMBER OF BUFFERS ASSIGNED INITIALLY FOR SENDING OPERATIONS, FOR EACH LINE IN LINE GROUP
21 (15)	ADDRESS	3	DCBMH	– ADDRESS OF MESSAGE HANDLER FOR THIS LINE GROUP
24 (18)	SIGNED	1	DCBINTVL	– NUMBER OF SECONDS OF INVITATION DELAY
25 (19)	BITSTRING	1	DCBPCI	– PROGRAM CONTROLLED INTERRUPTION HANDLING
	1.		DCBPCIX1	″DCBBIT0″– PCI=(X,)
	.1.		DCBPCIX2	″DCBBIT1″– PCI=(,X)
	. .1.		DCBPCIA1	″DCBBIT2″– PCI=(A,)
	. . .1		DCBPCIA2	″DCBBIT3″– PCI=(,A)
	. . . . 1. . .		DCBPCIN1	″DCBBIT4″– PCI=(N,)
	. . . . .1. .		DCBPCIN2	″DCBBIT5″– PCI=(,N)
	. . . . . .1.		DCBPCIR1	″DCBBIT6″– PCI=(R,)
	1		DCBPCIR2	″DCBBIT7″– PCI=(,R)

Offset	Type/Value	Len	Name (Dim)	Description
26 (1A)	BITSTRING	2	DCBDSORG(0)	– DATA SET ORGANIZATION BEING USED
26 (1A)	BITSTRING	1	DCBDSRG1	– FIRST BYTE OF DCBDSORG
	1.		DCBDSGIS	″DCBBIT0″– IS – INDEXED SEQUENTIAL ORGANIZATION
	.1.		DCBDSGPS	″DCBBIT1″– PS – PHYSICAL SEQUENTIAL ORGANIZATION
	. .1.		DCBDSGDA	″DCBBIT2″– DA – DIRECT ORGANIZATION
	. . .1		DCBDSGCX	″DCBBIT3″– CX – BTAM OR QTAM LINE GROUP
	. . . . . .1.		DCBDSGPO	″DCBBIT6″– PO – PARTITIONED ORGANIZATION
	1		DCBDSGU	″DCBBIT7″– U – UNMOVABLE, THE DATA CONTAINS LOCATION DEPENDENT INFORMATION
27 (1B)	BITSTRING	1	DCBDSRG2	– SECOND BYTE OF DCBDSORG
	1.		DCBDSGGS	″DCBBIT0″– GS – GRAPHICS ORGANIZATION
	.1.		DCBDSGTX	″DCBBIT1″– TX – TCAM LINE GROUP
	. .1.		DCBDSGTQ	″DCBBIT2″– TQ – TCAM MESSAGE QUEUE
	. . . . 1. . .		DCBACBM	″DCBBIT4″– ACCESS METHOD CONTROL BLOCK
	. . . . .1. .		DCBDSGTR	″DCBBIT5″– TR – TCAM 3705
28 (1C)	SIGNED	1	DCBBUFMA	– MAXIMUM NUMBER OF BUFFERS TO BE USED FOR DATA TRANSFER FOR EACH LINE IN THIS GROUP
	QTAM LINE GROUP INTERFACE			
20 (14)	ADDRESS	4	DCBCLPS(0)	– ADDRESS OF LINE PROCEDURE SPECIFICATION ROUTINE
20 (14)	SIGNED	1	DCBBUFRQ	– NUMBER OF BUFFERS REQUESTED FOR A READ OR WRITE OPERATION
21 (15)	ADDRESS	3	DCBCLPSA	– SAME AS DCBCLPS ABOVE
24 (18)	SIGNED	1	DCBINTVL	– NUMBER OF SECONDS OF INTENTIONAL DELAY BETWEEN PASSES THROUGH A POLLING LIST FOR NONSWITCHED LINES
25 (19)	BITSTRING	1		– RESERVED
26 (1A)	BITSTRING	2	DCBDSORG(0)	– DATA SET ORGANIZATION
26 (1A)	BITSTRING	1	DCBDSRG1	– FIRST BYTE OF DCBDSORG
27 (1B)	BITSTRING	1	DCBDSRG2	– SECOND BYTE OF DCBDSORG
28 (1C)	ADDRESS	4	DCBIOBAD(0)	– ADDRESS OF FIRST IOB
28 (1C)	ADDRESS	1	DCBDEVTP	– DEVICE TYPE POINTER
29 (1D)	ADDRESS	3	DCBIOBAA	– ADDRESS OF FIRST IOB
32 (20)	ADDRESS	4	DCBTRANA(0)	– ADDRESS OF TRANSLATION TABLE
32 (20)	BITSTRING	1	DCBCPRI	– COMMUNICATION PRIORITY BITS
33 (21)	ADDRESS	3	DCBTRANS	– ADDRESS OF TRANSLATION TABLE
32 (20)	ADDRESS	4	DCBLCBAD(0)	– BASE FOR ADDRESSING LCB’S (BASE = ADDRESS OF FIRST LCB MINUS LENGTH OF ONE LCB)
32 (20)	BITSTRING	1	DCBCPRI	– COMMUNICATION PRIORITY BITS
	. . . . .1. .		DCBCPR	″DCBBIT5″– RECEIVING HAS PRIORITY
	. . . . . .1.		DCBCPE	″DCBBIT6″– RECEIVING AND SENDING HAVE EQUAL PRIORITY
	1		DCBCPS	″DCBBIT7″– SENDING HAS PRIORITY
33 (21)	ADDRESS	3	DCBLCBA	– SAME AS DCBLCBAD ABOVE
36 (24)	ADDRESS	4	DCBEXLST(0)	– ADDRESS OF EXIT LIST
36 (24)	SIGNED	1	DCBEIOBX	– EXTENDED IOB INDEX. SIZE OF LCB
37 (25)	ADDRESS	3	DCBEXLSA	– ADDRESS OF EXIT LIST
	TCAM MESSAGE QUEUE INTERFACE			
20 (14)	ADDRESS	4	DCBTRMAD(0)	– ADDRESS OF USER–PROVIDED AREA IN WHICH THE TERMINAL NAME

Offset	Type/Value	Len	Name (Dim)	Description
20 (14)	SIGNED	1	DCBBUFRQ	– NUMBER OF BUFFERS TO BE FILLED FROM THE DIRECT ACCESS QUEUE
21 (15)	ADDRESS	3	DCBTRMA	– SAME AS DCBTRMAD ABOVE
24 (18)	SIGNED	2	DCBSOWA	– SIZE OF USER–PROVIDED WORK AREA
26 (1A)	BITSTRING	2	DCBDSORG(0)	– DATA SET ORGANIZATION
26 (1A)	BITSTRING	1	DCBDSRG1	– FIRST BYTE OF DCBDSORG
27 (1B)	BITSTRING	1	DCBDSRG2	– SECOND BYTE OF DCBDSORG
28 (1C)	ADDRESS	4	DCBIOBAD	– BASE FOR ADDRESSING IOB'S
28 (1C)	ADDRESS	4	DCBSEGAD	– ADDRESS OF CURRENT SEGMENT
32 (20)	SIGNED	1	DCBTHRES	– FOR NON–REUSABLE MESSAGE QUEUE RECORDS, PERCENTAGE OF NON–REUSABLE DISK MESSAGE QUEUE RECORDS TO BE USED BEFORE A FLUSH CLOSEDOWN OF THE SYSTEM IS INITIATED. FOR REUSABLE MESSAGE QUEUE RECORDS AND CHECKPOINT RECORDS, THIS FIELD IS RESERVED
32 (20)	ADDRESS	4	DCBEODAD	– ADDRESS OF USER–PROVIDED ROUTINE
36 (24)	ADDRESS	4	DCBEXLST(0)	– ADDRESS OF EXIT LIST
36 (24)	CHARACTER	1	DCBRECFM	– RECORD FORMAT
	. . . . . .1.		DCBRECR	″X'02'″– RECORD
	. . . . .1. .		DCBRECG	″X'04'″– MESSAGE
	. . . . 1. . .		DCBRECS	″X'08'″– SEGMENT
37 (25)	ADDRESS	3	DCBEXLSA	– ADDRESS OF EXIT LIST
20 (14)	ADDRESS	4	DCBDUMPD(0)	– FULLWORD LABEL
20 (14)	BITSTRING	1	DCBUNITN	– NUMBER OF UNITS FOR READ FOLLOWING ATTENTION.
21 (15)	ADDRESS	3	DCBDUMPA	– ADDRESS OF THE DCB USED FOR DUMPING THE 3705
24 (18)	ADDRESS	1		– RESERVED
25 (19)	BITSTRING	1	DCBTRSTA	– STATUS BYTE. WHEN SET TO 1, THE INDICATORS HAVE THE SPECIFIED MEANING
	1.		DCBAUTOI	″DCBBIT0″– IPLAUTO=YES WAS SPECIFIED IN THE DCB
	.1.		DCBAUTOD	″DCBBIT1″– DMPAUTO=YES WAS SPECIFIED IN THE DCB MACRO.
	. .1.		DCBINITL	″DCBBIT2″– BRINGUP=YES WAS SPECIFIED IN THE DCB MACRO.
	. . .1		DCBRSTRT	″DCBBIT3″– RESTART IS IN PROCESS
	. . . . 1. . .		DCBIPLED	″DCBBIT4″– 3705 HAS BEEN IPL'D.
	. . . . .1. .		DCBBAKUP	″DCBBIT5″– BACKUP=YES WAS SPECIFIED IN THE DCB MACRO.
	. . . . . .1.		DCBNIDLE	″DCBBIT6″– IDLE=NO WAS SPECIFIED IN THE OPEN MACRO OR WAS IMPLIED BY DEFAULT
	1		DCBCHNGL	″DCBBIT7″– IPL TEXT HAS BEEN CHANGED
26 (1A)	SIGNED	2	DCBDSORG	
28 (1C)	ADDRESS	4	DCBIOBAD	
32 (20)	ADDRESS	4	DCBRNCKD(0)	– FULLWORD LABEL
32 (20)	BITSTRING	1		– RESERVED
33 (21)	ADDRESS	3	DCBRNCKA	– ADDRESS OF THE DCB USED TO RETAIN INCIDENT CHECKPOINT RECORDS GENERATED BY THE 3705.
36 (24)	ADDRESS	4	DCBEXLST(0)	
36 (24)	BITSTRING	1	DCBEIOBX	
37 (25)	ADDRESS	3	DCBEXLSA	
	FOUNDATION BEFORE OPEN			

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Offset	Type/Value	Len	Name (Dim)	Description
40 (28)	CHARACTER	8	DCBDDNAM	– NAME ON THE DD STATEMENT WHICH DEFINES THE DATA SET ASSOCIATED WITH THIS DCB
48 (30)	BITSTRING	1	DCBOFLGS	– FLAGS USED BY OPEN ROUTINE
	1.		DCBOFLWR	″DCBBIT0″– IF ZERO, LAST I/O OPERATION WAS READ OR POINT. IF ONE, LAST I/O OPERATION WAS WRITE.
	1.		DCBOFIOD	″DCBBIT0″– DATA SET IS BEING OPENED FOR INPUT OR OUTPUT (BDAM)
	.1.		DCBOFLRB	″DCBBIT1″– LAST I/O OPERATION WAS IN READ BACKWARD MODE
	. .1.		DCBOFEOV	″DCBBIT2″– SET TO 1 BY EOV WHEN IT CALLS CLOSE ROUTINE FOR CONCATENATION OF DATA SETS WITH UNLIKE ATTRIBUTES
	. . .1		DCBOFOPN	″DCBBIT3″– AN OPEN HAS BEEN SUCCESSFULLY COMPLETED
	. . . . 1. . .		DCBOFPPC	″DCBBIT4″– SET TO 1 BY PROBLEM PROGRAM TO INDICATE A CONCATENATION OF UNLIKE ATTRIBUTES
	. . . . .1. .		DCBOFTM	″DCBBIT5″– TAPE MARK HAS BEEN READ
	. . . . . .1.		DCBOFUEX	″DCBBIT6″– SET TO 0 BY AN I/O SUPPORT FUNCTION WHEN THAT FUNCTION TAKES A USER EXIT. SET TO 1 ON RETURN FROM USER EXIT TO THE I/O SUPPORT FUNCTION WHICH TOOK THE EXIT.
	1		DCBOFIOF	″DCBBIT7″– SET TO 1 BY AN I/O SUPPORT FUNCTION IF DCB IS TO BE PROCESSED BY THAT FUNCTION
49 (31)	BITSTRING	1	DCBIFLG	– FLAGS USED BY IOS IN COMMUNICATING ERROR CONDITIONS AND IN DETERMINING CORRECTIVE PROCEDURES
	11.		DCBIBEC	″DCBBIT0+DCBBIT1″– ERROR CORRECTION INDICATOR
	. . . . .		DCBIFNEP	″X'00'″– NOT IN ERROR PROCEDURE
	.1.		DCBEX	″DCBBIT1″–ERROR CORRECTION OR IOS PAGE FIX IN PROCESS
	11.		DCBIFPEC	″DCBBIT0+DCBBIT1″– PERMANENT ERROR CORRECTION
	. .11		DCBIBPCT	″DCBBIT2+DCBBIT3″– PRINTER CARRIAGE TAPE PUNCH INDICATOR
	. .1.		DCBIFC9	″DCBBIT2″– CHANNEL 9 PRINTER CARRIAGE TAPE PUNCH SENSED
	. . .1		DCBIFC12	″DCBBIT3″– CHANNEL 12 PRINTER CARRIAGE TAPE PUNCH SENSED
	. . . . 11. .		DCBIBIOE	″DCBBIT4+DCBBIT5″– IOS ERROR ROUTINE USE INDICATOR
	. . . . .		DCBIFER	″X'00'″– ALWAYS USE I/O SUPERVISOR ERROR ROUTINE
	. . . . .1. .		DCBIFNE1	″DCBBIT5″– NEVER USE I/O SUPERVISOR ERROR ROUTINE
	. . . . .1. .		DCBIFTIM	″DCBBIT5″– TEST IOS MASK (IMSK) FOR ERROR PROCEDURE (BTAM)
	. . . . 1. . .		DCBIFNE2	″DCBBIT4″– NEVER USE I/O SUPERVISOR ERROR ROUTINE
	. . . . 11. .		DCBIFNE3	″DCBBIT4+DCBBIT5″– NEVER USE I/O SUPERVISOR ERROR ROUTINE
50 (32)	BITSTRING	2	DCBMACR(0)	– MACRO INSTRUCTION REFERENCE

Offset	Type/Value	Len	Name (Dim)	Description
50 (32)	BITSTRING	1	DCBMACR1	–FIRST BYTE OF DCBMACR
	1.		DCBMRECP	″DCBBIT0″– EXECUTE CHANNEL PROGRAM (EXCP)– – – ALWAYS ZERO (BSAM, QSAM, BPAM, BISAM, QISAM, BDAM) – – – RESERVED (QT)
	.1.		DCBMRFE	″DCBBIT1″– FOUNDATION EXTENSION IS PRESENT (EXCP)
	.1.		DCBMRGET	″DCBBIT1″– GET (QSAM, QISAM, TCAM)
	.1.		DCBMRPTQ	″DCBBIT1″– PUT FOR MESSAGE GROUP (QTAM) – – – ALWAYS ZERO (BSAM, BPAM, BISAM, BDAM) – – – RESERVED (BTAM)
	. .1.		DCBMRAPG	″DCBBIT2″– APPENDAGES ARE REQUIRED (EXCP)
	. .1.		DCBMRRD	″DCBBIT2″– READ (BSAM, BPAM, BISAM, BDAM, BTAM)
	. .1.		DCBMRWRQ	″DCBBIT2″– WRITE FOR LINE GROUP (QTAM) – – – ALWAYS ZERO (QSAM, QISAM)
	. . .1		DCBMRCI	″DCBBIT3″– COMMON INTERFACE (EXCP)
	. . .1		DCBMRMVG	″DCBBIT3″– MOVE MODE OF GET (QSAM, QISAM)
	. . .1		DCBMRRDK	″DCBBIT3″– KEY SEGMENT WITH READ (BDAM) – – – ALWAYS ZERO (BISAM) – – – RESERVED (BSAM, BPAM, QTAM, BTAM)
	. . . . 1. . .		DCBMRLCG	″DCBBIT4″– LOCATE MODE OF GET (QSAM, QISAM)
	. . . . 1. . .		DCBMRRDI	″DCBBIT4″– ID ARGUMENT WITH READ (BDAM) – – – ALWAYS ZERO (BISAM) – — RESERVED (EXCP, BSAM, BPAM, QTAM, BTAM)
	. . . . .1. .		DCBMRABC	″DCBBIT5″– USER'S PROGRAM MAINTAINS ACCURATE BLOCK COUNT (EXCP)
	. . . . .1. .		DCBMRPT1	″DCBBIT5″– POINT (WHICH IMPLIES NOTE) (BSAM, BPAM)
	. . . . .1. .		DCBMRSBG	″DCBBIT5″– SUBSTITUTE MODE OF GET (QSAM)
	. . . . .1. .		DCBMRDBF	″DCBBIT5″–DYNAMIC BUFFERING (BISAM, BDAM) —— ALWAYS ZERO (QISAM) —— RESERVED (QTAM, BTAM)
	. . . . . .1.		DCBPGFXA	″DCBBIT6″– PAGE FIX APPENDAGE IS SPECIFIED (EXCP)
	. . . . . .1.		DCBMRCRL	″DCBBIT6″– CNTRL (BSAM, QSAM)
	. . . . . .1.		DCBMRCHK	″DCBBIT6″– CHECK (BISAM)
	. . . . . .1.		DCBMRRDX	″DCBBIT6″– READ EXCLUSIVE (BDAM) – – – RESERVED (BPAM, QISAM, QTAM, BTAM)
	1		DCBMRDMG	″DCBBIT7″– DATA MODE OF GET (QSAM)
	1		DCBMRCK	″DCBBIT7″– CHECK (BDAM) – – – RESERVED (EXCP, BSAM, BPAM, BISAM, QISAM, QTAM, BTAM)
51 (33)	BITSTRING	1	DCBMACR2	– SECOND BYTE OF DCBMACR
	1.		DCBMRSTL	″DCBBIT0″– SETL (QISAM) – – – ALWAYS ZERO (BSAM, QSAM BPAM, BISAM, BDAM) – – – RESERVED (EXCP, QTAM, BTAM)
	.1.		DCBMRPUT	″DCBBIT1″– PUT (QSAM, TCAM) – PUT OR PUTX (QISAM)
	.1.		DCBMRGTQ	″DCBBIT1″– GET FOR MESSAGE GROUP (QTAM) – – – ALWAYS ZERO (BSAM, BPAM, BISAM, BDAM) – – – RESERVED (EXCP, BTAM)
	. .1.		DCBMRWRT	″DCBBIT2″– WRITE (BSAM, BPAM, BISAM, BDAM, BTAM)
	. .1.		DCBMRRDQ	″DCBBIT2″– READ FOR LINE GROUP (QTAM) – – – ALWAYS ZERO (QSAM, QISAM) – – – RESERVED (EXCP)

Offset	Type/Value	Len	Name (Dim)	Description
	. . .1		DCBMRMVP	″DCBBIT3″– MOVE MODE OF PUT (QSAM, QISAM)
	. . .1		DCBMRWRK	″DCBBIT3″– KEY SEGMENT WITH WRITE (BDAM) – – – ALWAYS ZERO (BISAM) – – –RESERVED (EXCP, BSAM, BPAM, QTAM, BTAM)
	. . . . 1. . .		DCBMR5WD	″DCBBIT4″– FIVE–WORD DEVICE INTERFACE (EXCP)
	. . . . 1. . .		DCBMRLDM	″DCBBIT4″– LOAD MODE BSAM (CREATE BDAM DATA SET) (BSAM)
	. . . . 1. . .		DCBMRLCP	″DCBBIT4″– LOCATE MODE OF PUT (QSAM, QISAM)
	. . . . 1. . .		DCBMRIDW	″DCBBIT4″– ID ARGUMENT WITH WRITE (BDAM) – – – ALWAYS ZERO (BISAM) —— RESERVED (BPAM, QTAM, BTAM)
	. . . . .1. .		DCBMR4WD	″DCBBIT5″– FOUR-WORD DEVICE INTERFACE (EXCP)
	. . . . .1. .		DCBMRPT2	″DCBBIT5″– POINT (WHICH IMPLIES NOTE) (BSAM, BPAM)
	. . . . .1. .		DCBMRTMD	″DCBBIT5″– SUBSTITUTE MODE (QSAM)
	. . . . .1. .		DCBMRUIP	″DCBBIT5″– UPDATE IN PLACE (PUTX) (QISAM) —— ALWAYS ZERO (BISAM) —— RESERVED (BDAM, QTAM, BTAM)
	. . . . . .1.		DCBMR3WD	″DCBBIT6″– THREE–WORD DEVICE INTERFACE (EXCP)
	. . . . . .1.		DCBMRCTL	″DCBBIT6″– CNTRL (BSAM, QSAM)
	. . . . . .1.		DCBMRSTK	″DCBBIT6″– SETL BY KEY (QISAM)
	. . . . . .1.		DCBMRAWR	″DCBBIT6″– ADD TYPE OF WRITE (BDAM) – – – ALWAYS ZERO (BISAM) – – – RESERVED (BPAM, QTAM, BTAM)
	1		DCBMR1WD	″DCBBIT7″– ONE–WORD DEVICE INTERFACE (EXCP)
	1		DCBMRSWA	″DCBBIT7″– USER'S PROGRAM HAS PROVIDED A SEGMENT WORK AREA POOL (BSAM CREATE BDAM, BDAM)
	1		DCBMRDMD	″DCBBIT7″– DATA MODE (QSAM)
	1		DCBMRSTI	″DCBBIT7″– SETL BY ID (QISAM) – – – ALWAYS ZERO (BISAM) – – – RESERVED (BPAM, QTAM, BTAM)
	FOUNDATION AFTER OPEN			
40 (28)	ADDRESS	2	DCBTIOT	OFFSET FROM TIOT ORIGIN TO TIOELNGH FIELD IN TIOT ENTRY FOR DD STATEMENT ASSOCIATED WITH THIS DCB
42 (2A)	BITSTRING	2	DCBMACRF(0)	– SAME AS DCBMACR BEFORE OPEN
42 (2A)	BITSTRING	1	DCBMACF1	– FIRST BYTE OF DCBMACRF
43 (2B)	BITSTRING	1	DCBMACF2	– SECOND BYTE OF DCBMACRF
44 (2C)	ADDRESS	4	DCBDEBAD(0)	– ADDRESS OF ASSOCIATED DEB
44 (2C)	BITSTRING	1	DCBIFLGS	– SAME AS DCBIFLG BEFORE OPEN
	11.		DCBIFEC	″DCBBIT0+DCBBIT1″– ERROR CORRECTION INDICATOR
	. .11		DCBIFPCT	″DCBBIT2+DCBBIT3″– PRINTER CARRIAGE TAPE PUNCH INDICATOR
	. . . . 11. .		DCBIFIOE	″DCBBIT4+DCBBIT5″– IOS ERROR ROUTINE USE INDICATOR
	. . . . . .1.		DCBIFLDT	″DCBBIT6″– POSSIBLE LOST DATA CONDITION SUCH AS FOR A PRINTER
45 (2D)	ADDRESS	3	DCBDEBA	– ADDRESS OF ASSOCIATED DEB
48 (30)	ADDRESS	4	DCBGET(0)	– ADDRESS OF GET MODULE

Offset	Type/Value	Len	Name (Dim)	Description
48 (30)	ADDRESS	4	DCBPUT(0)	– ADDRESS OF PUT MODULE
48 (30)	BITSTRING	1	DCBOFLG1	SAME AS DCBOFLGS BEFORE OPEN
49 (31)	ADDRESS	3	DCBGETA(0)	ADDRESS OF GET MODULE
49 (31)	ADDRESS	3	DCBPUTA	ADDRESS OF PUT MODULE
	TCAM LINE GROUP EXTENSION			
	3705 EXTENSION			
48 (30)	ADDRESS	4	DCBSCTAB(0)	– ADDRESS OF SPECIAL CHARACTERS TABLE (SCT)
48 (30)	BITSTRING	1		– DCBOFLGS – FLAGS USED BY OPEN ROUTINE
49 (31)	ADDRESS	3	DCBSCTAD	– ADDRESS OF SPECIAL CHARACTERS TABLE (SCT)
52 (34)	SIGNED	1	DCBILCT	– COUNT OF INVITATION LISTS
53 (35)	SIGNED	1	DCBUNTCT	– BEFORE OPEN – NUMERICAL VALUE OF SCT. AFTER OPEN – COUNT OF UNITS FOR 1 BUFFER.
54 (36)	SIGNED	2	DCBBUFSI	– SIZE OF ALL BUFFERS USED FOR THIS LINE GROUP
56 (38)	CHARACTER	4	DCBRESER(0)	– NUMBER OF RESERVED BYTES IN BUFFERS
56 (38)	SIGNED	1	DCBRESB1	– NUMBER OF BYTES RESERVED IN THE BUFFER RECEIVING FIRST INCOMING SEGMENT OF A MESSAGE
57 (39)	SIGNED	1	DCBRESB2	– NUMBER OF BYTES RESERVED IN ALL BUFFERS EXCEPT THE ONE CONTAINING FIRST SEGMENT OF A MESSAGE
58 (3A)	BITSTRING	2		– RESERVED
	THE FOLLOWING 4 BYTES MAY BE REPEATED 'N' TIMES			
60 (3C)	ADDRESS	4	DCBINVLI(0)	– ADDRESS OF INVITATION LIST
60 (3C)	BITSTRING	1	DCBINVCI	– TYPE OF COMMUNICATION INTERFACE FOR 2701 DATA ADAPTER UNIT
	. .1.		DCBINVB1	″DCBBIT2″– IF ZERO, UNIT (A,) IF ONE, UNIT (B,)
	. . . . 1. . .		DCBINVB2	″DCBBIT4″–IF ZERO, UNIT (,A) IF ONE, UNIT (,B)
61 (3D)	ADDRESS	3	DCBINVLA	– ADDRESS OF INVITATION LIST
	TCAM MESSAGE QUEUE INTERFACE			
52 (34)	BITSTRING	1	DCBOPTCD	– OPTION CODES
	1.		DCBOPTWP	″DCBBIT0″– SOURCE OR DESTINATION NAME PRECEDES MESSAGE (AFTER CONTROL BYTE) (TCAM PROCESS QUEUE)
	.1.		DCBOPTUM	″DCBBIT1″– WORK UNIT IS A MESSAGE. DEFAULT WORK UNIT IS A RECORD. (TCAM PROCESS QUEUE)
	. .1.		DCBOPTCB	″DCBBIT2″– CONTROL BYTE PRECEDES WORK UNIT (TCAM PROCESS QUEUE)
	. .1.		DCBOPTCP	″DCBBIT2″– CHECKPOINT DATA SET
	. . . . . .1.		DCBOPTIM	″DCBBIT6″– NON–REUSABLE MESSAGE QUEUE DATA SET
	1		DCBOPTRM	″DCBBIT7″– REUSABLE MESSAGE QUEUE DATA SET
53 (35)	BITSTRING	9		– RESERVED
62 (3E)	SIGNED	2	DCBBLKSI	– BLOCK SIZE

Cross-Reference

Name	Hex Offset	Hex Value	Level
DCBACBM	1B	8	2
DCBAUTOD	19	40	2
DCBAUTOI	19	80	2
DCBBAKUP	19	4	2
DCBBCKUA	11		2
DCBBCKUP	10		2
DCBBFIN	14	F0	2
DCBBFOUT	14	F	2
DCBBIT0	0	80	2
DCBBIT1	0	40	2
DCBBIT2	0	20	2
DCBBIT3	0	10	2
DCBBIT4	0	8	2
DCBBIT5	0	4	2
DCBBIT6	0	2	2
DCBBIT7	0	1	2
DCBBLKSI	3E		2
DCBBQFLG	10		2
DCBBQIAM	10	20	2
DCBBQWRE	10	8	2
DCBBQWRS	10	10	2
DCBBQWRU	10	40	2
DCBBUFIN	14		2
DCBBUFMA	1C		2
DCBBUFOU	14		2
DCBBUFRQ	14		2
DCBBUFSI	36		2
DCBCHNGL	19	1	2
DCBCLPS	14		2
DCBCLPSA	15		2
DCBCPE	20	2	2
DCBCPR	20	4	2
DCBCPRI	20		2
DCBCPS	20	1	2
DCBDDNAM	28		2
DCBDEBA	2D		2
DCBDEBAD	2C		2
DCBDEVTP	1C		2
DCBDSGCX	1A	10	2
DCBDSGDA	1A	20	2
DCBDSGGS	1B	80	2
DCBDSGIS	1A	80	2
DCBDSGPO	1A	2	2
DCBDSGPS	1A	40	2
DCBDSGTQ	1B	20	2
DCBDSGTR	1B	4	2
DCBDSGTX	1B	40	2
DCBDSGU	1A	1	2
DCBDSORG	1A		2
DCBDSRG1	1A		2
DCBDSRG2	1B		2
DCBDUMPA	15		2
DCBDUMPD	14		2

Name	Hex Offset	Hex Value	Level
DCBEIOBX	24		2
DCBEODAD	20		2
DCBEX	31	40	2
DCBEXLSA	25		2
DCBEXLST	24		2
DCBGET	30		2
DCBGETA	31		2
DCBIBEC	31	C0	2
DCBIBIOE	31	C	2
DCBIBPCT	31	30	2
DCBIFC12	31	10	2
DCBIFC9	31	20	2
DCBIFEC	2C	C0	2
DCBIFER	31	0	2
DCBIFIOE	2C	C	2
DCBIFLDT	2C	2	2
DCBIFLG	31		2
DCBIFLGS	2C		2
DCBIFNEP	31	0	2
DCBIFNE1	31	4	2
DCBIFNE2	31	8	2
DCBIFNE3	31	C	2
DCBIFPCT	2C	30	2
DCBIFPEC	31	C0	2
DCBIFTIM	31	4	2
DCBILCT	34		2
DCBINITL	19	20	2
DCBINTVL	18		2
DCBINVB1	3C	20	2
DCBINVB2	3C	8	2
DCBINVCI	3C		2
DCBINVLA	3D		2
DCBINVLI	3C		2
DCBIOBAA	1D		2
DCBIOBAD	1C		2
DCBIPLED	19	8	2
DCBIPLTX	8		2
DCBLCBA	21		2
DCBLCBAD	20		2
DCBLNGTQ	3E	40	2
DCBLNGTR	36	38	2
DCBLNGTX	3D	40	2
DCBLNGXE	33	34	2
DCBMACF1	2A		2
DCBMACF2	2B		2
DCBMACR	32		2
DCBMACRF	2A		2
DCBMACR1	32		2
DCBMACR2	33		2
DCBMH	15		2
DCBMHA	14		2
DCBMRABC	32	4	2
DCBMRAPG	32	20	2
DCBMRAWR	33	2	2

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Name	Hex Offset	Hex Value	Level
DCBMRCHK	32	2	2
DCBMRCI	32	10	2
DCBMRCK	32	1	2
DCBMRCRL	32	2	2
DCBMRCTL	33	2	2
DCBMRDBF	32	4	2
DCBMRDMD	33	1	2
DCBMRDMG	32	1	2
DCBMRECP	32	80	2
DCBMRFE	32	40	2
DCBMRGET	32	40	2
DCBMRGTQ	33	40	2
DCBMRIDW	33	8	2
DCBMRLCG	32	8	2
DCBMRLCP	33	8	2
DCBMRLDM	33	8	2
DCBMRMVG	32	10	2
DCBMRMVP	33	10	2
DCBMRPTQ	32	40	2
DCBMRPT1	32	4	2
DCBMRPT2	33	4	2
DCBMRPUT	33	40	2
DCBMRRD	32	20	2
DCBMRRDI	32	8	2
DCBMRRDK	32	10	2
DCBMRRDQ	33	20	2
DCBMRRDX	32	2	2
DCBMRSBG	32	4	2
DCBMRSTI	33	1	2
DCBMRSTK	33	2	2
DCBMRSTL	33	80	2
DCBMRSWA	33	1	2
DCBMRTMD	33	4	2
DCBMRUIP	33	4	2
DCBMRWRK	33	10	2
DCBMRWRQ	32	20	2
DCBMRWRT	33	20	2
DCBMR1WD	33	1	2
DCBMR3WD	33	2	2
DCBMR4WD	33	4	2
DCBMR5WD	33	8	2
DCBNIDLE	19	2	2
DCBOFEOV	30	20	2
DCBOFIOD	30	80	2
DCBOFIOF	30	1	2
DCBOFLGS	30		2
DCBOFLG1	30		2
DCBOFLRB	30	40	2
DCBOFLWR	30	80	2
DCBOFOPN	30	10	2
DCBOFPPC	30	8	2
DCBOFTM	30	4	2
DCBOFUEX	30	2	2
DCBOPTCB	34	20	2

Name	Hex Offset	Hex Value	Level
DCBOPTCD	34		2
DCBOPTCP	34	20	2
DCBOPTIM	34	2	2
DCBOPTRM	34	1	2
DCBOPTUM	34	40	2
DCBOPTWP	34	80	2
DCBPCI	19		2
DCBPCIA1	19	20	2
DCBPCIA2	19	10	2
DCBPCIN1	19	8	2
DCBPCIN2	19	4	2
DCBPCIR1	19	2	2
DCBPCIR2	19	1	2
DCBPCIX1	19	80	2
DCBPCIX2	19	40	2
DCBPGFXA	32	2	2
DCBPUT	30		2
DCBPUTA	31		2
DCBRECFM	24		2
DCBRECG	24	4	2
DCBRECR	24	2	2
DCBRECS	24	8	2
DCBRESB1	38		2
DCBRESB2	39		2
DCBRESER	38		2
DCBRNCKA	21		2
DCBRNCKD	20		2
DCBRSTRT	19	10	2
DCBSCTAB	30		2
DCBSCTAD	31		2
DCBSEGAD	1C		2
DCBSOWA	18		2
DCBTHRES	20		2
DCBTIOT	28		2
DCBTRANA	20		2
DCBTRANS	21		2
DCBTRMA	15		2
DCBTRMAD	14		2
DCBTRSTA	19		2
DCBUNITN	14		2
DCBUNTCT	35		2
DCBWTEOM	11		2
DCBWTEOT	12		2
DCBWTPAD	13		2

DCB for GAM

Common Name: Data Control Block for GAM
Macro Name: DCBD
DSECT Name: IHADCB
Owning Component: Data Management, subcomponent OPEN/CLOSE/EOV
Eye-Catcher ID: None
Subpool and Key: Problem program subpool and key

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Size: 52 bytes
Created by: Problem program
Pointed to by:

- DEBDCBAD field of the DEB data area
- IOBDCBPT field of the IOB data area

Serialization: None
Function: This data control block (DCB) is used by the graphics access method (GAM) routine. The GAM access method uses the common DCB interface and foundation sections used by all other access method routines, although some differences might exist. An interface section with information about a particular graphics device precedes the common section.

Offset	Type/Value	Len	Name (Dim)	Description
DATA CONTROL BLOCK DEFINITION				
	X'80'		DCBBIT0	″128″
	X'40'		DCBBIT1	″64″
	X'20'		DCBBIT2	″32″
	X'10'		DCBBIT3	″16″
	X'8'		DCBBIT4	″8″
	X'4'		DCBBIT5	″4″
	X'2'		DCBBIT6	″2″
	X'1'		DCBBIT7	″1″
	GRAPHIC DEVICE INTERFACE			
0 (0)	BITSTRING	12		– RESERVED
12 (C)	ADDRESS	2	DCBBRSA	– BUFFER RESTART ADDRESS. BLANK BEFORE EXECUTION OF SECOND I/O OPERATION
14 (E)	CHARACTER	1	DCBGTYPE	– TYPE OF BUFFER MANAGEMENT AND ATTENTION HANDLING
	. . . . .		DCBGTEXP	″X'00'″– EXPRESS
	1		DCBGTBAS	″X'01'″– BASIC
	. . . . . .1.		DCBGTNBM	″X'02'″– DEFAULT – NO BUFFER MANAGEMENT
15 (F)	BITSTRING	1		– RESERVED
16 (10)	ADDRESS	2	DCBBFRST	– BLANK BEFORE EXECUTION OF OPEN ROUTINE. STARTING ADDRESS FOR BUFFER AFTER EXECUTION OF OPEN ROUTINE
18 (12)	SIGNED	2	DCBBFRSZ	– BLANK BEFORE EXECUTION OF OPEN ROUTINE. SIZE OF BUFFER AFTER EXECUTION OF OPEN ROUTINE.
	COMMON INTERFACE			
20 (14)	BITSTRING	6		– RESERVED
26 (1A)	BITSTRING	2	DCBDSORG (0)	– DATA SET ORGANIZATION BEING USED
26 (1A)	BITSTRING	1	DCBDSRG1	– FIRST BYTE OF DCBDSORG
	1.		DCBDSGIS	″DCBBIT0″– IS – INDEXED SEQUENTIAL ORGANIZATION
	.1.		DCBDSGPS	″DCBBIT1″– PS – PHYSICAL SEQUENTIAL ORGANIZATION
	. .1.		DCBDSGDA	″DCBBIT2″– DA – DIRECT ORGANIZATION
	. . .1		DCBDSGCX	″DCBBIT3″– CX – BTAM OR QTAM LINE GROUP
	. . . . . .1.		DCBDSGPO	″DCBBIT6″– PO – PARTITIONED ORGANIZATION
	1		DCBDSGU	″DCBBIT7″– U – UNMOVABLE, THE DATA CONTAINS LOCATION DEPENDENT INFORMATION
27 (1B)	BITSTRING	1	DCBDSRG2	– SECOND BYTE OF DCBDSORG

Offset	Type/Value	Len	Name (Dim)	Description
	1.		DCBDSGGS	″DCBBIT0″–GS – GRAPHICS ORGANIZATION
	.1.		DCBDSGTX	″DCBBIT1″– TX – TCAM LINE GROUP
	. .1.		DCBDSGTQ	″DCBBIT2″– TQ – TCAM MESSAGE QUEUE
	. . . . 1. . .		DCBACBM	″DCBBIT4″– ACCESS METHOD CONTROL BLOCK
	. . . . .1. .		DCBDSGTR	″DCBBIT5″– TR – TCAM 3705
28 (1C)	ADDRESS	4	DCBIOBAD	– BLANK BEFORE EXECUTION OF OPEN ROUTINE. ADDRESS OF STANDARD FIELDS OF FIRST IOB AFTER EXECUTION OF OPEN ROUTINE
	FOUNDATION EXTENSION			
32 (20)	ADDRESS	4	DCBPOLST (0)	– ADDRESS OF AREA WHERE A DCB LIST IS TO BE CONSTRUCTED FOR POLLING PURPOSES
32 (20)	SIGNED	1	DCBGNCP	– NUMBER OF I/O INSTRUCTIONS TO BE ISSUED BEFORE A WAIT MACRO INSTRUCTION
33 (21)	ADDRESS	3	DCBPOLSA	– SAME AS DCBPOLST ABOVE
36 (24)	ADDRESS	4	DCBEXLST (0)	– ADDRESS OF USER'S EXIT LIST
36 (24)	BITSTRING	1		– RESERVED
37 (25)	ADDRESS	3	DCBEXLSA	– ADDRESS OF USER'S EXIT LIST
	FOUNDATION BEFORE OPEN			
40 (28)	CHARACTER	8	DCBDDNAM	– 8–BYTE NAME FROM DD STATEMENT THAT DEFINES DATA SET ASSOCIATED WITH THIS DCB
48 (30)	BITSTRING	1	DCBOFLG	– FLAGS USED BY OPEN ROUTINE
	1.		DCBOFGRW	″DCBBIT0″– IF ZERO, LAST I/O OPERATION WAS GREAD. IF ONE, LAST I/O OPERATION WAS GWRITE.
1	. .1.		DCBOFEOV	″DCBBIT2″– SET TO 1 BY EOV WHEN IT CALLS CLOSE ROUTINE FOR CONCATENATION OF DATA SETS WITH UNLIKE ATTRIBUTES
	. . .1		DCBOFOPN	″DCBBIT3″– AN OPEN HAS BEEN SUCCESSFULLY COMPLETED
1	. . . . 1. . .		DCBOFPPC	″DCBBIT4″– SET TO 1 BY PROBLEM PROGRAM TO INDICATE A CONCATENATION OF UNLIKE ATTRIBUTES
	. . . . .1. .		DCBOFTM	″DCBBIT5″– TAPE MARK HAS BEEN READ
1	. . . . . .1.		DCBOFUEX	″DCBBIT6″– SET TO 0 BY AN I/O SUPPORT FUNCTION WHEN THAT FUNCTION TAKES A USER EXIT. SET TO 1 ON RETURN FROM USER EXIT TO THE I/O SUPPORT FUNCTION WHICH TOOK THE EXIT.
	1		DCBOFIOF	″DCBBIT7″– SET TO 1 BY AN I/O SUPPORT FUNCTION IF DCB IS TO BE PROCESSED BY THAT FUNCTION
49 (31)	BITSTRING	1	DCBIFLG	– SET TO ZERO BY GRAPHIC ROUTINES BUT USED BY IOS IN COMMUNICATING ERROR CONDITIONS AND IN DETERMINING CORRECTIVE PROCEDURES
50 (32)	BITSTRING	2	DCBMACR (0)	– MACRO INSTRUCTION REFERENCE
50 (32)	BITSTRING	1	DCBMACR1	– FIRST BYTE OF DCBMACR
	. .1.		DCBMRRD	″DCBBIT2″– READ
	. . . . . .1.		DCBMRCRL	″DCBBIT6″– CNTRL
51 (33)	BITSTRING	1	DCBMACR2	– SECOND BYTE OF DCBMACR
	. .1.		DCBMRWRT	″DCBBIT2″– WRITE
	. . . . . .1.		DCBMRCTL	″DCBBIT6″– CNTRL
	FOUNDATION AFTER OPEN			
40 (28)	ADDRESS	2	DCBTIOT	– OFFSET FROM TIOT ORIGIN TO DD ENTRY ASSOCIATED WITH THIS DCB
42 (2A)	BITSTRING	2	DCBMACRF (0)	– SAME AS DCBMACR BEFORE OPEN

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Offset	Type/Value	Len	Name (Dim)	Description
42 (2A)	BITSTRING	1	DCBMACF1	– FIRST BYTE OF DCBMACRF
43 (2B)	BITSTRING	1	DCBMACF2	– SECOND BYTE OF DCBMACRF
44 (2C)	ADDRESS	4	DCBDEBAD (0)	– ADDRESS OF ASSOCIATED DEB
44 (2C)	BITSTRING	1	DCBIFLGS	– SAME AS DCBIFLG BEFORE OPEN
45 (2D)	ADDRESS	3	DCBDEBA	– ADDRESS OF ASSOCIATED DEB
48 (30)	ADDRESS	4	DCBGIOCR (0)	– ADDRESS OF GRAPHICS I/O CONTROL ROUTINE
48 (30)	BITSTRING	1	DCBOFLGS	– SAME AS DCBOFLG BEFORE OPEN
49 (31)	ADDRESS	3	DCBGIOCA	– ADDRESS OF GRAPHICS I/O CONTROL ROUTINE

Cross-Reference

Name	Hex Offset	Hex Value	Level
DCBACBM	1B	8	2
DCBBFRST	10		2
DCBBFRSZ	12		2
DCBBIT0	0	80	2
DCBBIT1	0	40	2
DCBBIT2	0	20	2
DCBBIT3	0	10	2
DCBBIT4	0	8	2
DCBBIT5	0	4	2
DCBBIT6	0	2	2
DCBBIT7	0	1	2
DCBBRSA	C		2
DCBDDNAM	28		2
DCBDEBA	2D		2
DCBDEBAD	2C		2
DCBDSGCX	1A	10	2
DCBDSGDA	1A	20	2
DCBDSGGS	1B	80	2
DCBDSGIS	1A	80	2
DCBDSGPO	1A	2	2
DCBDSGPS	1A	40	2
DCBDSGTQ	1B	20	2
DCBDSGTR	1B	4	2
DCBDSGTX	1B	40	2
DCBDSGU	1A	1	2
DCBDSORG	1A		2
DCBDSRG1	1A		2
DCBDSRG2	1B		2
DCBEXLSA	25		2
DCBEXLST	24		2
DCBGIOCA	31		2
DCBGIOCR	30		2
DCBGNCP	20		2
DCBGTBAS	E	1	2
DCBGTEXP	E	0	2
DCBGTNBM	E	2	2
DCBGTYPE	E		2
DCBIFLG	31		2
DCBIFLGS	2C		2

Name	Hex Offset	Hex Value	Level
DCBIOBAD	1C		2
DCBLNGGS	31	34	2
DCBMACF1	2A		2
DCBMACF2	2B		2
DCBMACR	32		2
DCBMACRF	2A		2
DCBMACR1	32		2
DCBMACR2	33		2
DCBMRCRL	32	2	2
DCBMRCTL	33	2	2
DCBMRRD	32	20	2
DCBMRWRT	33	20	2
DCBOFEOV	30	20	2
DCBOFGRW	30	80	2
DCBOFIOF	30	1	2
DCBOFLG	30		2
DCBOFLGS	30		2
DCBOFOPN	30	10	2
DCBOFPPC	30	8	2
DCBOFTM	30	4	2
DCBOFUEX	30	2	2
DCBPOLSA	21		2
DCBPOLST	20		2
DCBTIOT	28		2

IOB

Common Name: Input/Output Block

Macro Name: IEZIOB

DSECT Name: IOB (DSECT card precedes prefix); label, IOBSTDRD should be used in the USING statement for the standard section.

Eye-Catcher ID: None

Subpool and Key: Any subpool, user key

Size: Variable

Created by: Access method OPEN executor

Pointed to by:

- DCBIOBAD field of the DCB data area
- DCBIOBA field of the DCB data area
- IOBNIOBA field of the IOB data area
- RQEIOB field of the IOB data area (depending on access method used)
- QPLIOB field of the QPL data area
- TCBIOBRC field of the TCB data area (for first quiesced TCB)

Serialization: The user is responsible for serialization. The LOCAL lock is held during I/O interrupt processing. The way IOBs are chained and serialized depends on the access method used.

Function: The IOB is the communications medium between a routine that requests an I/O operation and the I/O supervisor's needs to execute the I/O operation.

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Note: This section includes fields that are added to the IOB by various access methods. The general use fields that are intended as programming interfaces are described in *z/OS DFSMSdfp Advanced Services*.

Offset	Type/Value	Len	Name (Dim)	Description
PREFIX SECTIONS OF THE IOB				
–16 (–10)	DBL WORD	8	IOBPREFX(0)	
QSAM, BSAM, BPAM PREFIX CHAINED SCHEDULING 16 BYTES				
–16 (–10)	DBL WORD	8	IOBQSAMC (0)	
–16 (–10)	DBL WORD	8	IOBBSAMC (0)	
–16 (–10)	DBL WORD	8	IOBBPAMC (0)	
–16 (–10)	BITSTRING	1	IOBCFLG1	– I/O INDICATORS
	1.		IOBV6CHN	″X'80'″– I/O CHAINED BIT SET BY IGG019V6
	.1.		IOBRSV02	″X'40',,CX'X'″RESERVED
	. .1.		IOBRSV03	″X'20',,CX'X'″RESERVED
	. . .1		IOBRSV04	″X'10',,CX'X'″RESERVED
	. . . . 1. . .		IOBPTST	″X'08'″– NOTE OR POINT OPERATION IS IN PROCESS.
	. . . . .1. .		IOBABAPP	″X'04'″– ERROR HAS BEEN PROCESSED ONCE BY ABNORMAL–END APPENDAGE ROUTINE.
	. . . . . .1.		IOBRSTCH	″X'02'″– RESTART CHANNEL
	1		IOBPCI	″X'01'″– SET WHEN A PROGRAM– CONTROLLED INTERRUPTION (PCI) OCCURS
–15 (–F)	BITSTRING	1	IOBRSV05	– RESERVED
–14 (–E)	CHARACTER	1	IOBCINOP	– OFFSET OF THE LAST I/O COMMAND FOR INPUT OPERATION (NOP CCW) FROM THE ORIGIN OF THE ICB
–13 (–D)	CHARACTER	1	IOBCONOP	– OFFSET OF THE LAST I/O COMMAND FOR AN OUTPUT OPERATION (NOP CCW) FROM THE ORIGIN OF THE ICB
–12 (–C)	SIGNED	4	IOBCECB	– EVENT CONTROL BLOCK USED BY BSAM OR QSAM. SHOWS THE STATUS OF THE I/O OPERATION.
–8 (–8)	ADDRESS	4	IOBCICB	– ADDRESS OF THE FIRST INTERRUPT CONTROL BLOCK (ICB) ON THE ICB QUEUE
–4 (–4)	ADDRESS	4	IOBCNOPA	– ADDRESS OF THE NOP COMMAND AT THE END OF THE QUEUE
QSAM, BSAM, BPAM PREFIX NORMAL SCHEDULING 8 BYTES				
–8 (–8)	DBL WORD	8	IOBQSAMN (0)	
–8 (–8)	DBL WORD	8	IOBBSAMN (0)	
–8 (–8)	DBL WORD	8	IOBBPAMN (0)	
–8 (–8)	ADDRESS	4	IOBNIOBA (0)	– ADDRESS OF THE NEXT IOB ASSOCIATED WITH ONE PARTICULAR DCB. THE IOB'S ARE CHAINED IN SEQUENTIAL ORDER.
–8 (–8)	BITSTRING	1	IOBNFLG1	– FLAG BYTE
	1.		IOBPRTOV	″X'80'″– PRTOV HAS OCCURRED (PRINTER DEVICES) (MDC331).
	1.		IOBSEGMT	″X'80'″– SEGMENTING OF A SPANNED RECORD IS IN PROCESS (QSAM LOCATE MODE, LOGICAL RECORD INTERFACE, UPDATE PROCESSING) (DIRECT ACCESS) (OS/VS2) (MDC332).
	.1.		IOBWRITE	″X'40'″– A WRITE OPERATION IS IN PROCESS.
	. .1.		IOBREAD	″X'20'″– A READ OPERATION IS IN PROCESS.

Offset	Type/Value	Len	Name (Dim)	Description
	. . .1		IOBUPDAT	″X'10'″– UPDATE FLAG. SET ON TOGETHER WITH BIT 1 OF THIS BYTE TO SHOW THAT THE BLOCK IS TO BE UPDATED. CAN ONLY OCCUR IF THE OPEN PARAMETER IS UPDAT.
	. . . . 1. . .		IOBBKSPC	″X'08'″– IOB BEING USED FOR BACKSPACE, CONTROL, OR NOTE/POINT OPERATION
	. . . . .1. .		IOBSPAN	″X'04'″– THE RECORD CURRENTLY BEING PROCESSED HAS MORE THAN ONE SEGMENT (QSAM LOCATE MODE, LOGICAL RECORD INTERFACE, UPDATE PROCESSING OF SPANNED RECORDS).
	. . . . . .1.		IOBUPERR	″X'02'″– UPDATE CHANNEL PROGRAM HAS BEEN SPLIT INTO TWO PARTS (MDC036).
	1		IOBFIRST	″X'01'″– THIS IS THE FIRST IOB ON CHAIN.
–7 (–7)	ADDRESS	3	IOBNIOBB	– ADDRESS OF THE NEXT IOB ASSOCIATED WITH ONE PARTICULAR DCB. THE IOBS ARE CHAINED IN SEQUENTIAL ORDER.
–4 (–4)	SIGNED	4	IOBNECB	– EVENT CONTROL BLOCK USED BY QSAM TO INDICATE THE STATUS OF THE I/O EVENT
	BDAM PREFIX 8 BYTES			
–8 (–8)	DBL WORD	8	IOBBDAM (0)	
–8 (–8)	ADDRESS	4	IOBDQADA (0)	– ADDRESS OF THE OTHER IOB REFERRED TO IN DESCRIPTION OF IOBDEQ BELOW
–8 (–8)	BITSTRING	1	IOBDEQIN	– DEQUEUE LOOP INDICATOR
	1.		IOBDEQ	″X'80'″– THIS IOB IS USING A TRACK THAT WAS DEQUEUED BY ANOTHER IOB, WHICH IS NOW WAITING TO DEQUEUE ANOTHER TRACK. THE OTHER IOB ENQUEUED ON TWO OR MORE TRACKS TO FIND SPACE IN WHICH TO WRITE/ADD A SPANNED RECORD. THE OTHER IOB REMAINED ENQUEUED UNTIL IT EITHER WROTE THE RECORD OR DETERMINED THAT THERE WAS ENOUGH CONTIGUOUS FREE SPACE ON THE TRACKS TO CONTAIN THE RECORD. AFTER THE OTHER IOB DEQUEUED THE CURRENT TRACK, THE DEQUEUING WAS INTERRUPTED BY THE NEED OF THIS IOB FOR THE CURRENT TRACK.
	.1.		IOBRSV07	″X'40',,C'X'″RESERVED
	. .1.		IOBRSV08	″X'20',,C'X'″RESERVED
	. . .1		IOBRSV09	″X'10',,C'X'″RESERVED
	. . . . 1. . .		IOBRSV10	″X'08',,C'X'″RESERVED
	. . . . .1. .		IOBRSV11	″X'04',,C'X'″RESERVED
	. . . . . .1.		IOBRSV12	″X'02',,C'X'″RESERVED
	1		IOBRSV13	″X'01',,C'X'″RESERVED
–7 (–7)	ADDRESS	3	IOBDQADB	– ADDRESS OF THE OTHER IOB REFERRED TO IN DESCRIPTION OF IOBDEQ ABOVE
–4 (–4)	ADDRESS	4	IOBSWAP	– ADDRESS OF THE SEGMENT WORK AREA USED BY THIS IOB TO READ OR WRITE A RECORD OF A FORMAT VS DATA SET
	GAM, QISAM PREFIX 4 BYTES			
–4 (–4)	SIGNED	4	IOBGQECB	– EVENT CONTROL BLOCK THAT IS WITHIN FIRST IOB ONLY (GAM) EVENT CONTROL BLOCK USED TO INDICATE STATUS OF AN I/O EVENT (QISAM)
	STANDARD SECTION OF THE IOB			
0 (0)	DBL WORD	8	IOBSTDRD (0)	

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Offset	Type/Value	Len	Name (Dim)	Description
0 (0)	BITSTRING	1	IOBFLAG1	– FLAG BYTE 1
	1.		IOBDATCH	″X'80'″– DATA CHAINING USED IN CHANNEL PROGRAM
	.1.		IOBCMDCH	″X'40'″– COMMAND CHAINING USED IN CHANNEL PROGRAM
	. .1.		IOBERRTN	″X'20'″– ERROR ROUTINE IS IN CONTROL.
	. . .1		IOBRPSTN	″X'10'″– DEVICE IS TO BE REPOSITIONED.
	. . . . 1. . .		IOBCYCCK	″X'08'″— CYCLIC REDUNDANCY CHECK (CRC) NEEDED (TAPE)
	. . . . 1. . .		IOBFCREX	″X'08'″– FETCH COMMAND RETRY EXIT (DIRECT ACCESS)
	. . . . .1. .		IOBIOERR	″X'04'″– EXCEPTIONAL CONDITION. AFTER THE ERROR ROUTINE RETURNS AND THIS BIT IS ON, THE ERROR IS CONSIDERED PERMANENT.
	. . . . . .1.		IOBUNREL	″X'02'″– IOB UNRELATED FLAG (THAT IS, NONSEQUENTIAL)
	1		IOBRSTRT	″X'01'″– IF 1, RESTART ADDRESS IN IOB TO BE USED. IF 0, START. (OS/VS1) MDC034
	1		IOBSPSVC	″X'01'″– FOR SAM/PAM, SET BY SVC IF I/O APPENDAGE SHOULD NOT PROCESS INTERRUPT (OS/VS2) MDC035
1 (1)	BITSTRING	1	IOBFLAG2	– FLAG BYTE 2
	1.		IOBHALT	″X'80'″– HALT I/O HAS BEEN ISSUED BY SVC PURGE ROUTINE
	.1.		IOBSENSE	″X'40'″– SENSE WILL NOT BE PERFORMED UNTIL THE DEVICE IS FREE
	. .1.		IOBPURGE	″X'20'″–IOB HAS BEEN PURGED TO ALLOW I/O ACTIVITY TO QUIESCE. (OS/VS1) MDC046
	. .1.		IOBRRT3	″X'20'″– TYPE 3 RELATED REQUEST (OS/VS2) MDC048
	. . .1		IOBRDHA0	″X'10'″– HOME ADDRESS (R0) RECORD IS TO BE READ. SEEK COMMAND NOT NEEDED. (OS/VS1) MDC047
	. . .1		IOBRRT2	″X'10'″– TYPE 2 RELATED REQUEST (OS/VS2) MDC049
	. . . . 1. . .		IOBALTTR	″X'08'″– NO TEST FOR OUT–OF–EXTENT. AN ALTERNATE TRACK IS IN USE.
	. . . . .1. .		IOBSKUPD	″X'04'″– SEEK ADDRESS IS BEING UPDATED. CYLINDER END OR FILE MASK VIOLATION HAS OCCURRED.
	. . . . . .1.		IOBSTATO	″X'02'″– DEVICE END STATUS HAS BEEN OR'ED WITH CHANNEL END STATUS (GRAPHICS DEVICE)
	1		IOBPNCH	″X'01'″– ERROR RECOVERY IN CONTROL FOR A 2540 CARD PUNCH WITH THREE BUFFERS (QSAM) RESETPL MACRO INSTRUCTION WAS USED (BTAM)
2 (2)	BITSTRING	1	IOBSENS0	– FIRST SENSE BYTE
	1.		IOBS0B0	″X'80'″– BIT 0 (DEVICE DEPENDENT)
	.1.		IOBS0B1	″X'40'″– BIT 1 (DEVICE DEPENDENT)
	. .1.		IOBS0B2	″X'20'″– BIT 2 (DEVICE DEPENDENT)
	. . .1		IOBS0B3	″X'10'″– BIT 3 (DEVICE DEPENDENT)
	. . . . 1. . .		IOBS0B4	″X'08'″– BIT 4 (DEVICE DEPENDENT)
	. . . . .1. .		IOBS0B5	″X'04'″– BIT 5 (DEVICE DEPENDENT)
	. . . . . .1.		IOBS0B6	″X'02'″– BIT 6 (DEVICE DEPENDENT)
	1		IOBS0B7	″X'01'″– BIT 7 (DEVICE DEPENDENT)
	1		IOBSNSC9	″X'01'″– CHANNEL 9 SENSED IN CARRIAGE TAPE

Offset	Type/Value	Len	Name (Dim)	Description
3 (3)	BITSTRING	1	IOBSENS1	– SECOND SENSE BYTE
	1.		IOBS1B0	″X'80'″– BIT 0 (DEVICE DEPENDENT)
	1.		IOBS1B1	″X'40'″– BIT 1 (DEVICE DEPENDENT)
	. .1.		IOBS1B2	″X'20'″– BIT 2 (DEVICE DEPENDENT)
	. . .1		IOBS1B3	″X'10'″– BIT 3 (DEVICE DEPENDENT)
1	. . . . 1. . .		IOBS1B4	″X'08'″– BIT 4 (DEVICE DEPENDENT)
	. . . . .1. .		IOBS1B5	″X'04'″– BIT 5 (DEVICE DEPENDENT)
	. . . . . .1.		IOBS1B6	″X'02'″– BIT 6 (DEVICE DEPENDENT)
	1		IOBS1B7	″X'01'″– BIT 7 (DEVICE DEPENDENT)
4 (4)	ADDRESS	4	IOBECBPT (0)	– ADDRESS OF ECB TO BE POSTED ON I/O COMPLETION
4 (4)	CHARACTER	1	IOBECBCC	– COMPLETION CODE FOR AN I/O REQUEST. THIS CODE WILL APPEAR IN THE FIRST BYTE OF AN ECB.
5 (5)	ADDRESS	3	IOBECBPB	–ADDRESS OF THE ECB TO BE POSTED UPON THE COMPLETION OF AN I/O EVENT. FOR BSAM/BPAM, ECB IS IN THE DECB. FOR QSAM, ECB IS IN THE QSAM PREFIX OF THE IOB.
8 (8)	BITSTRING	1	IOBFLAG3 (0)	– I/O SUPERVISOR ERROR ROUTINE FLAG BYTE (DEVICE DEPENDENT)
8 (8)	BITSTRING	1	IOBFL3	– FLAG 3 – STATUS ERROR COUNTS FOR MAGNETIC DOCUMENT READER (3890) OR FLAGS FOR 3800 (OS/VS1) (MDC307)
	1.		IOBCCC	″X'80'″– CHANNEL CONTROL CHECK ERROR COUNT (3890) MDC038
	.1.		IOBICC	″X'40'″– INTERFACE CONTROL CHECK ERROR COUNT (3890) MDC039
	. .1.		IOBCDC	″X'20'″– CHANNEL DATA CHECK ERROR (3890) MDC040
	. . .1		IOBACU	″X'10'″– ATTENTION/CONTROL UNIT ERROR (3890) MDC041
	. . . . 1. . .		IOBCNC	″X'08'″– CHAIN CHECK ERROR (3890) MDC042
	. . . . 1. . .		IOBSDR	″X'08'″– STATISTICS ONLY FLAG (3800) (MDC306)
	. . . . .1. .		IOBMSG	″X'04'″– MESSAGE FLAG (3890 OR 3800) (MDC308)
	. . . . . .1.		IOBICL	″X'02'″– INCORRECT LENGTH ERROR (3890) MDC020
	. . . . . .1.		IOBJAM	″X'02'″– SET ON WHEN JES SUBSYSTEM HAS DETECTED A PAPER JAM SO 3800 ERP WILL SUPPRESS ITS INTERVENTION REQUIRED MESSAGE (3800) (MDC330)
	1		IOBLOG	″X'01'″– LOG OUT FLAG (3890 OR 3800) (MDC309)
9 (9)	CHARACTER	7	IOBCSW (0)	– LOW–ORDER SEVEN BYTES OF THE LAST CSW THAT REFLECTS THE STATUS FOR THIS REQUEST
9 (9)	CHARACTER	5	IOBIOCSW (0)	– LOW–ORDER BYTES OF CSW FOR MAGNETIC DOCUMENT READER (3890) (OS/VS1) MDC022
9 (9)	ADDRESS	3	IOBCMDA	– COMMAND ADDRESS (3890) MDC023
12 (C)	BITSTRING	2	IOBSTBYT (0)	– STATUS BITS 32–47 (3890) MDC024
12 (C)	BITSTRING	1	IOBUSTAT	– CSW UNIT STATUS FLAGS (3800) (MDC311)
	1.		IOBUSB0	″X'80'″– ATTENTION (MDC312)
	.1.		IOBUSB1	″X'40'″– STATUS MODIFIER (MDC313)
	. .1.		IOBUSB2	″X'20'″– CONTROL UNIT END (MDC314)
	. . .1		IOBUSB3	″X'10'″– BUSY (MDC315)
	. . . . 1. . .		IOBUSB4	″X'08'″– CHANNEL END (MDC316)
	. . . . .1. .		IOBUSB5	″X'04'″– DEVICE END (MDC317)

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . . .1.		IOBUSB6	″X'02'″– UNIT CHECK (MDC318)
	1		IOBUSB7	″X'01'″– UNIT EXCEPTION (MDC319)
13 (D)	BITSTRING	1	IOBCSTAT	– CSW CHANNEL STATUS FLAGS (3800) (MDC320)
	1.		IOBCSB0	″X'80'″– PROGRAM CONTROL INTERRUPT (MDC321)
	.1.		IOBCSB1	″X'40'″– INCORRECT LENGTH (MDC322)
	. .1.		IOBCSB2	″X'20'″– PROGRAM CHECK (MDC323)
	. . .1		IOBCSB3	″X'10'″– PROTECTION CHECK (MDC324)
	. . . . 1. . .		IOBCSB4	″X'08'″– CHANNEL DATA CHECK (MDC325)
	. . . . .1. .		IOBCSB5	″X'04'″– CHANNEL CONTROL CHECK (MDC326)
	. . . . . .1.		IOBCSB6	″X'02'″– INTERFACE CONTROL CHECK (MDC327)
	1		IOBCSB7	″X'01'″– CHAINING CHECK (MDC328)
14 (E)	BITSTRING	2		– LAST TWO BYTES OF IOBCSW
16 (10)	ADDRESS	4	IOBSTART (0)	– ADDRESS OF CHANNEL PROGRAM TO BE EXECUTED
16 (10)	BITSTRING	1	IOBSIOCC	– SIO CODE. BITS 2 AND 3 CONTAIN CONDITION CODE RETURNED AFTER EXECUTION OF SIO INSTRUCTION FOR THIS I/O EVENT.
17 (11)	ADDRESS	3	IOBSTRTB	– ADDRESS OF CHANNEL PROGRAM TO BE EXECUTED
20 (14)	ADDRESS	4	IOBDCBPT (0)	– ADDRESS OF DCB ASSOCIATED WITH THIS IOB
20 (14)	BITSTRING	1	IOBFLAG4	– FLAG BYTE MDC011
	1.		IOBGDPOL	″X'80'″– RE–ENTER SIO APPENDAGE FOR OLTEP GUARANTEED DEVICE PATH MDC012
	.1.		IOBCC3WE	″X'40'″– USER REQUESTS THAT IOS POST A X'6D' FOR A CONDITION CODE 3 ON ATTEMPTED I/O OPERATIONS (OS/VS2) (MDC310)
	. .1.		IOBPMERR	″X'20'″– VTAM SETS THIS BIT ON TO INDICATE TO IOS THAT VTAM SHOULD BE POSTED WITH A PERMANENT I/O ERROR BECAUSE ALL ALTERNATE PATHS TO THE 3705 HAVE BEEN TRIED (OS/VS1) (MDC329)
	. . .1		IOBCEF	″X'10'″– IOB COMMON EXTENSION IS AVAILABLE
	. . . . 1. . .		IOBRSV41	″X'08',,C’X’″– RESERVED
	. . . . .1. .		IOBRSV42	″X'04',,C’X’″– RESERVED
	. . . . . .1.		IOBJES3I	″X'02'″– JES3 INTERVENTION REQUIRED NOTIFICATION. SETTING THIS BIT WILL RESULT IN TURNING ON BIT IOSPGDPX IN THE IOSB. (OS/VS2)(MDC305)
	1		IOBRSV44	″X'01',,C’X’″– RESERVED
21 (15)	ADDRESS	3	IOBDCBPB	– ADDRESS OF DCB ASSOCIATED WITH THIS IOB
24 (18)	ADDRESS	4	IOBRESTR (0)	– AFTER SVC 16 (PURGE) – QUIESCE – ADDRESS OF THE NEXT IOB IN THE PURGE CHAIN. (LAST IOB IN THE CHAIN, BYTE 4 IS FF.) DURING I/O SUPERVISOR WRITE-TO-OPERATOR ROUTINE CONTROL - CCHH PART OF THE ADDRESS OF A DEFECTIVE TRACK. DURING I/O ERROR CORRECTION (MEANINGFUL ONLY IF BIT 3 IN THE IOBFLAG1 FIELD IS ON) – ADDRESS OF THE CHANNEL PROGRAM USED TO CORRECT AN ERROR CONDITION. AFTER I/O ERROR CORRECTION – IF A CHANNEL PROGRAM IS RESTARTED THROUGH A CCW OTHER THAN THE ONE POINTED TO BY THE IOBSTART FIELD, ITS ADDRESS IS HERE.

Offset	Type/Value	Len	Name (Dim)	Description
24 (18)	CHARACTER	1	IOBREPOS	– DURING I/O ERROR CORRECTION (MEANINGFUL ONLY IF BIT 3 IN THE IOBFLAG1 FIELD IS ON) FOR MAGNETIC TAPE ONLY – THE CONTROL COMMAND (BSR, FSR, ERG) REQUIRED TO REPOSITION OVER A BLOCK.
25 (19)	ADDRESS	3	IOBRSTRB	– SAME AS IOBRESTR ABOVE
28 (1C)	SIGNED	2	IOBINCAM (0)	– QSAM, BSAM, EXCP ACCESS METHOD – NORMAL SCHEDULING – VALUE USED TO INCREMENT BLOCK COUNT FIELD IN DCB FOR MAGNETIC TAPE. CHAINED SCHEDULING – ZEROS. QSAM, BSAM – OPERATION CODE OF WRITE CCW WHEN A USASI CONTROL CHARACTER AND NO DATA IS TO BE WRITTEN (PRINTER AND CARD PUNCH ONLY)
28 (1C)	BITSTRING	1	IOBBTAMF	– FLAG BYTE FOR BTAM
	1.		IOBPRMER	″X'80'″– SAD OR ENABLE ISSUED BY OPEN RESULTED IN A PERMANENT I/O ERROR
	.1.		IOBINUSE	″X'40'″– THIS IOB IS CURRENTLY IN USE BY AN I/O OPERATION
	. .1.		IOBRSV14	″X'20',,C'X'″RESERVED
	. . .1		IOBRSV15	″X'10',,C'X'″RESERVED
	. . . . 1. . .		IOBRSV16	″X'08',,C'X'″RESERVED
	. . . . .1. .		IOBRSV17	″X'04',,C'X'″RESERVED
	. . . . . .1.		IOBRFTMG	″X'02'″″– A REQUEST–FOR–TEST MESSAGE RECEIVED FROM A REMOTE 3270 DISPLAY STATION ICB478
	1		IOBOLTST	″X'01'″– LINE IS UNDER ON– LINE TEST OPERATION
29 (1D)	BITSTRING	1	IOBRSV19	–RESERVED
28 (1C)	BITSTRING	1	IOBFL4	– FLAG 4 – SENSE ERROR COUNTS FOR MAGNETIC DOCUMENT READER (3890) (OS/VS1) OR ERROR CODE PASSBACK BYTE FOR 3895 (FOR ERROR CODE VALUES, SEE IBM 3895 DOCUMENT READER/INSCRIBER MACHINE AND PROGRAMMING DESCRIPTION, GA24–3620) (MDC333)
	1.		IOBOVR	″X'80'″– OVERRUN ERROR (3890) (MDC026)
	.1.		IOBREJ	″X'40'″– COMMAND REJECT ERROR (3890) (MDC027)
	. .1.		IOBDCK	″X'20'″– DATA CHECK ERROR (3890) (MDC028)
	. . .1		IOBBUS	″X'10'″– BUS–OUT ERROR (3890) (MDC029)
	. . . . 1. . .		IOBEQP	″X'08'″– EQUIPMENT CHECK ERROR (3890) (MDC030)
	. . . . .1. .		IOBENT	″X'04'″– FIRST TIME ENTRY SWITCH (3890) (MDC045)
	. . . . . .1.		IOBRSV47	″X'02',,C'X'″– RESERVED FOR 3890 MDC044
	1		IOBRSV46	″X'01',,C'X'″– RESERVED FOR 3890 MDC033
28 (1C)	CHARACTER	1	IOBCRDCC	– DATA CHECK ERROR COUNT (OPTICAL READER)
29 (1D)	CHARACTER	1	IOBCRILC	– INCORRECT LENGTH ERROR COUNT (OPTICAL READER)
28 (1C)	BITSTRING	1	IOBAMAF	PDSE DIAGNOSTICS BYTE TO DESCRIBE ERRORS DETECTED BY AMA PDSE ROUTINES
	1		IOBBDFPL	″X'01'″– BAD FPL PASSED TO ASYNC COMP RTN (IGG019SY) BY JCDM

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . . .1.		IOBPGMCK	″X'02'″– PROGRAM CHECK ENCOUNTERED IN IGG019SY (IGGT19SY ENTERED).
	11		IOBBADRC	″X'03'″– BAD RETURN CODE ENCOUNTERED FROM SMS –IOBDIAG RETURN/REASON CODES UPDATED
	. . . . .1. .		IOBBADLT	″X'04'″– INVALID LOCATOR TOKEN DETECTED BY POINT
	. . . . .1.1		IOBNCRLT	″X'05'″– RLT WAS INPUT BUT NO MEMBER WAS CONNECTED (DETECTED BY POINT)
	. . . . .11.		IOBPOUT	″X'06'″– DCB OPEN FOR OUTPUT & AN MLT FOR OTHER THAN THE CURRENT MEMBER WAS INPUT (DETECTED BY POINT)
	111		IOBOUTIO	″X'07'″– OUTSTANDING I/O IN PROGRESS DETECTED BY POINT
	. . . . 1. . .		IOBRLTTB	″X'08'″– RLT TOO BIG (>= HWM) DETECTED BY POINT
	. . . . 1. .1		IOBFLOCK	″X'09'″– CONNECT OR RECONNECT UNABLE TO GET FILE LOCK. DETECTED BY POINT. COULD BE A USER ERROR.
	. . . . 1.1.		IOBNFILE	″X'0A'″– MLT INPUT TO POINT BUT NO SUCH FILE
30 (1E)	SIGNED	2	IOBERRCT	– USED BY I/O SUPERVISOR ERROR ROUTINES TO COUNT TEMPORARY ERRORS DURING RETRY
	EXTENSION SECTIONS OF THE IOB			
32 (20)	DBL WORD	8	IOBEXTEN(0)	
	DIRECT ACCESS EXTENSION 8 BYTES			
32 (20)	CHARACTER	8	IOBSEEK (0)	– A SEEK ADDRESS (IN THE FORMAT MBBCCHHR) USED WITH A CHANNEL PROGRAM
32 (20)	CHARACTER	1	IOBM	– THE NUMBER OF THE DEB EXTENT TO BE USED FOR THIS REQUEST. THE FIRST EXTENT IS NUMBER 0.
33 (21)	CHARACTER	2	IOBBB (0)	– BIN NUMBER(DATA CELL)
33 (21)	CHARACTER	1	IOBBB1	BIN NUMBER
34 (22)	CHARACTER	1	IOBBB2	BIN NUMBER
35 (23)	CHARACTER	2	IOBCC (0)	– CYLINDER NUMBER
35 (23)	CHARACTER	1	IOBCC1	– CYLINDER NUMBER
36 (24)	CHARACTER	1	IOBCC2	– CYLINDER NUMBER
37 (25)	CHARACTER	2	IOBHH (0)	– TRACK NUMBER
37 (25)	CHARACTER	1	IOBHH1	– TRACK NUMBER
38 (26)	CHARACTER	1	IOBHH2	– TRACK NUMBER
39 (27)	CHARACTER	1	IOBR	– RECORD NUMBER
	BISAM EXTENSION 16 BYTES			
40 (28)	ADDRESS	4	IOBCCWAD	– FOR FIXED LENGTH RECORDS, ADDRESS OF FIRST CCW OF CHANNEL PROGRAM. FOR VARIABLE LENGTH RECORDS, ADDRESS OF BUFFER, IF DYNAMIC BUFFERING SPECIFIED, AFTER COMPLETION OF A READ FOR UPDATE (READ KU)
44 (2C)	BITSTRING	1	IOBINDCT	– INDICATORS
	1.		IOBDEQCP	″X'80'″– DEQUEUE CHANNEL PROGRAM FROM QUEUE
	.1.		IOBUNSCH	″X'40'″– UNSCHEDULED QUEUE
	. .1.		IOBOVPTR	″X'20'″– IF 0, DECBAREA + 6 POINTS TO OVERFLOW RECORD DATA. IF 1, DCBMSWA POINTS TO OVERFLOW RECORD KEY FOLLOWED BY DATA.

Offset	Type/Value	Len	Name (Dim)	Description
1	. . .1		IOBKEYAD	″X'10'″– IF 0, DECBKEY POINTS TO OVERFLOW RECORD KEY. IF 1, DCBMSWA + 8 POINTS TO OVERFLOW RECORD KEY.
	. . . . 1. . .		IOBRSV27	″X'08',,C'X'″RESERVED
	. . . . .1. .		IOBRSV28	″X'04',,C'X'″RESERVED
	. . . . . .1.		IOBRSV29	″X'02',,C'X'″RESERVED
	1		IOBCHNNL	″X'01'″– IF 0, NORMAL CHANNEL END HAS OCCURRED. IF 1, ABNORMAL CHANNEL END HAS OCCURRED.
45 (2D)	BITSTRING	1	IOBUNSQR	– REASON FOR UNSCHEDULED QUEUE
	1.		IOBCPBSY	″X'80'″– CHANNEL PROGRAM CP1 OR CP2 BUSY
	.1.		IOBNTAV1	″X'40'″– NO CP4, CP5 OR CP6 AVAILABLE
	. .1.		IOBNTAV2	″X'20'″– NO CP7 AVAILABLE
1	. . .1		IOBKNWR	″X'10'″– WRITE KN IS IN EFFECT (UNSCHEDULED IOB IS FOR WRITE KN)
	. . . . 1. . .		IOBKNRWR	″X'08'″– WRITE KN IS IN EFFECT (UNSCHEDULED IOB IS FOR READ OR WRITE KN)
	. . . . .1. .		IOBRSV30	″X'04',,C'X'″RESERVED
	. . . . . .1.		IOBRSV31	″X'02',,C'X'″RESERVED
	1		IOBRSV32	″X'01',,C'X'″RESERVED
46 (2E)	CHARACTER	1	IOBAPP	– APPENDAGE CODE
47 (2F)	CHARACTER	1	IOBASYN	– ASYNCHRONOUS ROUTINE CODE
48 (30)	ADDRESS	4	IOBFCHAD (0)	– FORWARD CHAIN ADDRESS
48 (30)	CHARACTER	1	IOBCOUNT	– WRITE CHECK COUNTER
49 (31)	ADDRESS	3	IOBFCHNB	– FORWARD CHAIN ADDRESS
52 (34)	ADDRESS	4	IOBBCHAD	– BACKWARD CHAIN ADDRESS
GAM EXTENSION 40 BYTES				
32 (20)	CHARACTER	1	IOBUCBXG	– UCB INDEX
33 (21)	BITSTRING	3	IOBRSV37	– RESERVED
36 (24)	ADDRESS	4	IOBNXTPT (0)	– ADDRESS OF NEXT AVAILABLE IOB. SET TO ZERO IF THIS IS LAST IOB.
36 (24)	BITSTRING	1	IOBSTATA	– STATUS INDICATORS
	1.		IOBAVLFL	″X'80'″– IF 0, IOB IS AVAILABLE. IF 1, IOB IS NOT AVAILABLE.
	.1.		IOBRSV20	″X'40',,C'X'″RESERVED
	. .1.		IOBRSV21	″X'20',,C'X'″RESERVED
	. . .1		IOBRSV22	″X'10',,C'X'″RESERVED
	. . . . 1. . .		IOBRSV23	″X'08',,C'X'″RESERVED
	. . . . .1. .		IOBRSV24	″X'04',,C'X'″RESERVED
1	. . . . . .1.		IOBRSV25	″X'02',,C'X'″RESERVED
	1		IOBRSV26	″X'01',,C'X'″RESERVED
37 (25)	ADDRESS	3	IOBNXTPB	– ADDRESS OF NEXT AVAILABLE IOB. SET TO ZERO IF THIS IS LAST IOB
40 (28)	CHARACTER	32	IOBCCW	– LIST OF CHANNEL COMMAND WORDS TO TRANSFER DATA
QISAM EXTENSION 2 BYTES				
40 (28)	CHARACTER	2	W1IEXTEN (0)	– APPENDAGE CODES FOR BOTH NORMAL AND ABNORMAL CHANNEL END CONDITIONS
40 (28)	CHARACTER	2	W1OEXTEN	– SAME AS W1IEXTEN ABOVE
BDAM EXTENSION 40 BYTES + CHANNEL PROGRAM				
40 (28)	SIGNED	2	IOBDBYTR	– NUMBER OF UNUSED BYTES REMAINING ON THE TRACK
42 (2A)	SIGNED	2	IOBDIOBS	– OVERALL SIZE OF THE IOB

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Offset	Type/Value	Len	Name (Dim)	Description
44 (2C)	ADDRESS	4	IOBDPLAD (0)	– ADDRESS OF THE NEXT IOB IN THE POOL OF IOB'S
44 (2C)	BITSTRING	1	IOBDAYLI	– ALL BITS SET TO ZERO INDICATE THE AVAILABILITY OF THIS IOB
45 (2D)	ADDRESS	3	IOBDPLB	– ADDRESS OF THE NEXT IOB IN THE POOL OF IOB'S
48 (30)	BITSTRING	1	IOBDTYPE	– THE TYPE OF REQUEST AND SPECIFIED OPTIONS
	1.		IOBVERFY	″X'80'″– VERIFY
	.1.		IOBOVFLO	″X'40'″– OVERFLOW
	. .1.		IOBEXTSC	″X'20'″– EXTENDED SEARCH
	. . .1		IOBFDBCK	″X'10'″– FEEDBACK
	. . . . 1. . .		IOBACTAD	″X'08'″– ACTUAL ADDRESSING
	. . . . .1. .		IOBDYNBF	″X'04'DYNAMIC BUFFERING
	. . . . . .1.		IOBRDEXC	″X'02'″– READ EXCLUSIVE
	1		IOBRELBL	″X'01'″– RELATIVE BLOCK ADDRESSING
49 (31)	BITSTRING	1	IOBDTYP2	– SECOND BYTE OF OPTIONS AND REQUESTS
	1.		IOBSKEY	″X'80'″– KEY ADDRESS CODED AS 'S'
	.1.		IOBSBLKL	″X'40'″– BLOCK LENGTH CODED AS 'S'
	. .11		IOBSUFFX	″X'30'″– IF BITS 2 AND 3 ARE ONE, RU IS SUFFIXED TO THE TYPE, INDICATING THAT THE FEEDBACK ADDRESS IN DECNXADR CAN BE THE ADDRESS OF EITHER THE NEXT DATA RECORD OR THE NEXT CAPACITY RECORD, WHICHEVER OCCURS FIRST. IF BIT 2 IS ZERO AND BIT 3 IS ONE, R IS SUFFIXED TO THE TYPE, INDICATING THAT THE FEEDBACK ADDRESS IN DECNXADR IS THE ADDRESS OF THE NEXT DATA RECORD.
	. . . . 1. . .		IOBRQUST	″X'08'″– IF 1, READ REQUEST. IF 0, WRITE REQUEST.
	. . . . .1. .		IOBTYPE	″X'04'″– IF 1, KEY TYPE. IF 0, ID TYPE.
	. . . . . .1.		IOBADDTY	″X'02'″– ADD TYPE
	1		IOBRELEX	″X'01'″– RELEX MACRO ISSUED
50 (32)	CHARACTER	2	IOBDSTAT (0)	– STATUS OF THE I/O REQUEST
50 (32)	BITSTRING	1	IOBSTAT1	– FLAG BYTE
1.	1.		IOBABNRM	″X'80'″– ABNORMAL COMPLETION
	.1.		IOBNEWVL	″X'40'″– ON EXTENDED SEARCH, THE NEXT EXTENT IS ON A NEW VOLUME. THE ASI ROUTINE MUST ISSUE THE EXCP MACRO. THE END–OF–EXTENT APPENDAGE CANNOT.
	. .1.		IOBSYNCH	″X'20'″– MODULE WAS ENTERED VIA SYNCH MDC037
	. . .1		IOBPASS2	″X'10'″– ON EXTENDED SEARCH, INDICATES TO THE RELATIVE BLOCK CONVERSION ROUTINE THAT THE SECOND PASS OF A TWO-PASS CONVERSION ROUTINE HAS COMPLETED
	. . . . 1. . .		IOBENQUE	″X'08E'″– FOR EXCLUSIVE CONTROL REQUEST, INDICATES THAT A RECORD HAS BEEN ENQUEUED
	. . . . .1. .		IOBBUFF	″X'04'″– A BUFFER HAS BEEN ASSIGNED TO THIS IOB
	. . . . . .1.		IOBADDVU	″X'02'″– IOB BEING USED TO ADD A VARIABLE (V) OR UNDEFINED (U) TYPE RECORD TO THE DATA SET

Offset	Type/Value	Len	Name (Dim)	Description
	1		IOBSIORT	″X'01'″– INDICATES TO THE DYNAMIC BUFFERING ROUTINE THAT IT WAS ENTERED FROM, AND IS TO RETURN TO, THE START I/O APPENDAGE MODULE
51 (33)	CHARACTER	1	IOBSTAT2	– ERROR CODE FOR ABNORMAL COMPLETION USED AS POST CODE IN ECB
52 (34)	ADDRESS	4	IOBDCPND	– ADDRESS OF LOCATION WHERE CHANNEL END PROGRAM SHOULD END
56 (38)	SIGNED	2	IOBDBYTN	– NUMBER OF BYTES NEEDED ON A TRACK TO WRITE A NEW BLOCK
58 (3A)	BITSTRING	1	IOBREQ	– REQUEST FLAGS
	1.		IOBREC31	″X'80'″– BLOCK REFERENCE ADDR (DECRECPT) IS A 31 BIT ADDR
	.1.		IOBKEY31	″X'40'″– KEY ADDR (DECKYADR) IS A 31 BIT ADDR
	. .1.		IOBDAT31	″X'20'″– DATA ADDR (DECAREA) IS A 31 BIT ADDR
59 (3B)	BITSTRING	1	IOBRSV34	– RESERVED
60 (3C)	ADDRESS	4	IOBDQPTR	– ADDRESS OF IOB FOR NEXT I/O OPERATION TO BE EXECUTED
64 (40)	BITSTRING	8	IOBRSV35	– RESERVED
72 (48)	CHARACTER	8	IOBDNCRF	– COUNT FIELD FOR NEW BLOCK
80 (50)	DBL WORD	8	IOBCHNPR (0)	– CHANNEL PROGRAM USED TO TRANSFER DATA AS REQUESTED BY THE READ OR WRITE MACRO INSTRUCTION STARTS HERE

Cross-Reference

Name	Hex Offset	Hex Value	Level
IOBABAPP	-10	4	2
IOBABNRM	32	80	2
IOBACTAD	30	8	2
IOBACU	8	10	2
IOBADDTY	31	2	2
IOBADDVU	32	2	2
IOBALTTR	1	8	2
IOBAMAF	1C		2
IOBAPP	2E		2
IOBASYN	2F		2
IOBAVLFL	24	80	2
IOBBADLT	1C	4	2
IOBBADRC	1C	3	2
IOBBB	21		2
IOBBB1	21		2
IOBBB2	22		2
IOBBCHAD	34		2
IOBBDAM	-8		2
IOBBDFPL	1C	1	2
IOBBKSPC	-8	8	2
IOBBPAMC	-10		2
IOBBPAMN	-8		2
IOBBSAMC	-10		2
IOBBSAMN	-8		2
IOBBTAMF	1C		2
IOBBUFF	32	4	2
IOBBUS	1C	10	2

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Name	Hex Offset	Hex Value	Level
IOBCC	23		2
IOBCCC	8	80	2
IOBCCW	28		2
IOBCCWAD	28		2
IOBCC1	23		2
IOBCC2	24		2
IOBCC3WE	14	40	2
IOBCDC	8	20	2
IOBCECB	-C		2
IOBCEF	14	10	2
IOBCFLG1	-10		2
IOBCHNNL	2C	1	2
IOBCHNPR	50		2
IOBCICB	-8		2
IOBCINOP	-E		2
IOBCMDA	9		2
IOBCMDCH	0	40	2
IOBCNC	8	8	2
IOBCNOPA	-4		2
IOBCONOP	-D		2
IOBCOUNT	30		2
IOBCPBSY	2D	80	2
IOBCRDCC	1C		2
IOBCRILC	1D		2
IOBCSB0	D	80	2
IOBCSB1	D	40	2
IOBCSB2	D	20	2
IOBCSB3	D	10	2
IOBCSB4	D	8	2
IOBCSB5	D	4	2
IOBCSB6	D	2	2
IOBCSB7	D	1	2
IOBCSTAT	D		2
IOBCSW	9		2
IOBCYCCK	0	8	2
IOBDATCH	0	80	2
IOBDAT31	3A	20	2
IOBDAYLI	2C		2
IOBDBYTN	38		2
IOBDBYTR	28		2
IOBDCBPB	15		2
IOBDCBPT	14		2
IOBDCK	1C	20	2
IOBDCPND	34		2
IOBDEQ	-8	80	2
IOBDEQCP	2C	80	2
IOBDEQIN	-8		2
IOBDIOBS	2A		2
IOBDNCRF	48		2
IOBDPLAD	2C		2
IOBDPLB	2D		2
IOBDQADA	-8		2
IOBDQADB	-7		2
IOBDQPTR	3C		2

Name	Hex Offset	Hex Value	Level
IOBDSTAT	32		2
IOBDTYPE	30		2
IOBDTYP2	31		2
IOBDYNBF	30	4	2
IOBECBCC	4		2
IOBECBPB	5		2
IOBECBPT	4		2
IOBENQUE	32	8	2
IOBENT	1C	4	2
IOBEQP	1C	8	2
IOBERRCT	1E		2
IOBERRTN	0	20	2
IOBEXTEN	20		2
IOBEXTSC	30	20	2
IOBFCHAD	30		2
IOBFCHNB	31		2
IOBFCREX	0	8	2
IOBFDBCK	30	10	2
IOBFIRST	-8	1	2
IOBFLAG1	0		2
IOBFLAG2	1		2
IOBFLAG3	8		2
IOBFLAG4	14		2
IOBFLOCK	1C	9	2
IOBFL3	8		2
IOBFL4	1C		2
IOBGAM	0	C	2
IOBGDPOL	14	80	2
IOBGQECB	-4		2
IOBHALT	1	80	2
IOBHH	25		2
IOBHH1	25		2
IOBHH2	26		2
IOBICC	8	40	2
IOBICL	8	2	2
IOBINCAM	1C		2
IOBINDCT	2C		2
IOBINUSE	1C	40	2
IOBIOCSW	9		2
IOBIOERR	0	4	2
IOBJAM	8	2	2
IOBJES3I	14	2	2
IOBKEYAD	2C	10	2
IOBKEY31	3A	40	2
IOBKNRWR	2D	8	2
IOBKNWR	2D	10	2
IOBLOG	8	1	2
IOBM	20		2
IOBMSG	8	4	2
IOBNCRLT	1C	5	2
IOBNECB	-4		2
IOBNEWVL	32	40	2
IOBNFILE	1C	A	2
IOBNFLG1	-8		2

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Name	Hex Offset	Hex Value	Level
IOBNIOBA	-8		2
IOBNIOBB	-7		2
IOBNTAV1	2D	40	2
IOBNTAV2	2D	20	2
IOBNXTPB	25		2
IOBNXTPT	24		2
IOBOLTST	1C	1	2
IOBOUTIO	1C	7	2
IOBOVFLO	30	40	2
IOBOVPTR	2C	20	2
IOBOVR	1C	80	2
IOBPASS2	32	10	2
IOBPCI	-10	1	2
IOBPGMCK	1C	2	2
IOBPMERR	14	20	2
IOBPNCH	1	1	2
IOBPOUT	1C	6	2
IOBPREFX	-10		2
IOBPRMER	1C	80	2
IOBPRTOV	-8	80	2
IOBPTST	-10	8	2
IOBPURGE	1	20	2
IOBQISAM	0	C	2
IOBQSAMC	-10		2
IOBQSAMN	-8		2
IOBR	27		2
IOBRDEXC	30	2	2
IOBRDHA0	1	10	2
IOBREAD	-8	20	2
IOBREC31	3A	80	2
IOBREJ	1C	40	2
IOBRELBL	30	1	2
IOBRELEX	31	1	2
IOBREPOS	18		2
IOBREQ	3A		2
IOBRESTR	18		2
IOBRFTMG	1C	2	2
IOBRLTTB	1C	8	2
IOBRPSTN	0	10	2
IOBRQUST	31	8	2
IOBRRT2	1	10	2
IOBRRT3	1	20	2
IOBRSTCH	-10	2	2
IOBRSTRB	19		2
IOBRSTRT	0	1	2
IOBRSV02	-10	40	2
IOBRSV03	-10	20	2
IOBRSV04	-10	10	2
IOBRSV05	-F		2
IOBRSV07	-8	40	2
IOBRSV08	-8	20	2
IOBRSV09	-8	10	2
IOBRSV10	-8	8	2
IOBRSV11	-8	4	2

Name	Hex Offset	Hex Value	Level
IOBRSV12	-8	2	2
IOBRSV13	-8	1	2
IOBRSV14	1C	20	2
IOBRSV15	1C	10	2
IOBRSV16	1C	8	2
IOBRSV17	1C	4	2
IOBRSV19	1D		2
IOBRSV20	24	40	2
IOBRSV21	24	20	2
IOBRSV22	24	10	2
IOBRSV23	24	8	2
IOBRSV24	24	4	2
IOBRSV25	24	2	2
IOBRSV26	24	1	2
IOBRSV27	2C	8	2
IOBRSV28	2C	4	2
IOBRSV29	2C	2	2
IOBRSV30	2D	4	2
IOBRSV31	2D	2	2
IOBRSV32	2D	1	2
IOBRSV34	3B		2
IOBRSV35	40		2
IOBRSV37	21		2
IOBRSV41	14	8	2
IOBRSV42	14	4	2
IOBRSV44	14	1	2
IOBRSV46	1C	1	2
IOBRSV47	1C	2	2
IOBRTYPE	20		2
IOBSBLKL	31	40	2
IOBSDR	8	8	2
IOBSEEK	20		2
IOBSEGMT	-8	80	2
IOBSENSE	1	40	2
IOBSENS0	2		2
IOBSENS1	3		2
IOBSIOCC	10		2
IOBSIORT	32	1	2
IOBSKEY	31	80	2
IOBSKUPD	1	4	2
IOBSNSC9	2	1	2
IOBSPAN	-8	4	2
IOBSPSVC	0	1	2
IOBSTART	10		2
IOBSTATA	24		2
IOBSTATO	1	2	2
IOBSTAT1	32		2
IOBSTAT2	33		2
IOBSTBYT	C		2
IOBSTDRD	0		2
IOBSTRTB	11		2
IOBSUFFX	31	30	2
IOBSWAP	-4		2
IOBSYNCH	32	20	2

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Name	Hex Offset	Hex Value	Level
IOBS0B0	2	80	2
IOBS0B1	2	40	2
IOBS0B2	2	20	2
IOBS0B3	2	10	2
IOBS0B4	2	8	2
IOBS0B5	2	4	2
IOBS0B6	2	2	2
IOBS0B7	2	1	2
IOBS1B0	3	80	2
IOBS1B1	3	40	2
IOBS1B2	3	20	2
IOBS1B3	3	10	2
IOBS1B4	3	8	2
IOBS1B5	3	4	2
IOBS1B6	3	2	2
IOBS1B7	3	1	2
IOBTYPE	31	4	2
IOBUCBXG	20		2
IOBUNREL	0	2	2
IOBUNSCH	2C	40	2
IOBUNSQR	2D		2
IOBUPDAT	-8	10	2
IOBUPERR	-8	2	2
IOBUSB0	C	80	2
IOBUSB1	C	40	2
IOBUSB2	C	20	2
IOBUSB3	C	10	2
IOBUSB4	C	8	2
IOBUSB5	C	4	2
IOBUSB6	C	2	2
IOBUSB7	C	1	2
IOBUSTAT	C		2
IOBVERFY	30	80	2
IOBV6CHN	-10	80	2
IOBWRITE	-8	40	2
W1IEXTEN	28		2
W1OEXTEN	28		2

DEB

Common Name: Data Extent Block (DEB)
Macro Name: IEZDEB
DSECT Name:

- DEB (DSECT card precedes AVT section)
- DEBBASIC should be used for USING basic section.
- DEBDASD (DSECT name for direct access section)
- DEBACSMD (DSECT name for access method sections)
- DEBSUBNM (DSECT name for subroutine name section)
- DEBXTN (DSECT name for DEB extension)

Owning Component: Data Management, subcomponent OPEN/CLOSE/EOV

Eye-Catcher ID: None

Subpool and Key: 230 and key 5

Size: Variable (device and access method dependent sections)

Created by: Access method OPEN executor

Pointed to by:

- DCBDEBAD field of the DCB data area
- RQEDEB field of the RQE data area
- DCBODEBA field of the DCB data area (old DEB prior to OPEN)
- DEBDEBAD field of the DEB data area (next DEB on the chain)
- SSDADEBP field of the SSOB data area (associated data management DEB)
- TCBDEBBP field of the TCB data area (first DEB on the chain)

Serialization: LOCAL lock serializes the placing of a DEB on the TCB DEB chain and in the DEB table. OPEN/CLOSE/EOV processing is serialized by local lock and DEBCHK.

Function: The Data Extent Block (DEB) is an extension of the information in the DCB. Each DEB is associated with a DCB, and the two point to each other. A DEB contains information about the physical characteristics of the data set and other information used by the control program.

Note: This section includes fields that are added to the DEB by various access methods. The general use fields that are intended as programming interfaces are described in *z/OS DFSMSdfp Advanced Services*.

Offset	Type/Value	Len	Name (Dim)	Description
APPENDAGE VECTOR TABLE SECTION OF THE DEB POINTED TO BY DEBAPPAD				
–36 (–24)	DBL WORD	8	DEBAVT (0)	– APPENDAGE VECTOR TABLE
–36 (–24)	ADDRESS	4	DEBEOEA (0)	–ADDRESS OF END–OF–EXTENT APPENDAGE ROUTINE ICB280
–36 (–24)	BITSTRING	1	DEBEOEAB	– FLAG BYTE ICB280
	X'0'		DEBTCAMQ	″DEBEOEA″–TCAM USES AS WORK AREA (BITS 0–3)
	 1111		DEBEOENP	″X'0F'″– NUMBER OF 2K PAGES TO BE FIXED FOR THE END–OF–EXTENT APPENDAGE ICB280
–35 (–23)	ADDRESS	3	DEBEOEAD	– ADDRESS OF END–OF–EXTENT APPENDAGE ROUTINE ICB280
–32 (–20)	ADDRESS	4	DEBSIOA (0)	– ADDRESS OF START I/O APPENDAGE ROUTINE ICB280
–32 (–20)	BITSTRING	1	DEBSIOAB	– FLAG BYTE ICB280
	1.		DEBPGFX	″X'80'″– ADDRESS IN DEBSIOAD CAN BE USED TO DETERMINE THE ENTRY POINT TO THE PAGE FIX (PGFX) APPENDAGE ROUTINE BY ADDING 4 TO THE ADDRESS IN DEBSIOAD ICB280

Offset	Type/Value	Len	Name (Dim)	Description
	.1.		DEBSIOX	″X'40'″– IF ZERO, DO NOT ENTER SIO APPENDAGE WHEN ERP IS ACTIVE. IF ONE, ENTER SIO APPENDAGE EVEN WHEN ERP IS ACTIVE. ICB372
	. .1.		DEBIOVR	″X'20'″– IF ONE, EXCPVR REQUEST IS VALID. IF ZERO, EXCPVR REQUEST IS INVALID AND WILL NOT BE EXECUTED. CURRENTLY HAS NO EFFECT. ICB401
	. . .1		DEBFIX	″X'10'″– INDICATION THAT DEB HAS BEEN FIXED (OS/VS2) (MDC032) YM5929
	 1111		DEBSIONP	″X'0F'″– NUMBER OF 2K PAGES TO BE FIXED FOR THE SIO APPENDAGE ICB280
–31 (–1F)	ADDRESS	3	DEBSIOAD	– ADDRESS OF START I/O APPENDAGE ROUTINE ICB280
–28 (–1C)	ADDRESS	4	DEBPCIA (0)	– ADDRESS OF PCI APPENDAGE ROUTINE ICB280
–28 (–1C)	BITSTRING	1	DEBPCIAB	– FLAG BYTE ICB280
	 1111		DEBPCINP	″X'0F'″– NUMBER OF 2K PAGES TO BE FIXED FOR THE PCI APPENDAGE ICB280
–27 (–1B)	ADDRESS	3	DEBPCIAD	– ADDRESS OF PROGRAM–CONTROLLED–INTERRUPTION (PCI) APPENDAGE ROUTINE ICB280
–24 (–18)	ADDRESS	4	DEBCEA (0)	– ADDRESS OF CHANNEL-END APPENDAGE ROUTINE ICB280
–24 (–18)	BITSTRING	1	DEBCEAB	– FLAG BYTE ICB280
	1.		DEBESMVR	″X'80',,C'X'″VALIDITY CHECK FOR EXCPVR CALLER
	 1111		DEBCENP	″X'0F'″– NUMBER OF 2K PAGES TO BE FIXED FOR THE CHANNEL–END APPENDAGE ICB280
–23 (–17)	ADDRESS	3	DEBCEAD	– ADDRESS OF CHANNEL–END APPENDAGE ROUTINE ICB280
–20 (–14)	ADDRESS	4	DEBXCEA (0)	– ADDRESS OF ABNORMAL–END APPENDAGE ROUTINE ICB280
–20 (–14)	BITSTRING	1	DEBXCEAB	– FLAG BYTE ICB280
	 1111		DEBXCENP	″X'0F'″– NUMBER OF 2K PAGES TO BE FIXED FOR THE ABNORMAL–END APPENDAGE ICB280
–19 (–13)	ADDRESS	3	DEBXCEAD	– ADDRESS OF ABNORMAL–END APPENDAGE ROUTINE ICB280
	DEB PREFIX TABLE			
–16 (–10)	ADDRESS	4	DEBPREFX (0)	– DEB PREFIX TABLE
–16 (–10)	BITSTRING	1	DEBWKARA	– O/C/E WORK AREA (DIRECT ACCESS)
–15 (–F)	BITSTRING	7	DEBDSCBA	– DSCB ADDRESS (BBCCHHR) USED BY I/O SUPPORT (DIRECT ACCESS)
–8 (–8)	ADDRESS	4	DEBXTNP (0)	– POINTER TO DEB EXTENSION (OS/VS2) MDC008
–8 (–8)	BITSTRING	4	DEBDCBMK	– DCB MODIFICATION MASK USED BY O/C/E. OBSOLETE MDC009
–4 (–4)	BITSTRING	1	DEBLNGTH	– LENGTH OF DEB IN DOUBLEWORDS
–3 (–3)	CHARACTER	1	DEBAMTYP	– ACCESS METHOD TYPE ICB380
	X'0'		DEBAMNON	″0″ ACCESS METHOD TYPE NOT KNOWN
	X'1'		DEBAMVSM	″1″ VSAM ACCESS METHOD TYPE
	X'2'		DEBAMXCP	″2″ EXCP ACCESS METHOD TYPE
	X'4'		DEBAMTCM	″4″ TCAM ACCESS METHOD TYPE
	X'8'		DEBAMGAM	″8″ GRAPHICS ACCESS METHOD TYPE
	X'10'		DEBAMTAM	″16″ BTAM ACCESS METHOD TYPE
	X'20'		DEBAMBPM	″32″ BPAM ACCESS METHOD TYPE
	X'20'		DEBAMSAM	″32″ SEQUENTIAL ACCESS METHOD TYPE

Offset	Type/Value	Len	Name (Dim)	Description
	X'40'		DEBAMBDM	″64″ DIRECT ACCESS METHOD TYPE
	X'80'		DEBAMISM	″128″ ISAM ACCESS METHOD TYPE
	X'81'		DEBAMSUB	″129″ SUBSYSTEM ACCESS METHOD TYPE
	X'82'		DEBAMVTM	″130″ VTAM ACCESS METHOD TYPE
	X'84'		DEBAMTAP	″132″ TCAM APPLICATION ACC METHOD TYPE
–2 (–2)	SIGNED	2	DEBTBLOF	– OFFSET IN THE DEB TABLE TO THE ENTRY FOR THIS DEB ICB380
	DEB BASIC SECTION			
0 (0)	ADDRESS	4	DEBTCBAD (0)	– ADDRESS OF TCB FOR THIS DEB
0 (0)	BITSTRING	1	DEBAMID (0)	– VTAM/TCAM DEB ID FLAG
	. . . . .		DEBTAMID	″X'00'″– TCAM DEB ID
	 1111		DEBVAMID	″X'0F'″– VTAM DEB ID
0 (0)	BITSTRING	1	DEBNMSUB	– OF SUBROUTINES LOADED BY OPEN EXECUTOR ROUTINES
1 (1)	ADDRESS	3	DEBTCBB	– ADDRESS OF TCB FOR THIS DEB
4 (4)	ADDRESS	4	DEBDEBAD (0)	– ADDRESS OF THE NEXT DEB IN THE SAME TASK
4 (4)	BITSTRING	1	DEBAMLNG	– NUMBER OF BYTES IN THE ACCESS METHOD DEPENDENT SECTION. FOR BDAM THIS FIELD CONTAINS THE LENGTH EXPRESSED IN NUMBER OF WORDS.
5 (5)	ADDRESS	3	DEBDEBB	– ADDRESS OF THE NEXT DEB IN THE SAME TASK
8 (8)	ADDRESS	4	DEBIRBAD (0)	– IRB STORAGE ADDRESS USED FOR APPENDAGE ASYNCHRONOUS EXITS
8 (8)	BITSTRING	1	DEBOFLGS	– DATA SET STATUS FLAGS
	11.		DEBDISP	″X'C0'″– DATA SET DISPOSITION FLAGS BIT SETTING DISPOSITION
	DEBDSOLD 01 OLD DATA SET DEBDSMOD 10 MOD DATA SET DEBDSNEW 11 NEW DATA SET			
	.1.		DEBDSOLD	″X'40'″ OLD DATA SET
	1.		DEBDSMOD	″X'80'″ MOD DATA SET
	11.		DEBDSNEW	″X'C0'″ NEW DATA SET
	. .1.		DEBEOF	″X'20'″– END–OF–FILE (EOF) ENCOUNTERED (TAPE INPUT) FORMAT 1 DSCB BIT 93.0 INDICATES THAT THE CURRENT VOLUME IS THE LAST VOLUME OF THE DATA SET (DASD INPUT)
	. . .1		DEBRLSE	″X'10'″– RELEASE UNUSED EXTERNAL STORAGE (DASD) EMULATOR TAPE WITH SECOND GENERATION FORMAT. TAPE MAY CONTAIN BLOCKS SHORTER THAN 12 CHARACTERS. (TAPE)
	. . . . 1. . .		DEBDCB	″X'08'″– DCB MODIFICATION
	. . . . .1. .		DEBSPLIT	″X'04'″– SPLIT CYLINDER (DASD) 7–TRACK EMULATOR TAPE WITH POSSIBLE MIXED PARITY RECORDS (TAPE)
	. . . . . .1.		DEBLABEL	″X'02'″– NONSTANDARD LABELS
	1		DEBRERR	″X'01'″– USE REDUCED ERROR RECOVERY PROCEDURE (TAPE) CONCATENATED PARTITIONED ORGANIZATION DATA SETS PROCESSED USING BPAM (DASD)
9 (9)	ADDRESS	3	DEBIRBB	– IRB STORAGE ADDRESS USED FOR APPENDAGE ASYNCHRONOUS EXITS

Offset	Type/Value	Len	Name (Dim)	Description
12 (C)	BITSTRING	1	DEBOPATB	– FLAGS INDICATING BOTH THE METHOD OF I/O PROCESSING AND THE DISPOSITION THAT IS TO BE PERFORMED WHEN AN END–OF–VOLUME (EOV) CONDITION OCCURS
	1.		DEBABEND	″X'80'″– SET BY ABEND INDICATING A SYSABEND OR SYSUDUMP DATA SET (OS/VS2)
	.1.		DEBZERO	″X'40'″– ALWAYS ZERO
	. .11		DEBPOSIT	″X'30'″– DATA SET POSITIONING FLAGS BIT SETTING POSITIONING
	DEBRERED 01 REREAD DEBLEAVE 11 LEAVE			
	. . .1		DEBRERED	″X'10'″ REREAD
	. .11		DEBLEAVE	″X'30'″ LEAVE
	 1111		DEBACCS	″X'0F'″– TYPE OF I/O ACCESSING BEING DONE BIT SETTING ACCESSING
	 1111		DEBOUTPT	″X'0F'″ OUTPUT
	. . . . 111.		DEBXTEND	″X'0E'″ EXTEND
	11		DEBINOUT	″X'03'″ INOUT
	111		DEBOUTIN	″X'07'″ OUTIN
	. . . . .11.		DEBOTINX	″X'06'″ OUTINX
	1		DEBRDBCK	″X'01'″ RDBACK
	. . . . .1. .		DEBUPDAT	″X'04'″ UPDAT
13 (D)	BITSTRING	1	DEBQSCNT	– PURGE (SVC 16) – QUIESCE COUNT. NUMBER OF DEVICES EXECUTING USER'S CHANNEL PROGRAMS, AS SHOWN BY BITS 5 AND 6 OF UCBFL1 FIELDS.
14 (E)	BITSTRING	1	DEBFLGS1	– FLAG FIELD
	1.		DEBPWCKD	″X'80'″– PASSWORD WAS SUPPLIED DURING OPEN. EOV WILL NOT REQUEST A PASSWORD FOR EACH ADDITIONAL VOLUME OF A MULTIVOLUME DATA SET.
	.1.		DEBEOFDF	″X'40'″– SET BY EOV TO INFORM CLOSE THAT AN END– OF–FILE HAS BEEN ENCOUNTERED AND, THEREFORE, DEFERRED USER LABEL PROCESSING IS ALLOWED.
	. .1.		DEBRSIOA	″X'20'″– SIO APPENDAGE RE–ENTRY AUTHORIZATION BIT (OS/VS1) (MDC331)
	. . .1		DEBEXCPA	″X'10'″– EXCP IS AUTHORIZED FOR THIS DEB
	. . . . 1. . .		DEBCINDI	″X'08'″– DCB ASSOCIATED WITH THIS DEB IS BEING PROCESSED BY THE COMPATIBILITY INTERFACE ROUTINES (VSAM OR SAM) ICB462
	. . . . .1. .		DEBF1CEV	″X'04'″– EOV PROCESSING OCCURRED DURING CLOSE PROCESSING. TESTED AND SET TO ZERO BY CLOSE, SET TO ONE BY EOV. (ICB353) A42228
	. . . . . .1.		DEBAPFIN	″X'02'″– IF ON, AUTHORIZED PROGRAMS CAN BE LOADED MDC001
	1		DEBXTNIN	″X'01'″– IF ONE, DEB EXTENSION EXISTS (OS/VS2) MDC007
15 (F)	BITSTRING	1	DEBFLGS2	– FLAG FIELD TWO
	1.		DEBIOPAV	″X'80'″ THE I/O PREVENTION IDENTIFIER (IOPID) IN THE DEB EXTENSION IS VALID AND DEBXTNIN IS ON
	.1.		DEBFGRV1	″X'40'″ RESERVED
	. .1.		DEBFGRV2	″X'20'″ RESERVED
	. . .1		DEBFGRV3	″X'10'″ RESERVED

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . 1. . .		DEBVRLS	RLS ACCESS (VSAM ONLY). DEBXAMB POINTS TO A VSAM CONTROL BLOCK THE FOLLOWING TWO TAPE BITS ARE USED BY O/C/EOV TO MAINTAIN A UNIFORM RECORDING MODE (COMPACTION OR NON–COMPACTION) ON TAPE DATA SETS THAT SPAN OVER MORE THAN ONE VOLUME.
	. . . . 1. . .		DEBDSCMP	″X'08'″ TAPE DATA SET COMPACTION MODE
	. . . . .1. .		DEBDSNCP	″X'04'″ TAPE DATA SET NON–COMPACTION MODE
	. . . . . .1.		DEBFGRV6	″X'02'″ RESERVED
	1		DEBFGRV7	″X'01'″ RESERVED
16 (10)	ADDRESS	4	DEBUSRPG (0)	– ADDRESS OF FIRST IOB IN THE USER PURGE CHAIN (OS/VS1) ADDRESS OF PURGED I/O RESTORE LIST (PIRL) (OS/VS2) (MDC029) YM3158
16 (10)	BITSTRING	1	DEBNMEXT	– NUMBER OF EXTENTS DESCRIPTIONS STARTING AT DEBBASND. ONE EXTENT PER UNIT FOR EXTENDED FORMAT OR PDSE DATA SETS.
17 (11)	ADDRESS	3	DEBUSRPB	– ADDRESS OF FIRST IOB IN THE USER PURGE CHAIN (OS/VS1) ADDRESS OF PURGED I/O RESTORE LIST (PIRL) (OS/VS2) (MDC030) YM3158
20 (14)	ADDRESS	4	DEBRRQ (0)	– POINTER TO RELATED REQUEST QUEUE (OS/VS2) MDC026
20 (14)	ADDRESS	4	DEBECBAD (0)	– ADDRESS OF A PARAMETER LIST USED TO LOCATE THE PURGE ECB FOR AN SVC PURGE REQUEST (OS/VS1) MDC025
20 (14)	BITSTRING	1	DEBPRIOR	– PRIORITY OF THE TASK OWNING DEB
21 (15)	ADDRESS	3	DEBECBB	– ADDRESS OF A PARAMETER LIST USED TO LOCATE THE PURGE ECB FOR AN SVC PURGE REQUEST(OS/VS1) MDC025
24 (18)	ADDRESS	4	DEBDCBAD (0)	– ADDRESS OF DCB OR ACB ASSOCIATED WITH THIS DEB
24 (18)	BITSTRING	1	DEBPROTG (0)	– TASK PROTECTION KEY IN HIGH–ORDER 4 BITS
24 (18)	BITSTRING	1	DEBDEBID	– A HEX F IN LOW–ORDER 4 BITS TO IDENTIFY THIS BLOCK AS A DEB
25 (19)	ADDRESS	3	DEBDCBB	– ADDRESS OF DCB OR ACB ASSOCIATED WITH THIS DEB
28 (1C)	ADDRESS	4	DEBAPPAD (0)	– ADDRESS OF THE I/O APPENDAGE VECTOR TABLE
28 (1C)	ADDRESS	1	DEBEXSCL	THIS FIELD IS USED TO DETERMINE THE SIZE OF THE DEVICE DEPENDENT SECTION. TWO TO THIS POWER GIVES THE LENGTH OF THE DEVICE–DEPENDENT SECTION AT DEBBASND. EXTENT SCALE – 4 (16 BYTES) FOR DIRECT ACCESS DEVICE AND 3525 CARD PUNCH WITH DEVICE–ASSOCIATED DATA SET SUPPORT AND 2 (4 BYTES) FOR NONDIRECT ACCESS DEVICE AND COMMUNICATION DEVICE.
29 (1D)	ADDRESS	3	DEBAPPB	– ADDRESS OF THE I/O APPENDAGE VECTOR TABLE
	UNIT RECORD, MAGNETIC TAPE, TELECOMMUNICATIONS DEVICES SECTION NOTE FOR TELECOMMUNICATIONS DEVICES, THE UCB ADDRESS IS REPEATED FOR EACH LINE ASSIGNED			
32 (20)	ADDRESS	4	DEBSUCBA (0)	– ADDRESS OF A UCB ASSOCIATED WITH A GIVEN DATA SET
32 (20)	BITSTRING	1	DEBSDVM	– DEVICE MODIFIER. FOR MAGNETIC TAPE, MODESET OPERATION CODE OR MODE SET FUNCTION BYTE. FOR UNIT RECORD, NOT USED. 9–TRACK TAPE MODESET OPERATION CODES

Offset	Type/Value	Len	Name (Dim)	Description
	1 1.1 . .11		DEBMTDN4	″X'D3'″ 9–TRACK MODESET CCW CODE DENSITY=6250BPI
	11. . . .11		DEBMTDN3	″X'C3'″ 9–TRACK MODESET CCW CODE DENSITY=1600BPI
	11. . 1.11		DEBMTDN2	″X'CB'″ 9–TRACK MODESET CCW CODE DENSITY= 800BPI 7–TRACK TAPE MODESET SKELETON CODES (MUST BE COMPLETED WITH PARITY, TRANSLATION AND/OR CONVERSION)
	11		DEBM7DN0	″X'03'″ 7–TRACK MODESET SKELETON DENSITY=200BPI
	.1. . . .11		DEBM7DN1	″X'43'″ 7–TRACK MODESET SKELETON DENSITY=556BPI
	1. 11		DEBM7DN2	″X'83'″ 7–TRACK MODESET SKELETON DENSITY=800BPI 3480 TAPE OPERATION CODE
	11. . . .11		DEBMSTWI	″X'C3'″ 3480 SET TAPE WRITE IMMEDIATE CCW CODE TAPE MODE SET FUNCTION BYTE
	1.		DEBMTRF0	″X'80'″ TAPE RECORDING FORMAT BIT 0
	.1.		DEBMTRF1	″X'40'″ TAPE RECORDING FORMAT BIT 1
	. .1.		DEBMTWI	″X'20'″ TAPE WRITE IMMEDIATE (NON–BUFFERED WRITE)
	. . .1		DEBMINHS	″X'10'″ INHIBIT SUPERVISOR COMMANDS
	. . . . 1. . .		DEBMCOMP	″X'08'″ COMPACTED RECORDING MODE
	. . . . . .1.		DEBM3424	″X'02'″ 3424 MODE SET FLAG
	. . . . . .1.		DEBM9348	″X'02'″ 9348 MODE SET FLAG
	1		DEBMINHE	″X'01'″ INHIBIT CONTROL UNIT ERP
	X'C2'		DEBM6250	″DEBMTRF0+DEBMTRF1+DEBM3424″SET 3424 DENSITY=6250BPI
	X'42'		DEBM1600	″DEBMTRF1+DEBM3424″ SET 3424 DENSITY=1600BPI
	X'C2'		DEBM4TRK	RESERVED
	X'82'		DEBM2TRK	RESERVED
	X'42'		DEBM1TRK	RESERVED
33 (21)	ADDRESS	3	DEBSUCBB	ADDRESS OF A UCB ASSOCIATED WITH A GIVEN DATA SET
36 (24)	ADDRESS	4	DEBDEVED(0)	END OF COMMON UNIT RECORD FIELDS (MDC330)
THE FOLLOWING FIELDS ARE PRESENT ONLY FOR THE 3525 WITH DEVICE–ASSOCIATED DATA SET SUPPORT				
36 (24)	ADDRESS	4	DEBRDCB (0)	– ADDRESS OF DCB FOR THE READ ASSOCIATED DATA SET ICB394
36 (24)	BITSTRING	1	DEBRSV06	– RESERVED ICB394
37 (25)	ADDRESS	3	DEBRDCBA	– ADDRESS OF DCB FOR THE READ ASSOCIATED DATA SET ICB394
40 (28)	ADDRESS	4	DEBPDCB (0)	– ADDRESS OF DCB FOR THE PUNCH ASSOCIATED DATA SET ICB394
40 (28)	BITSTRING	1	DEBRSV07	– RESERVED ICB394
41 (29)	ADDRESS	3	DEBPDCBA	– ADDRESS OF DCB FOR THE PUNCH ASSOCIATED DATA SET ICB394
44 (2C)	ADDRESS	4	DEBWDCB (0)	– ADDRESS OF DCB FOR THE PRINT ASSOCIATED DATA SET ICB394
44 (2C)	BITSTRING	1	DEBRSV08	– RESERVED ICB394
45 (2D)	ADDRESS	3	DEBWDCBA	– ADDRESS OF DCB FOR THE PRINT ASSOCIATED DATA SET ICB394

Offset	Type/Value	Len	Name (Dim)	Description
	3540 ACCESS METHOD DEPENDENT SECTION (OS/VS1 ONLY) NOTE THIS SECTION FOLLOWS DEBSUCBA IN UNIT RECORD, MAGNETIC TAPE, TELECOMMUNICATIONS DEVICES SECTION IF DEB IS FOR 3540 DEVICE.			
36 (24)	CHARACTER	16	DEBASC09 (0)	– 3540 ACCESS METHOD DEPENDENT SECTION (MDC317)
36 (24)	CHARACTER	1	DEBVOLAC	– VOLUME ACCESSIBILITY INDICATOR (MDC327)
37 (25)	CHARACTER	1	DEBDSSQL	– DATA SET SECURITY QUALIFIER (MDC328)
38 (26)	SIGNED	1	DEBVSEQU	– VOLUME SEQUENCE NUMBER (MDC318)
39 (27)	BITSTRING	1	DEBEAMFG	– FLAG BYTE (MDC319)
	1.		DEBMULTI	″X'80'″– MULTI–VOLUME INDICATOR (MDC320)
	.1.		DEBDSOPN	″X'40'″– DATA SET IS OPEN (MDC321)
	. .1.		DEBVAMSG	″X'20'″– VOLUME ACCESSIBILITY MESSAGE HAS BEEN ISSUED (MDC329)
	. . .1		DEBSECVL	″X'10'″– SECURE VOLUME (MDC332)
40 (28)	CHARACTER	8	DEBDSID (0)	– DATA SET IDENTIFIER (DSID) (INPUT) (MDC305)
40 (28)	CHARACTER	6	DEBEXDTE	– EXPIRATION DATE (OUTPUT) (MDC333)
46 (2E)	CHARACTER	1	DEBWTPTI	– WRITE PROTECT INDICATOR (OUTPUT) (MDC334)
47 (2F)	CHARACTER	1	DEBRV008	– RESERVED (OUTPUT) (MDC335)
48 (30)	CHARACTER	4	DEBEOD (0)	– END OF DATA (EOD) ADDRESS (INPUT) (MDC306)
48 (30)	CHARACTER	4	DEBBOE (0)	– BEGINNING OF EXTENT (BOE) ADDRESS (OUTPUT) (MDC322)
48 (30)	BITSTRING	1	DEBEODRV (0)	– RESERVED (MDC307)
48 (30)	BITSTRING	1	DEBBOERV	– RESERVED (MDC323)
49 (31)	BITSTRING	1	DEBEODTT (0)	– EOD TRACK NUMBER (MDC308)
49 (31)	BITSTRING	1	DEBBOETT	– BOE TRACK NUMBER (MDC324)
50 (32)	BITSTRING	1	DEBEOD0 (0)	– MUST BE ZERO (MDC309)
50 (32)	BITSTRING	1	DEBBOE0	– MUST BE ZERO (MDC325)
51 (33)	BITSTRING	1	DEBEODSS (0)	– EOD SECTOR NUMBER (MDC310)
51 (33)	BITSTRING	1	DEBBOESS	– BOE SECTOR NUMBER (MDC326)
	DEB ISAM DEPENDENT SECTION NOTE: PRESENT ONLY IF ISAM IS USED. FOLLOWS THE BASIC SECTION AND PRECEDES THE DIRECT ACCESS STORAGE DEVICE SECTION. COUNTED AS ONE EXTENT IN DEBNMEXT.			
32 (20)	ADDRESS	4	DEBFIEAD(0)	– ADDRESS OF FIRST INDEX EXTENT
32 (20)	BITSTRING	1	DEBNIEE	– NUMBER OF EXTENTS OF INDEPENDENT INDEX AREA
33 (21)	ADDRESS	3	DEBFIEB	– ADDRESS OF FIRST INDEX EXTENT
36 (24)	ADDRESS	4	DEBFPEAD (0)	– ADDRESS OF THE FIRST PRIME DATA EXTENT
36 (24)	BITSTRING	1	DEBNPEE	– NUMBER OF EXTENTS OF PRIME DATA AREA (M=0 EXTENT)
37 (25)	ADDRESS	3	DEBFPEB	– ADDRESS OF THE FIRST PRIME DATA EXTENT
40 (28)	ADDRESS	4	DEBFOEAD (0)	– ADDRESS OF THE FIRST OVERFLOW EXTENT
40 (28)	BITSTRING	1	DEBNOEE	– NUMBER OF EXTENTS OF INDEPENDENT OVERFLOW AREA
41 (29)	ADDRESS	3	DEBFOEB	– ADDRESS OF THE FIRST OVERFLOW EXTENT
44 (2C)	ADDRESS	4	DEBEXPT (0)	– ADDRESS OF ISAM DEB EXTENSION ICB379
44 (2C)	BITSTRING	1	DEBRPSID	– ROTATIONAL POSITION SENSING (RPS) DEVICE INDICATORS
	1.		DEBRPSP	″X'80'″– PRIME DATA AREA IS ON RPS DEVICE
	.1.		DEBRPSI	″X'40'″– INDEPENDENT INDEX AREA IS ON RPS DEVICE

Offset	Type/Value	Len	Name (Dim)	Description
	. .1.		DEBRPSO	″X'20'″– INDEPENDENT OVERFLOW AREA IS ON RPS DEVICE
	. . .1		DEBRPSAP	″X'10'″– RPS SIO APPENDAGE HAS BEEN LOADED
	. . . . 1. . .		DEBRSV09	″X'08',,C'X'″RESERVED
	. . . . .1. .		DEBRSV10	″X'04',,C'X'″RESERVED
	. . . . . .1.		DEBRSV11	″X'02',,C'X'″RESERVED
	1		DEBRSV12	″X'01',,C'X'″RESERVED
45 (2D)	ADDRESS	3	DEBEXPTA	– ADDRESS OF ISAM DEB EXTENSION ICB379
	DIRECT–ACCESS STORAGE DEVICE SECTION			
	NOTE IF ISAM IS BEING USED, THIS SECTION FOLLOWS THE ISAM DEVICE DEPENDENT SECTION. OTHERWISE, IT FOLLOWS THE BASIC SECTION. THERE IS ONE OF THESE SECTIONS FOR EACH EXTENT.			
0 (0)	ADDRESS	4	DEBUCBAD (0)	– ADDRESS OF UCB ASSOCIATED WITH THIS DATA EXTENT
0 (0)	BITSTRING	1	DEBDVMOD	– DEVICE MODIFIER – FILE MASK
1 (1)	ADDRESS	3	DEBUCBA	– ADDRESS OF UCB ASSOCIATED WITH THIS DATA EXTENT
4 (4)	CHARACTER	2	DEBBINUM	– BIN NUMBER
6 (6)	CHARACTER	2	DEBSTRCC	– CYLINDER ADDRESS FOR THE START OF AN EXTENT. FOR A PDSE OR EXTENDED FORMAT DATA SET, BIT ZERO OF THE FIRST BYTE IS ON.
8 (8)	CHARACTER	2	DEBSTRHH	– TRACK ADDRESS FOR THE START OF AN EXTENT.
10 (A)	CHARACTER	2	DEBENDCC	– CYLINDER ADDRESS FOR THE END OF AN EXTENT. FOR A PDSE THIS FIELD IS RESERVED. FOR AN EXTENDED FORMAT DATA SET, THIS FIELD CONTAINS THE CYLINDER NUMBER OF THE FORMAT-1 DSCB ADDRESS.
12 (C)	CHARACTER	2	DEBENDHH	– TRACK ADDRESS FOR THE END OF AN EXTENT. FOR A PDSE THIS FIELD IS RESERVED. FOR AN EXTENDED FORMAT DATA SET, THIS FIELD CONTAINS THE TRACK NUMBER OF THE FORMAT-1 DSCB ADDRESS.
14 (E)	CHARACTER	2	DEBNMTRK	– NUMBER OF TRACKS ALLOCATED TO A GIVEN EXTENT. FOR SPLIT CYLINDER DATA SETS, THIS FIELD REPRESENTS THE NUMBER OF TRACKS BETWEEN THE START ADDRESS OF THE EXTENT AND THE END ADDRESS OF THE EXTENT.
	EXCP ACCESS METHOD, BSAM AND QSAM DEPENDENT SECTION			
0 (0)	CHARACTER	2	DEBVOLSQ (0)	– VOLUME SEQUENCE NUMBER FOR MULTIVOLUME SEQUENTIAL DATA SETS
0 (0)	BITSTRING	1	DEBVOLBT	– FIRST BYTE OF DEBVOLSQ (MDC016) YA00318
	1.		DEBEXFUL	″X'80''″– SET BY EOV WHEN REWRITING AN OLD DIRECT ACCESS DATA SET TO INDICATE THAT ALL PREVIOUS EXISTING EXTENTS HAVE BEEN FILLED (MDC017) YA00318
	.1.		DEBRSV36	″X'40',,C'X'″– RESERVED MDC018
	. .1.		DEBRSV37	″X'20',,C'X'″– RESERVED MDC018
	. . .1		DEBRSV38	″X'10',,C'X'″– RESERVED MDC018
	. . . . 1. . .		DEBRSV39	″X'08',,C'X'″– RESERVED MDC018
	. . . . .1. .		DEBRSV40	″X'04',,C'X'″– RESERVED MDC018
	. . . . . .1.		DEBRSV41	″X'02',,C'X'″– RESERVED MDC018
	1		DEBRSV42	″X'01',,C'X'″– RESERVED MDC018

Offset	Type/Value	Len	Name (Dim)	Description
1 (1)	SIGNED	1	DEBVLSEQ	– FOR DIRECT ACCESS, SEQUENCE NUMBER OF THE VOLUME OF THE DATA SET RELATIVE TO THE FIRST VOLUME OF THE DATA SET. FOR TAPE, SEQUENCE NUMBER OF THE VOLUME OF THE DATA SET RELATIVE TO THE FIRST VOLUME PROCESSED.(MDC019) YA00318
2 (2)	CHARACTER	2	DEBVOLNM	– TOTAL NUMBER OF VOLUMES IN A MULTIVOLUME SEQUENTIAL DATA SET.
4 (4)	CHARACTER	8	DEBDSNM (0)	– MEMBER NAME. THIS FIELD APPEARS ONLY WHEN AN OUTPUT DATA SET HAS BEEN OPENED FOR A MEMBER NAME AND THE DSCB SPECIFIES A PARTITIONED DATA SET.
4 (4)	ADDRESS	4	DEBUTSAA (0)	– ADDRESS OF THE USER TOTALING SAVE AREA
4 (4)	BITSTRING	1	DEBRSV13	– RESERVED
5 (5)	ADDRESS	3	DEBUTSAB	– ADDRESS OF THE USER TOTALING SAVE AREA
8 (8)	BITSTRING	4	DEBRSV14	– RESERVED (IF USER TOTALING WAS SPECIFIED)
12 (C)	SIGNED	2	DEBBLKSI	– MAXIMUM BLOCK SIZE (MDC346)
14 (E)	SIGNED	2	DEBLRECL	– LOGICAL RECORD LENGTH (MDC347)
BPAM DEPENDENT SECTION				
0 (0)	CHARACTER	1	DEBEXTNM (0)	– FOR A PARTITIONED DATA SET OPENED FOR INPUT, EACH ONE–BYTE FIELD CONTAINS THE EXTENT NUMBER OF THE FIRST EXTENT ENTRY FOR EACH DATA SET EXCEPT THE FIRST, IF TWO OR MORE DATA SETS ARE CONCATENATED. THE NUMBER OF BYTES IN THE FIELD IS EQUAL TO ONE LESS THAN THE NUMBER OF DATA SETS CONCATENATED.
0 (0)	CHARACTER	8	DEBDSNAM	– FOR A PARTITIONED DATA SET OPENED FOR OUTPUT FOR A MEMBER NAME, THIS FIELD IS THE MEMBER NAME.
BDAM DEPENDENT SECTION FOR FIXED LENGTH RECORDS WITH THE OPTION OF RELATIVE BLOCK ADDRESSING (BUT NOT TRACK OVERFLOW)				
0 (0)	SIGNED	4	DEBDBLK (0)	– ONE FOUR–BYTE FIELD FOR EACH EXTENT DESCRIBED IN THE DEVICE DEPENDENT SECTION
0 (0)	ADDRESS	1	DEBDBPT	– NUMBER OF BLOCKS PER TRACK
1 (1)	CHARACTER	3	DEBDBPE	– NUMBER OF BLOCKS PER EXTENT
FOR FIXED LENGTH RECORDS WITH THE OPTION OF RELATIVE BLOCK ADDRESSING AND TRACK OVERFLOW.				
0 (0)	SIGNED	4	DEBDTPP	– NUMBER OF TRACKS PER PERIOD
4 (4)	SIGNED	4	DEBDBPP	– NUMBER OF BLOCKS PER PERIOD THE FOLLOWING FIELD OCCURS ONCE FOR EACH EXTENT.
8 (8)	SIGNED	4	DEBDBPEF	– NUMBER OF BLOCKS PER EXTENT
BTAM DEPENDENT SECTION				
NOTE THIS SEGMENT IS ALWAYS PRESENT FOR BTAM. IT IS USED WHEN A BUFFER POOL OR DYNAMIC BUFFERING IS USED. OTHERWISE, THE FIELDS ARE ZERO.				
0 (0)	ADDRESS	4	DEBTBFRA (0)	– ADDRESS OF THE BUFFER ROUTINE
0 (0)	BITSTRING	1	DEBRSV15	– RESERVED

Offset	Type/Value	Len	Name (Dim)	Description
1 (1)	ADDRESS	3	DEBTBFRB	– ADDRESS OF THE BUFFER ROUTINE THE FOLLOWING FIELD IS REPEATED FOR EACH CCW ON THE CHANNEL PROGRAM QUEUE
4 (4)	ADDRESS	4	DEBTCCWA (0)	– ADDRESS OF THE FIRST (OR FOLLOWING) CCW ON THE QUEUE
4 (4)	BITSTRING	1	DEBRSV16	– RESERVED
5 (5)	ADDRESS	3	DEBTCCWB	– ADDRESS OF THE FIRST (OR FOLLOWING) CCW ON THE QUEUE
	GAM DEPENDENT SECTION			
0 (0)	ADDRESS	4	DEBFUCBA (0)	– ADDRESS OF FIRST UCB
0 (0)	BITSTRING	1	DEBRSV17	– RESERVED
1 (1)	ADDRESS	3	DEBFUCBB	– ADDRESS OF FIRST UCB
4 (4)	ADDRESS	4	DEBLUCBA (0)	– ADDRESS OF LAST UCB
4 (4)	BITSTRING	1	DEBRSV18	– RESERVED
5 (5)	ADDRESS	3	DEBLUCBB	– ADDRESS OF LAST UCB
	ISAM LOAD MODE EXTENSION POINTED TO BY DEBEXPT			
0 (0)	ADDRESS	4	DEBDCBFA	– ADDRESS OF DCB FIELD AREA MDC013
4 (4)	ADDRESS	4	DEBPUT	– ADDRESS OF PUT MODULE ICB379
	ISAM SCAN MODE EXTENSION POINTED TO BY DEBEXPT			
0 (0)	ADDRESS	4	DEBDCBFA	– ADDRESS OF DCB FIELD AREA MDC014
4 (4)	ADDRESS	4	DEBGET	– ADDRESS OF GET OR PUT MODULE – THIS FIELD IS ALSO CALLED DEBPUT ICB379
8 (8)	ADDRESS	4	DEBWKPT4	– SAME AS DCBWKPT4 – ADDRESS OF UCB ICB379
12 (C)	ADDRESS	4	DEBWKPT5	– SAME AS DCBWKPT5 – ADDRESS OF GET APPENDAGE MODULE ICB379
16 (10)	ADDRESS	4	DEBCREAD	– ADDRESS OF CHANNEL-END APPENDAGE FOR READ ICB379
20 (14)	ADDRESS	4	DEBCSETL	– ADDRESS OF CHANNEL–END APPENDAGE FOR SETL ICB379
24 (18)	ADDRESS	4	DEBCWRIT	– ADDRESS OF CHANNEL–END APPENDAGE FOR WRITE ICB379
28 (1C)	ADDRESS	4	DEBCCHK	– ADDRESS OF CHANNEL–END APPENDAGE FOR WRITE VALIDITY CHECK ICB379
32 (20)	ADDRESS	4	DEBCREWT	– ADDRESS OF CHANNEL–END APPENDAGE FOR RE–WRITE ICB379
36 (24)	ADDRESS	4	DEBCRECK	– ADDRESS OF CHANNEL–END APPENDAGE FOR RE–CHECK ICB379
40 (28)	ADDRESS	4	DEBAREAD	– ADDRESS OF ABNORMAL–END APPENDAGE FOR READ ICB379
44 (2C)	ADDRESS	4	DEBASETL	– ADDRESS OF ABNORMAL–END APPENDAGE FOR SETL ICB379
48 (30)	ADDRESS	4	DEBAWRIT	– ADDRESS OF ABNORMAL–END APPENDAGE FOR WRITE ICB379
52 (34)	ADDRESS	4	DEBACHK	– ADDRESS OF ABNORMAL–END APPENDAGE FOR WRITE VALIDITY CHECK ICB379
56 (38)	ADDRESS	4	DEBAREWT	– ADDRESS OF ABNORMAL–END APPENDAGE FOR RE–WRITE ICB379
60 (3C)	ADDRESS	4	DEBARECK	– ADDRESS OF ABNORMAL–END APPENDAGE FOR RE–CHECK ICB379
64 (40)	ADDRESS	4	DEBRPSST	– ADDRESS OF RPS SIO APPENDAGE IF ADDRSPC=REAL WAS NOT SPECIFIED ICB379
	BISAM MODE EXTENSION POINTED TO BY DEBEXPT			

Offset	Type/Value	Len	Name (Dim)	Description
0 (0)	ADDRESS	4	DEBDCBFA	– ADDRESS OF DCB FIELD AREA MDC015
4 (4)	ADDRESS	4	DEBDISAD	– ADDRESS OF PRIVILEGED MODULE ENTERED WHEN A BISAM MACRO INSTRUCTION IS EXECUTED ICB379
8 (8)	ADDRESS	4	DEBWKPT4	– SAME AS DCBWKPT4 – ADDRESS OF THE PART 1 APPENDAGE MODULE (ABNORMAL AND CHANNEL–END APPENDAGES) ICB379
12 (C)	ADDRESS	4	DEBWKPT5	– SAME AS DCBWKPT5 – ADDRESS OF THE PART 2 APPENDAGE MODULE (ABNORMAL AND CHANNEL–END APPENDAGES) ICB379
16 (10)	ADDRESS	4	DEBFREED	– ADDRESS OF DYNAMIC BUFFERING MODULE ICB379
20 (14)	ADDRESS	4	DEBRPSIO	– ADDRESS OF RPS SIO APPENDAGE MODULE IF ADDRSPC=REAL WAS NOT SPECIFIED AND IF DYNAMIC BUFFERING IS USED ICB379
24 (18)	ADDRESS	4	DEBSIOA2	– ADDRESS OF DYNAMIC BUFFERING APPENDAGE MODULE DISPLACED BY PAGE FIX (PGFX) APPENDAGE IF ADDRSPC=REAL WAS NOT SPECIFIED ICB379
SUBROUTINE NAME SECTION NOTE FOLLOWS THE ACCESS METHOD DEPENDENT SECTION, OR THE DEVICE DEPENDENT SECTION IF THERE IS NO ACCESS METHOD SECTION				
0 (0)	CHARACTER	2	DEBSUBID	– SUBROUTINE IDENTIFICATION. EACH ACCESS METHOD SUBROUTINE, APPENDAGE SUBROUTINE, AND IRB ROUTINE WILL HAVE A UNIQUE EIGHT–BYTE NAME. THE LOW–ORDER TWO BYTES OF EACH ROUTINE NAME WILL BE IN THIS FIELD IF THE SUBROUTINE IS LOADED BY THE OPEN ROUTINES.
DEB EXTENSION (OS/VS2) POINTED TO BY DEBXTNP				
0 (0)	SIGNED	2	DEBXLNGH	– LENGTH OF DEB EXTENSION MDC002
2 (2)	BITSTRING	1	DEBXFLG1	– FLAG BYTE (MDC027) YM1272
	1.		DEBXCDCB	″X'80'″– DEBDCBAD FIELD CONTAINS THE ADDRESS OF A COPIED DCB. USED BY END–OF–VOLUME, TCLOSE AND TASK CLOSE. (MDC028) YM1272
	.1.		DEBXTSKC	″X'40'″– TASK CLOSE IS CLOSING THE RELATED DCB. SET BY TASK CLOSE AND INTERROGATED BY END–OF–VOLUME, FEOV AND TCLOSE FOR DEB'S NOT ON THE CURRENT TCB DEB CHAIN. (MDC031) YM2869
	. .1.		DEBXDSSI	″X'20'″– DATA SET SECURITY INDICATOR. SET BY OPEN AND CHECKPOINT. INTERROGATED BY EOV. (MDC348)
	. . .1		DEBXWIND	″X'10'″– MSS WINDOW PROCESSING INDICATOR. SET BY OPEN. INTERROGATED BY EOV, SAM EOB AND CLOSE.
	. . . . 1. . .		DEBXACIS	″X'08'″– ACQUIRE ISSUED WITH INHIBIT STAGE INDICATED DURING OPEN OR EOV. SET BY OPEN AND EOV. INTERROGATED BY SVC 126.
	. . . . .1. .		DEBXNFLS	″X'04'″ DO NOT FLUSH QSAM BUFFERS IN CLOSE BECAUSE AN OPEN/CLOSE/EOV ABEND OCCURRED. SET BY IFG0RR0B. TESTED BY IGG0201Z.
	. . . . . .1.		DEBXSYSB	″X'02'″ SYSTEM DETERMINED BLOCK SIZE

Offset	Type/Value	Len	Name (Dim)	Description
	1		DEBXRACE	″X'01'″ RACF EXECUTE ONLY AUTHORITY OF A PROGRAM LIBRARY. SET BY OPEN AND INTERROGATED BY OPEN AFTER FINAL DCB MERGE AND BY A SAM EXECUTOR.
3 (3)	BITSTRING	1	DEBXFLG2	– FLAG BYTE
	1.		DEBXRSAP	″X'80'″– USED BY RESTART TO INDICATE THAT RESTART HAS INCREASED THE LENGTH OF THE DEB EXTENSION TO INCLUDE A CHKPT/RST APPENDAGE TO THE DEB EXTENSION. THE APPENDAGE IS FREEMAINED AND THIS BIT IS TURNED OFF BEFORE RESTART PROCESSING IS COMPLETED. DEBXLNTH INCLUDES THE LENGTH OF THE APPENDAGE WHILE THIS BIT IS ON.
	.1.		DEBBYP	″X'40'″– WHEN ON EXCP SCAN ROUTINE WILL SET IOSBYP ON AND BYPASS BUILDING A PREFIX
	. .1.		DEBCHCMP	″X'20'″– WHEN ON EXCP SCAN ROUTINE WILL SET IOSCHCMP ON AND BYPASS BUILDING A PREFIX
	. . .1		DEBXENQA	″X'10'″ ENQUEUED ON PDS, DISP=SHARE, OUTPUT
	. . . . 1. . .		DEBXENQS	″X'08'″ ENQUEUED ON DSCB,DISP=SHR
	. . . . .1. .		DEBXIMSC	″X'04',,C’X’″– IMS CLOSE IN PROGRESS
	. . . . . .1.		DEBXCASV	″X'02'″ THE CACHE ATTRIBUTE TOKEN IS VALID
	1		DEBXRV54	5″X'01',,C’X’″– RESERVED
4 (4)	ADDRESS	4	DEBXDSAB	– POINTER TO DSAB MDC004
8 (8)	BITSTRING	4	DEBXDCBM	– DCB MODIFICATION MASK USED BY I/O SUPPORT MDC005
12 (C)	ADDRESS	4	DEBXDBPR	– POINTER TO DEB MDC006
16 (10)	BITSTRING	1	DEBXDSO1	– SAME AS DCBDSORG BYTE 1 MDC020
17 (11)	BITSTRING	1	DEBXDSO2	– SAME AS DCBDSORG BYTE 2 MDC021
18 (12)	BITSTRING	1	DEBXMCF1	– SAME AS DCBMACRF BYTE 1 MDC022
19 (13)	BITSTRING	1	DEBXMCF2	– SAME AS DCBMACRF BYTE 2 MDC023
20 (14)	ADDRESS	4	DEBXXARG	– ADDRESS OF BDAM READ EXCLUSIVE LIST MDC024
24 (18)	ADDRESS	4	DEBXOPNJ	– POINTER TO DSAB (SEPARATE FROM DEBXDSAB) DYNAMICALLY ALLOCATED BY OPEN TYPE=J. THIS POINTER WILL EXIST FOR NON–AUTHORIZED CALLERS OF OPEN TYPE=J FOR A DIRECT ACCESS DATA SET WHERE THE DATA SET NAME BEING OPENED IS DIFFERENT FROM THE DATA SET DESCRIBED BY THE DDNAME IN DCB AND THE JFCNWRIT BIT IS ON IN JFCB. THE POINTER WILL BE USED BY CLOSE TO DYNAMICALLY UNALLOCATE THE DATA SET. (MDC352)
28 (1C)	ADDRESS	4	DEBXSAMB	– ADDRESS OF SAM BLOCK (SAMB) (MDC351)
32 (20)	BITSTRING	8	DEBXOPET	– DATASET OPEN TIME SET BY OPEN INITIAL
THE FOLLOWING MAPPING REPRESENTS THE FIRST 8 BYTES OF DEFINE EXTENT DATA WHICH IS PASSED VIA THE DEFINE EXTENT CCW TO A DASD STORAGE SUBSYSTEM WHICH SUPPORTS THE ECKD™ DEFINE EXTENT CCW. THIS INFORMATION WILL BE COPIED FROM THE DEB EXTENSION TO THE SYSTEM BUILT DEFINE EXTENT DATA AREA AND WILL BE SENT TO THE STORAGE SUBSYSTEM WITH EACH I/O REQUEST RELATED TO THIS DEB. REFER TO THE APPROPRIATE STORAGE CONTROL REFERENCE PUBLICATION FOR A DETAILED DESCRIPTION OF THESE FIELDS.				
40 (28)	BITSTRING	8	DEBXDEF (0)	DEFINE EXTENT DATA AREA

Offset	Type/Value	Len	Name (Dim)	Description
40 (28)	BITSTRING	1	DEBDEFG1	FLAG BYTE
	1.		DEBNSHED	″X'80'″ NO SEEK HEAD PERMITTED
	.1.		DEBXVDEF	″X'40'″ DEB DEF EXT DATA PARMS VALID. MUST BE ON FOR DX DATA TO BE USED.
41 (29)	BITSTRING	1	DEBGATTR	GLOBAL ATTRIBUTES
	11.		DEBECKD	″X'C0'″ EXTENT DEFINITION B’XX.....’ EXTENT DEFINITION
	1.		DEBGAEX1	″X'80'″ EXTENT DEFINITION 1
	.1.		DEBGAEX2	″X'40'″ EXTENT DEFINITION 2 B’00.....’ DEFINE EXTENT OPERATES AS DEFINED FOR FIXED BLOCK ARCH B’11.....’ DEFINE EXTENT OPERATES AS DEFINED FOR CKD EXT-CCHH EXTENTS
	. .1.		DEBSTRTP	″X'20'″ CKD CONVERSION MODE – FOR SYSTEM USE B’..1.....’ CKD CONVERSION MODE REQUIRED B’..0.....’ CKD CONVERSION MODE NOT REQUIRED
	. . .1 11. .		DEBGA345	″X'1C'″ ATTRIBUTES BITS 3,4,5 B’...XXX..’ ATTRIBUTE BITS 3,4,5
	. . .1		DEBGA1	″X'10'″ ATTRIBUTE 1
	. . . . 1. . .		DEBGA2	″X'08'″ ATTRIBUTE 2
	. . . . .1. .		DEBGA3	″X'04'″ ATTRIBUTE 3 B’...000..’ NORMAL CACHE ACCESS B’...001..’ BYPASS CACHE LOAD B’...010..’ INHIBIT CACHE LOAD B’...011..’ SEQUENTIAL ACCESS B’...100..’ SEQUENTIAL STAGING MODE B’...101..’ RECORD ACCESS MODE B’...110..’ RESERVED B’...111..’ RESERVED
	. . . . .		DEBNCACH	″X'00'″ NORMAL CACHE ACCESS
	. . . . .1. .		DEBBCACH	″X'04'″ BYPASS CACHE LOAD
	. . . . 1. . .		DEBICACH	″X'08'″ INHIBIT CACHE LOAD
	. . . . 11. .		DEBSCACH	″X'0C'″ SEQUENTIAL ACCESS
	. . .1		DEBXRFLG	″X'10'″ SEQUENTIAL PRESTAGE MODE
	. . .1 . 1 . .		DEBRLC	″X'14'″ RECORD ACCESS MODE
	. . . . . .1.		DEBNRD	″X'02'″ NON RETENTIVE DATA ACCESS
	1		DEBINHFW	″X'01'″ INHIBIT FAST WRITE B’.....10’ USE NON-RETENTIVE DATA B’.....11’ INHIBIT DASD FAST WRITE
42 (2A)	BITSTRING	2	DEBBLKSZ	BLOCK SIZE IN BYTES
44 (2C)	CHARACTER	4	DEBEXTOK	ZERO,EXTENT LOCATOR OR TOKEN
44 (2C)	CHARACTER	2	DEBNRDID	SUBSYSTEM FUNCTION ID: NON-RETENTIVE DATA OR CONCURRENT COPY/ XRC SESSION ID
46 (2E)	CHARACTER	1		

Offset	Type/Value	Len	Name (Dim)	Description
47 (2F)	BITSTRING	1	DEBGATTX	GLOBAL ATTRIBUTE EXTENDED 2 B’00.....’ NO SPECIAL MODE B’01.....’ REGULAR DATA FORMAT B’10.....’ SPECIAL KEY SEARCH B’11.....’ RESERVED B’..111...’ RESERVED FOR SYSTEM B’.....1..’ R0 DATA NOT REQUIRED B’.....11’ RESERVED FOR SYSTEM DEBRMODE EQU X’00’ NO SPECIAL MODE DEBRDATA EQU X’40’ REGULAR DATA FORMAT DEBSKEY EQU X’80’ SPECIAL KEY SEARCH DEBDFTR0 EQU X’04’ R0 DATA NOT REQUIRED
END OF DEFINE EXTENT DATA AREA MAPPING				
48 (30)	CHARACTER	4	DEBIOPID	I/O PREVENTION IDENTIFIER
52 (34)	CHARACTER	4	DEBBLKID	BLOCK ID VALUE USED TO CALCULATE NUMBER OF BLOCKS IN THIS VOLUME FOR EXTENDED BLOCK COUNT CHECKING
56 (38)	CHARACTER	4	DEBXCASF	CACHE ATTRIBUTE SELECTION TOKEN
60 (3C)	BITSTRING	1	DEBXFLG3	FLAG BYTE
	1.		DEBXTRNC	″X'80'″ – QSAM TRUNC MACRO HAS BEEN ISSUED FOR A PDSE – IT WAS IGNORED.
	.1.		DEBXNSEG	″X'40'″ – NULL SEGMENT ENCOUNTERED IN PDSE
	. .1.		DEBXSMSG	″X'20'″ – SMS GUARANTEED SPACE DISP NEW OR MOD
	. . . . 1. . .		DEBXCAP	UCB ADDRESS IN DEB CAPTURED BY OCE
BIT 5 – 7 – RESERVED				
61 (3D)	BITSTRING	1	DEBXAMBF	FLAG BYTE
	1.		DEBXABSS	DEBXAMB DEFINES AN SSCB POINTER, DEBXSSCB IS VALID
	.1.		DEBXARBA	DEBXAMB DEFINES AN RBA VALUE. DEBXRBA IS VALID. VALID ONLY UNTIL OPEN EXECUTORS CHANGE IT
	. .1.		DEBXAOEF	DEBXAMB DEFINES AN IGGSXCB PTR FOR HFS OR REMOTE FILE
62 (3E)	SIGNED	2	DEBXSCNT	NUMBER OF STRIPES FOR EXTENDED FORMAT
64 (40)	ADDRESS	4	DEBXSACB	POINTER TO SACB CONTROL BLOCK
68 (44)	BITSTRING	1	DEBXAFLG	″ATTRIBUTE EXTENSION″ FLAGS
	1.		DEBXADS	″ATTRIBUTE EXTENSION″ DATA SET
	.1.		DEBXAINP	INPUT PROCESSING PERFORMED
	. .1.		DEBXAOUT	OUTPUT PROCESSING PERFORMED
	. . .1		DEBXAOFO	″ATTRIBUTE EXTENSION″ DATA SET OPENED FOR OUTPUT
69 (45)		1		RESERVED
70 (46)	SIGNED	2	DEBXVLSQ	EXPECTED TAPE VOL SEQUENCE
72 (48)	POINTER	4	DEBXDSSB	ADDRESS OF DSSB
76 (4C)	POINTER	4	DEBXAMB	ACCESS METHOD BLOCK. USE OF THIS FIELD IS DEFINED BY DEBXAMBF (SEE ALSO THE NEXT 2 FIELDS)
76 (4C)	POINTER	4	DEBXSSCB	ADDRESS OF STRIPED SAM CONTROL BLOCK (SSCB)
76 (4C)	POINTER	4	DEBXRBA	RBA OF VVDS CI OF NVR
80 (50)	POINTER	4	DEBXDEBX	ADDRESS OF 2ND DEB EXTENSION
DEB 2nd EXTENSION POINTED TO BY DEBXDEBX				
0 (0)	CHARACTER	8	DEB2XTNN	IDENTIFIER(DEBXDEBX)

Offset	Type/Value	Len	Name (Dim)	Description
8 (8)	SIGNED	2	DEB2XLGH	LENGTH OF 2ND DEB EXTENSION
10 (A)	CHARACTER	32	DEB2XRSG(0)	STORAGE GROUP LENGTH AND NAME
10 (A)	SIGNED	2	DEB2XSGL	STORAGE GROUP NAME LENGTH
12 (C)	CHARACTER	30	DEB2XSGN	STORAGE GROUP NAME
42 (2A)	CHARACTER	32	DEB2XRMC(0)	MANAGEMENT CLASS LENGTH AND NAME
42 (2A)	SIGNED	2	DEB2XMCL	MANAGEMENT CLASS NAME LENGTH
44 (2C)	CHARACTER	30	DEB2XMCM	MANAGEMENT CLASS NAME
74 (4A)	CHARACTER	32	DEB2XRSC(0)	STORAGE CLASS LENGTH AND NAME
74 (4A)	SIGNED	2	DEB2XSCL	STORAGE CLASS NAME LENGTH
76 (4C)	CHARACTER	30	DEB2XSCN	STORAGE CLASS NAME
106 (6A)	CHARACTER	32	DEB2XRDC(0)	DATA CLASS LENGTH AND NAME
106 (6A)	SIGNED	2	DEB2XDCL	DATA CLASS NAME LENGTH
108 (6C)	CHARACTER	30	DEB2XDCN	DATA CLASS NAME
138 (8A)	BITSTRING	1	DEB2XFG1	FLAG BYTE
	1.		DEB2XOSM	JFCMEDIA TYPE SET BY OPEN
139 (8B)	SIGNED	1	DEB2XUOF	TIOT DD OFFSET OF UNIT SELECTED BY OPEN, CLOSE, EOV
140 (8C)	SIGNED	2	DEB2XLSQ	TAPE VOLUME SEQUENCE NUMBER FROM LABEL
142 (8E)	SIGNED	1	DEB2XALV	ISO/ANSI VERSION DURING OUTPUT
143 (8F)		1		RESERVED
144 (90)	ADDRESS	4	DEB2XTIM	POINTER TO TIME STAMP
148 (94)		3		RESERVED
151 (97)	BITSTRING	1	DEB2XCFG	ISO/ANSI VERSION 4 CCSID FLAGS
	1...		DEB2XIBM	IBM VERSION 4 CREATED TAPE
	.1..		DEB2XOUT	VERSION 4 TAPE OPENED FOR OUTPUT DISP NOT MOD
	..1.		DEB2XMOD	VERSION 4 TAPE OPENED FOR OUTPUT DISP MOD
	...1		DEB2XUDF	USER APPLICATION CCSID WAS DEFAULTED
	.... 1...		DEB2XTDF	TAPE CCSID WAS DEFAULTED
	.... .1..		DEB2XIGN	TAPE CCSID SPECIFIED BUT NO CONVERSION DONE - CCSID WAS IGNORED
152 (98)	SIGNED	4	DEB2XUSR	CCSID OF USER APPLICATION
156 (9C)	SIGNED	4	DEB2XTPE	CCSID OF TAPE
160 (A0)	SIGNED	4	DEB2XLBL	CCSID OF EXISTING TAPE LABEL
164 (A4)	ADDRESS	4	DEB2XSDC	POINTER TO SAM DATA CONVERSION BLOCK
168 (A8)		16		RESERVED

Cross-Reference

Name	Hex Offset	Hex Value	Level
DEBABEND	C	80	2
DEBACCS	C	F	2
DEBACHK	34		2
DEBAMBDM	-3	40	2
DEBAMBPM	-3	20	2
DEBAMGAM	-3	8	2
DEBAMID	0		2
DEBAMISM	-3	80	2
DEBAMLNG	4		2
DEBAMNON	-3	0	2
DEBAMSAM	-3	20	2
DEBAMSUB	-3	81	2

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Name	Hex Offset	Hex Value	Level
DEBAMTAM	-3	10	2
DEBAMTAP	-3	84	2
DEBAMTCM	-3	4	2
DEBAMTYP	-3		2
DEBAMVSM	-3	1	2
DEBAMVTM	-3	82	2
DEBAMXCP	-3	2	2
DEBAPFIN	E	2	2
DEBAPPAD	1C		2
DEBAPPB	1D		2
DEBAREAD	28		2
DEBARECK	3C		2
DEBAREWT	38		2
DEBASC09	24		2
DEBASC9E	33	58	2
DEBASDSE	2D	54	2
DEBASETL	2C		2
DEBAVT	-24		2
DEBAVTE	-13	14	2
DEBAWRIT	30		2
DEBBASIC	-2	24	2
DEBBASND	1D	44	2
DEBBCACH	29	4	2
DEBBINUM	4		2
DEBBLKID	34		2
DEBBLKSI	C		2
DEBBLKSZ	2A		2
DEBBOE	30		2
DEBBOERV	30		2
DEBBOESS	33		2
DEBBOETT	31		2
DEBBOE0	32		2
DEBBYP	3	40	2
DEBCCHK	1C		2
DEBCEA	-18		2
DEBCEAB	-18		2
DEBCEAD	-17		2
DEBCENP	-18	F	2
DEBCHCMP	3	20	2
DEBCINDI	E	8	2
DEBCMPAC	20	8	2
DEBCREAD	10		2
DEBCRECK	24		2
DEBCREWT	20		2
DEBCSETL	14		2
DEBCWRIT	18		2
DEBDASDE	E	10	2
DEBDBLK	0		2
DEBDBPE	1		2
DEBDBPEF	8		2
DEBDBPP	4		2
DEBDBPT	0		2
DEBDCB	8	8	2
DEBDCBAD	18		2

Name	Hex Offset	Hex Value	Level
DEBDCBB	19		2
DEBDCBFA	0		2
DEBDCBMK	-8		2
DEBDDS1	1D	44	2
DEBDEBAD	4		2
DEBDEBB	5		2
DEBDEBID	18		2
DEBDEFG1	28		2
DEBDEVED	24		2
DEBDISAD	4		2
DEBDISP	8	C0	2
DEBDSCBA	-F		2
DEBDSCMP	F	8	2
DEBDSID	28		2
DEBDSMOD	8	80	2
DEBDSNAM	0		2
DEBDSNCP	F	4	2
DEBDSNEW	8	C0	2
DEBDSNM	4		2
DEBDSOLD	8	40	2
DEBDSOPN	27	40	2
DEBDSSQL	25		2
DEBDTPP	0		2
DEBDVMOD	0		2
DEBEAMFG	27		2
DEBECBAD	14		2
DEBECBB	15		2
DEBECKD	29	C0	2
DEBENDCC	A		2
DEBENDHH	C		2
DEBEOD	30		2
DEBEODRV	30		2
DEBEODSS	33		2
DEBEODTT	31		2
DEBEOD0	32		2
DEBEOEA	-24		2
DEBEOEAB	-24		2
DEBEOEAD	-23		2
DEBEOENP	-24	F	2
DEBEOF	8	20	2
DEBEOFDF	E	40	2
DEBESMVR	-18	80	2
DEBEXCPA	E	10	2
DEBEXDTE	28		2
DEBEXFUL	0	80	2
DEBEXPT	2C		2
DEBEXPTA	2D		2
DEBEXSCL	1C		2
DEBEXTNM	0		2
DEBEXTOK	2C		2
DEBFGRV1	F	40	2
DEBFGRV2	F	20	2
DEBFGRV3	F	10	2
DEBFGRV6	F	2	2

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Name	Hex Offset	Hex Value	Level
DEBFGRV7	F	1	2
DEBFIEAD	20		2
DEBFIEB	21		2
DEBFIX	-20	10	2
DEBFLGS1	E		2
DEBFLGS2	F		2
DEBFOEAD	28		2
DEBFOEB	29		2
DEBFPEAD	24		2
DEBFPEB	25		2
DEBFREED	10		2
DEBFUCBA	0		2
DEBFUCBB	1		2
DEBF1CEV	E	4	2
DEBGAEX1	29	80	2
DEBGAEX2	29	40	2
DEBGATTR	29		2
DEBGA1	29	10	2
DEBGA2	29	8	2
DEBGA3	29	4	2
DEBGA345	29	1C	2
DEBGET	4		2
DEBICACH	29	8	2
DEBINHFW	29	1	2
DEBINOUT	C	3	2
DEBINPUT	C	0	2
DEBIOPAV	F	80	2
DEBIOPID	30		2
DEBIOVR	-20	20	2
DEBIRBAD	8		2
DEBIRBB	9		2
DEBISAM	34	44	2
DEBISAME	2D	54	2
DEBLABEL	8	2	2
DEBLEAVE	C	30	2
DEBLNGTH	-4		2
DEBLRECL	E		2
DEBLUCBA	4		2
DEBLUCBB	5		2
DEBMCOMP	20	8	2
DEBMINHE	20	1	2
DEBMINHS	20	10	2
DEBMSTWI	20	C3	2
DEBMTDN2	20	CB	2
DEBMTDN3	20	C3	2
DEBMTDN4	20	D3	2
DEBMTRF0	20	80	2
DEBMTRF1	20	40	2
DEBMTWI	20	20	2
DEBMULTI	27	80	2
DEBM1TRK	20	42	2
DEBM1600	20	42	2
DEBM2TRK	20	82	2
DEBM3424	20	2	2

Name	Hex Offset	Hex Value	Level
DEBM4TRK	20	C2	2
DEBM6250	20	C2	2
DEBM7DN0	20	3	2
DEBM7DN1	20	43	2
DEBM7DN2	20	83	2
DEBM9348	20	2	2
DEBNCACH	29	0	2
DEBNIEE	20		2
DEBNMEXT	10		2
DEBNMSUB	0		2
DEBNMTRK	E		2
DEBNOEE	28		2
DEBNPEE	24		2
DEBNRD	29	2	2
DEBNRDID	2C		2
DEBNSHED	28	80	2
DEBOFLGS	8		2
DEBOPATB	C		2
DEBOUTIN	C	7	2
DEBOUTPT	C	F	2
DEBPCIA	-1C		2
DEBPCIAB	-1C		2
DEBPCIAD	-1B		2
DEBPCINP	-1C	F	2
DEBPDCB	28		2
DEBPDCBA	29		2
DEBPGFX	-20	80	2
DEBPOSIT	C	30	2
DEBPREFE	-2	24	2
DEBPREFX	-10		2
DEBPRIOR	14		2
DEBPROTG	18		2
DEBPUT	4		2
DEBPWCKD	E	80	2
DEBQSCNT	D		2
DEBRDBCK	C	1	2
DEBRDCB	24		2
DEBRDCBA	25		2
DEBRERED	C	10	2
DEBRERR	8	1	2
DEBRLSE	8	10	2
DEBRPSAP	2C	10	2
DEBRPSI	2C	40	2
DEBRPSID	2C		2
DEBRPSIO	14		2
DEBRPSO	2C	20	2
DEBRPSP	2C	80	2
DEBRPSST	40		2
DEBRRQ	14		2
DEBRSIOA	E	20	2
DEBRSV06	24		2
DEBRSV07	28		2
DEBRSV08	2C		2
DEBRSV09	2C	8	2

Name	Hex Offset	Hex Value	Level
DEBRSV10	2C	4	2
DEBRSV11	2C	2	2
DEBRSV12	2C	1	2
DEBRSV13	4		2
DEBRSV14	8		2
DEBRSV15	0		2
DEBRSV16	4		2
DEBRSV17	0		2
DEBRSV18	4		2
DEBRSV24	-1C	80	2
DEBRSV25	-1C	40	2
DEBRSV26	-1C	20	2
DEBRSV27	-1C	10	2
DEBRSV29	-18	40	2
DEBRSV30	-18	20	2
DEBRSV31	-18	10	2
DEBRSV32	-14	80	2
DEBRSV33	-14	40	2
DEBRSV34	-14	20	2
DEBRSV35	-14	10	2
DEBRSV36	0	40	2
DEBRSV37	0	20	2
DEBRSV38	0	10	2
DEBRSV39	0	8	2
DEBRSV40	0	4	2
DEBRSV41	0	2	2
DEBRSV42	0	1	2
DEBRV004	27	8	2
DEBRV005	27	4	2
DEBRV006	27	2	2
DEBRV007	27	1	2
DEBRV008	2F		2
DEBSCACH	29	C	2
DEBSDVM	20		2
DEBSECVL	27	10	2
DEBSIOA	-20		2
DEBSIOAB	-20		2
DEBSIOAD	-1F		2
DEBSIOA2	18		2
DEBSIONP	-20	F	2
DEBSIOX	-20	40	2
DEBSPLIT	8	4	2
DEBSTRCC	6		2
DEBSTRHH	8		2
DEBSTRTP	29	20	2
DEBSUBID	0		2
DEBSUCBA	20		2
DEBSUCBB	21		2
DEBTAMID	0	0	2
DEBTBFRA	0		2
DEBTBFRB	1		2
DEBTBLOF	-2		2
DEBTCAMQ	-24	0	2
DEBTCBAD	0		2

Name	Hex Offset	Hex Value	Level
DEBTCBB	1		2
DEBTCCWA	4		2
DEBTCCWB	5		2
DEBUCBA	1		2
DEBUCBAD	0		2
DEBUPDAT	C	4	2
DEBUSRPB	11		2
DEBUSRPG	10		2
DEBUTSAA	4		2
DEBUTSAB	5		2
DEBVAMID	0	F	2
DEBVAMSG	27	20	2
DEBVLSEQ	1		2
DEBVOLAC	24		2
DEBVOLBT	0		2
DEBVOLNM	2		2
DEBVOLSQ	0		2
DEBVSEQU	26		2
DEBWDCB	2C		2
DEBWDCBA	2D		2
DEBWKARA	-10		2
DEBWKPT4	8		2
DEBWKPT5	C		2
DEBWTPTI	2E		2
DEBXACIS	2	8	2
DEBXADS	44	80	2
DEBXAFLG	44		2
DEBXAINP	44	40	2
DEBXAMBF	3D		2
DEBXABSS	3D	80	2
DEBXAMB	4C		2
DEBXAOFO	44	10	2
DEBXAOUT	44	20	2
DEBXARBA	3D	40	2
DEBXAOEF	3D	20	2
DEBXCAP	3C	08	2
DEBXCASF	38		2
DEBXCASV	3	2	2
DEBXCDCB	2	80	2
DEBXCEA	-14		2
DEBXCEAB	-14		2
DEBXCEAD	-13		2
DEBXCENP	-14	F	2
DEBXDBPR	C		2
DEBXDCBM	8		2
DEBXDEBX	50		2
DEBXDEF	28		2
DEBXDSAB	4		2
DEBXDSO1	10		2
DEBXDSO2	11		2
DEBXDSSB	48		2
DEBXDSSI	2	20	2
DEBXEND	54	54	2
DEBXENQA	3	10	2

OPEN/CLOSE/EOV (Common) Diagnostic Aids

Name	Hex Offset	Hex Value	Level
DEBXENQS	3	8	2
DEBXFLG1	2		2
DEBXFLG2	3		2
DEBXFLG3	3C		2
DEBXIMSC	3	4	2
DEBXLEN	54	54	2
DEBXLNGH	0		2
DEBXMCF1	12		2
DEBXMCF2	13		2
DEBXNFLS	2	4	2
DEBXNSEG	3C	40	2
DEBXOPET	20		2
DEBXOPNJ	18		2
DEBXRACE	2	1	2
DEBXRBA	4C		2
DEBXRSAP	3	80	2
DEBXRSV1	3D		2
DEBXRSV2	44		2
DEBXRSV3	48		2
DEBXRV54	3	1	2
DEBXSACB	40		2
DEBXSAMB	1C		2
DEBXSCNT	3E		2
DEBXSSCB	4C		2
DEBXSMSG	3C	20	2
DEBXSYSB	2	2	2
DEBXTNIN	E	1	2
DEBXTNP	-8		2
DEBXTRNC	3C	80	2
DEBXTSKC	2	40	2
DEBXVDEF	28	40	2
DEBXVLSQ	46		2
DEBXWIND	2	10	2
DEBXXARG	14		2
DEBZERO	C	40	2
DEB2XDCL	6A		2
DEB2XDCN	6C		2
DEB2XEND	98	98	2
DEB2XFG1	8A		2
DEB2XLEN	98	98	2
DEB2XLGH	8		2
DEB2XLSQ	8C		2
DEB2XMCL	2A		2
DEB2XMCN	2C		2
DEB2XOSM	8A	80	2
DEB2XRDC	6A		2
DEB2XRMC	2A		2
DEB2XRSC	4A		2
DEB2XRSG	A		2
DEB2XSCL	4A		2
DEB2XSCN	4C		2
DEB2XSGL	A		2
DEB2XSGN	C		2
DEB2XTIM	90		2

Name	Hex Offset	Hex Value	Level
DEB2XTNN	0		2
DEB2XUOF	8B		2

Chapter 13. HFS Diagnostic Aids

The following diagnostic aids are provided for Hierarchical File System (HFS):

- A description of the component trace facility for use with modules that manipulate HFSs
- IPCS VERBEXIT SMSXDATA dump formatting
- A sample entry in SYS1.LOGREC for an HFS problem
- Out-of-Space During Sync Processing message
- Return codes common to all HFS modules
- Reason codes common to all HFS modules

Note: HFS itself does not issue messages. All messages are issued by HFS callers.

HFS Component Trace

DFSMSdfp provides a trace service for use with all functions that manipulate HFSs. A special trace table is created in each address space where HFSs are used and tracing is selected.

You can use SYS1.PARMLIB member CTncccxx to specify trace options for the TRACE CT command and to turn off tracing. CTncccxx must exist at IPL for tracing to be available on the system. The default is CTISMS00. For more information on using CTncccxx to specify tracing options, see ″CTncccxx″ in *z/OS MVS Initialization and Tuning Reference*.

Trace also allows external writers (WTR) to capture trace buffers. Capturing trace buffers increases diagnostic capability and decreases the possibility of losing trace data.

Complete syntax and usage information for TRACE can be found in *z/OS MVS System Commands*.

TRACE Command for HFS Component Trace

The syntax of the TRACE command for tracing the HFS component follows.

```
TRACE        ,WTRSTART=membername [WRAP|NOWRAP ] ] ]
             CT [,WTRSTOP=jobname ] ]
             [CT [,ON ] { {COMP=SYSSMS } [,SUB=(sub[sub],...)]
             [,PARM=CTISMSxx] } ]
             [CT [,nnnK ] {                                  } ]
             [CT [,nnnnM] {                                  } ]
             [CT [,OFF] {                                    } ]
```

The default **SYSSMS** trace table size is 72 KB.

PARM=CTISMS*xx* identifies the SYS1.PARMLIB member that contains the tracing options.

The *nnn***K** and *nnnn***M** keywords can be used to specify trace table sizes of 16 - 999 KB or 1 - 2047 MB, respectively.

© Copyright IBM Corp. 1979, 2002

You are then prompted to specify trace options. You must respond with the REPLY command, using the following syntax.

```
REPLY      id
           [,JOBNAME=(jobnamelist)]
           ASID[,=(asidlist.)]
           [,OPTIONS=(namename...)]
           [,WTR=procname|DISCONNECT]
           [,END]
```

Notes:

1. The *id* value is the identification number, 0 - 99, specified on the prompting message.
2. You can list up to 16 jobs or ASIDs to be used as filters in tracing.
3. The REPLY can be continued on multiple lines. END must be the final parameter, to identify the end of the REPLY.

The TRACE and REPLY commands are fully described in *z/OS MVS System Commands*.

HFS Trace Options

The valid options for use with this trace facility are as follows.

Table 97. Valid Trace Facility Options

Trace Event	Description
ENTRY	All module entries.
EXIT	Module exits with nonzero return codes only.
EXITA	All module exits.
CALL	Equivalent to specifying both ENTRY and EXIT.
RRTN	Entries and exits of recovery routines.
CB	Control block changes.
SPECIAL	Entries and exits of commonly shared functions.
COMP=(*component- list*)	Specifies major functional areas of DFSMSdfp used to manipulate HFSs.
SUBCOMP=(*component- list*)	Specifies minor functional areas within the specified major areas.

The following terms are used to designate the functional areas of HFS management.

Table 98. Functional Areas of HFS Management

COMP	SUBCOMP	
PFS	—	
SSF	ARM	SAC
	CHF	
	DMP	
	ERS	VSM
	RCS	

Notes:

1. You can also specify **ALL** rather than a list for both the **COMP** or **SUBCOMP** keywords.
2. The **SUBCOMP** keyword can be specified without specifying the **COMP** keyword.
3. Any of the options listed in Table 98 except for **ALL**, **COMP**, and **SUBCOMP** can be individually turned off by prefixing them with **NO** (for example, **NOSSF**).

The following command turns off the trace facility:

TRACE CT,OFF,COMP=SYSSMS

HFS CTRACE Writer Options

You can trace without or with a CTRACE writer.

Tracing Without a CTRACE Writer Data Set

When the trace buffer is filled, it wraps and starts overwriting the oldest trace records with the newest trace records. Therefore, only the most recent records are available, and the trace buffer size is the limiting factor. Whether you get an ABEND dump or take a console dump, the dump contains the trace buffers with the most current data, and all the data in the trace buffer can be formatted by the CTRACE formatter.

Tracing With a CTRACE Writer Data Set

As the trace buffer fills up, it is spooled to the CTRACE writer queue. When the buffer is full, a second buffer is obtained, allowing the first buffer to complete being spooled to the CTRACE writer. When the second buffer is full, tracing wraps back to the first buffer and begins overwriting the oldest records.

The second buffer is not contiguous to the first buffer, and therefore is not formatable by IPCS CTRACE formatter. The consequence is that you cannot view the most recent trace records if they are in the second buffer. After the change to SSF trace, the latest records are spooled to the CTRACE writer queue, and therefore the CTRACE writer data set contains all the trace records.

If you take a console dump, the trace buffer is not spooled to the CTRACE writer queue. Therefore, the CTRACE writer data set does not contain the most recent trace records. This is important because the second buffer is not contiguous to the first buffer, and therefore is not formatable by IPCS CTRACE formatter. This means that you cannot view the most recent trace records if they are in the second buffer. After the SSF trace change, the trace formatter will be pointed at the last buffer. This should resolve the problem because the other buffer (if there is one) has already been spooled to the CTRACE writer queue.

Recommendations:

- If you are using a CTRACE writer data set, specify a small buffer size, such as 500 K, which will result in more frequent spooling of the trace records to the CTRACE writer queue.
- If you are not using a CTRACE writer data set, specify a large buffer so that it contains as many trace records as possible to avoid overwriting the trace records you might need to see.

  Identify the JOBNAME or ASID for which you want the large buffer sizes. A global trace applies to the entire system, and this might cause excessive paging activity.
- If you are planning to use a console dump to get your trace records, specify a large buffer size (for example 16 M), and do not specify a CTRACE writer data set. This avoids the second buffer noncontiguous problem when attempting to format the trace records in the dump. Identify the JOBNAME or ASID for which you want the large buffer sizes. A global trace applies to the entire system can cause excessive paging activity.

Producing a CTRACE:

1. Optional (when using CTRACE writer): Create a member in SYS1.PROCLIB to allocate CTRACE writer data sets. Run the following example job to create a member called CTWDASD, which allocates one data set for the CTRACE writer to write to. Change the data set name as appropriate. Use more space if available (suggest CYL (15,5)).

```
//PROCLIB  JOB ,
//      TIME=(,30),MSGCLASS=Z,MSGLEVEL=(1,1)
//*
//* CREATE PROC MEMBER FOR TRACE WRITER
//*
//STEP0 EXEC PGM=IEBGENER,REGION=1024K
//SYSPRINT DD SYSOUT=*
//SYSUT2 DD DSN=SYS1.PROCLIB(CTWDASD),DISP=SHR
//SYSUT1 DD DATA,DLM=EE
//WTRJCL  PROC
//*
//*   PROC FOR WRITER
//*
//CTWDASD  EXEC PGM=ITTTRCWR,TIME=5
//SYSPRINT  DD SYSOUT=A
//* if you have the dasd, allocate more space
//TRCOUT01  DD DSN=IBMUSER.TRACE.WRITER,
//          UNIT=SYSDA,SPACE=(CYL,(5,2)),
//          DISP=(,CATLG)
//*TRCOUT02  DD DSN=IBMUSER.TRACE.WRITER1,
//*          UNIT=SYSDA,SPACE=(CYL,(5,2)),
//*          DISP=(,CATLG)
EE
//SYSIN DD DUMMY
/*
```

2. Optional (when specifying options using a PARMLIB member): Create a PARMLIB member to specify your trace options:

```
//CTISMS01 JOB ,
//   TIME=(,30),MSGCLASS=A,MSGLEVEL=(1,1)
//*
//*  WTR(CTWDASD)
//*
//STEP0 EXEC PGM=IEBGENER,REGION=1024K
//SYSPRINT DD SYSOUT=*
//SYSUT2 DD DSN=SYS1.PARMLIB(CTISMS01),DISP=SHR
//SYSUT1 DD *
```

```
TRACEOPTS ON
     WTR(CTWDASD)
     OPTIONS=('ENTRY,EXIT,EXITA,SPECIAL,CB')
/*
//SYSIN DD DUMMY
/*
```

3. Optional (only to use CTRACE writer): Start the trace writer which allocates the trace data sets (so that the trace buffers are written, rather than wrapped):

   TRACE CT,WTRSTART=CTWDASD

   You get the following:

```
10.43.51           IEE839I ST=(ON,0016K,00016K) AS=ON  BR=OFF EX=ON
MT=(ON,024K)
   ISSUE DISPLAY TRACE CMD FOR SYSTEM AND COMPONENT TRACE STATUS

10.43.53           ICH408I JOB(MSTJCL00) STEP(ASCH    )
LOGON/JOB INITIATION - USER AT TERMINAL          NOT RACF-DEFINED
10.43.53           IRR012I  VERIFICATION FAILED. USER PROFILE NOT FOUND

10.43.54           AHL906I THE OUTPUT BLOCK SIZE OF    23476 WILL BE USED
FOR OUTPUT
   DATA SETS:
   IBMUSER.TRACE.WRITER
10.43.54           ITT110I INITIALIZATION OF CTRACE WRITER CTWDASD
COMPLETE.
```

   Activate trace with a PARMLIB member to specify options:

   TRACE CT,nnnK,COMP=PFS,PARM=CTISMS01

   where CTISMS01 has all the options you intend to trace, and '*nnnK*' is the trace buffer size.

   Or activate trace and specify options with response:

   TRACE CT,500K,COMP=PFS

   Notes:

   a. 500 K is the recommended trace table size if the CTRACE writer is used
   b. Use a larger size if CTRACE writer is not used, for example 4 M
   c. Buffer size range 72 K - 64 M
   d. Smaller buffer size causes frequent spooling to CTRACE writer queue
   e. Larger buffer size causes more paging

   Reply with the trace options:

   Rxx,JOBNAME=OMVS,OPTIONS=(ENTRY,EXIT,EXITA,SPECIAL,CB, COMP=(PFS)),wtr=ctwdasd,end

4. The following message indicates that you entered the command correctly:

```
 10.53.21           ITT038I ALL OF THE TRANSACTIONS REQUESTED VIA THE
 TRACE CT COMMAND WERE SUCCESSFULLY EXECUTED.
 10.53.21           IEE839I ST=(ON,0016K,00016K) AS=ON  BR=OFF EX=ON
 MT=(ON,024K)
    ISSUE DISPLAY TRACE CMD FOR SYSTEM AND COMPONENT TRACE STATUS
```

5. To display what you have just set up, enter the following:

   DISPLAY TRACE,COMP=PFS

The following is displayed:

```
10.51.51           IEE843I 10.51.50  TRACE DISPLAY 077
     SYSTEM STATUS INFORMATION
 ST=(ON,0016K,00016K) AS=ON  BR=OFF EX=ON  MT=(ON,024K)
 COMPONENT     MODE BUFFER HEAD SUBS
 ---------------------------------------------------------------
 PFS           ON   0072K
               ASIDS      *NONE*
               JOBNAMES   *NONE*
               OPTIONS    ENTRY,EXITA
               WRITER     CTWDASD
```

6. Run your testcases.
7. Take a console dump to include the trace buffers that have not yet been written to the trace data sets. If a dump results as part of the recreate, this step is not necessary.

   DUMP COMM=(your dump title)
8. Reply with the OMVS address space name and dataspace name:

   R xx,SDATA=(CSA,SQA,LSQA,PSA,RGN,TRT),CONTR

   R xx,JOBNAME=OMVS,DSPNAME='OMVS'.HFSDSP01,END
9. Terminate or Disconnect Trace:

   Disconnecting the trace retains all of your current trace options. Terminating the trace requires that you respecify the trace options if you restart the trace. Disconnect if you would like to redirect trace data to a new CTRACE writer data set.

   Terminate the trace:

   TRACE CT,OFF,COMP=PFS

   Or

   Disconnect the trace:

   TRACE CT,500K,COMP=PFS (Where 500 K is any size you choose)

   Respond with: **R #,WTR=DISCONNECT,END**
10. Terminate the trace writer:

    TRACE CT,WTRSTOP=CTWDASD,FLUSH

    FLUSH writes the spooled trace data. Note that trace records not already spooled to the CTRACE writer queue are not written to the CTRACE writer data set.
11. Send both the dump and the trace data sets to IBM Level-2 support

    A dump is necessary, either because the recreate took a dump or you took one from the console to get the trace buffers that were not flushed to DASD.
12. To view the trace under IPCS

    To view the writer data set:
    set defaults (option 0) dsname('ibmuser.trace.writer')
    under commands (either option 4 or 6)
    CTRACE comp(syssms) full local

    To view trace buffers in dump, assuming dump is sys.dumpxx:
    set defaults (option 0) dsname('sys1.dumpxx')
    under commands (either option 4 or 6)
    CTRACE comp(syssms) full local

Formatting the HFS Component Trace

The IPCS CTRACE command is used to format and view the information in the component trace table.

The MVS DUMP command can be used to produce an SVCDUMP-type dump containing the trace buffer data. See *z/OS MVS IPCS Commands* for a complete description of this service.

IPCS VERBEXIT SMSXDATA Dump Formatting

IPCS VERBEXIT SMSXDATA supports dump formatting of multiple DFSMScomponents. Control blocks are formatted as hexadecimal data.

An IPCS verb exit can be invoked during an IPCS session or from a batch job. The syntax of the IPCS VERBEXIT SMSXDATA subcommand is as follows.

VERBEXIT	**SMSXDATA**	**'[COMP(***component-list***) | ALL)]** **[GLOBAL | NOGLOBAL]** **[MAP | NOMAP]** **[DETAIL | EXCEPTION]** **[CURRENT | ASID(***asid***) | JOBNAME(***jobname***)]'**

The keywords are described below:

COMP(*component-list* **| ALL)**
You can specify a component or list of components whose control structures are to be formatted. These components are identified by the following terms:

PFS UNIX® System Services File System

SSF System services facility

The default value **ALL** selects all of the above components.

GLOBAL | NOGLOBAL
This keyword requests formatting of global control structures for the selected components.

MAP | NOMAP
This keyword requests formatting of a module map of the load modules.

DETAIL | EXCEPTION
These keywords are defined as follows:

DET Requests formatting of all control structures for the selected components.

EXC Validates all control blocks for the selected components and formats *only* those control blocks that are detected to be invalid for the selected component(s). Validation consists of control block header verification.

CURRENT | ASID(*asid***) | JOBNAME(***jobname***)**
These options are used as follows:

ASID(*asid-number***)**
asid-number is a decimal number that represents the ASID to select for dump access.

CUR Selects each active address space (for example, dispatched on a CPU when the dump was taken) for dump access.

JOB *jobname* represents the jobname and associated ASID to select for dump access.

See *z/OS MVS Dump Output Messages* for explanations of the dump output messages issued by SMSXDATA.

HFS Records Written to SYS1.LOGREC

This section shows an example of a formatted SYS1.LOGREC entry for an HFS-related software failure.

HFS symptom records are used primarily to identify incidents associated with HFS X'0F4' abends. When HFS symptom records occur without an X'0F4' abend, use the symptom strings to search for matching problems in the problem reporting database. If no fix exists, contact the IBM Support Center. Append the keyword prefix RSN to the symptom data for the return code that contains nonzero data in all four bytes and include this HFS reason code keyword in the search.

Figure 19 on page 383 shows a formatted SYS1.LOGREC entry example.

```
TYPE:  SOFTWARE RECORD      REPORT:  SOFTWARE EDIT REPORT           DAY.YEAR
       (SVC 13)                                          REPORT DATE: 280.99
FORMATTED BY: IEAVTFDE  HBB6601                           ERROR DATE: 280.99
                            MODEL:   9672                             HH:MM:SS.TH
                            SERIAL:  038702                   TIME: 08:28:25.38

JOBNAME: OMVS       SYSTEM NAME: AQFT
ERRORID: SEQ=01787  CPU=0040  ASID=000E  TIME=08:28:25.3

SEARCH ARGUMENT ABSTRACT

  PIDS/5695DF185 RIDS/GFUAPFS1#L RIDS/GFUBCNFS AB/S00F4 PRCS/0000009D REGS/0E012
  REGS/0B69C RIDS/GFUAGLRR#R

  SYMPTOM             DESCRIPTION
  -------             -----------
  PIDS/5695DF185      PROGRAM ID: 5695DF185
  RIDS/GFUAPFS1#L     LOAD MODULE NAME: GFUAPFS1
  RIDS/GFUBCNFS       CSECT NAME: GFUBCNFS
  AB/S00F4            SYSTEM ABEND CODE: 00F4
  PRCS/0000009D       ABEND REASON CODE: 0000009D
  REGS/0E012          REGISTER/PSW DIFFERENCE FOR R0E: 012
  REGS/0B69C          REGISTER/PSW DIFFERENCE FOR R0B: 69C
  RIDS/GFUAGLRR#R     RECOVERY ROUTINE CSECT NAME: GFUAGLRR

OTHER SERVICEABILITY INFORMATION

  RECOVERY ROUTINE LABEL:  IGWFRCSD
  DATE ASSEMBLED:          08/20/99
  MODULE LEVEL:            UW62508

SERVICEABILITY INFORMATION NOT PROVIDED BY THE RECOVERY ROUTINE

  SUBFUNCTION

TIME OF ERROR INFORMATION

  PSW: 075C1000 A3B84E60   INSTRUCTION LENGTH: 02   INTERRUPT CODE: 000D
  FAILING INSTRUCTION TEXT: 58F0C028 0A0D41C0 00011E6C

  REGISTERS 0-7
  GR: 5B200123 440F4000 7F2CDEC0 7F100018  0000009D 5B200123 00000013 00000006
  AR: 00000000 00000000 00000000 00000000  00000000 00000000 01010022 00000000
  REGISTERS 8-15
  GR: 7C2F81E0 7C2F8098 23B87A76 A3B847C4  7C2F7A10 7C2F7478 A3B84E4E 0000009D
  AR: 00000000 00000000 00000000 00000000  01010022 00000000 00000000 00000000
```

Figure 19. Formatted SYS1.LOGREC Entry Example (Part 1 of 5).

```
  HOME ASID: 000E    PRIMARY ASID: 000E    SECONDARY ASID: 000E
  PKM: 8040          AX: 0001              EAX: 0000

  RTM WAS ENTERED BECAUSE AN SVC WAS ISSUED IN AN IMPROPER MODE.
  THE ERROR OCCURRED WHILE AN ENABLED RB WAS IN CONTROL.
  NO LOCKS WERE HELD.
  NO SUPER BITS WERE SET.

RECOVERY ENVIRONMENT

  RECOVERY ROUTINE TYPE: FUNCTIONAL RECOVERY ROUTINE (FRR)
  PSW AT ENTRY TO FRR: 070C0000 82D115D8
  FRR PARAMETER AREA ON ENTRY TO FRR:
  +00  C9C7E6C6 C5C6D740 7C2F5028 7C2F5250 00000000 00000000

RECOVERY ROUTINE ACTION

  THE RECOVERY ROUTINE RETRIED TO ADDRESS 23B82D0C.
  THE REQUESTED SVC DUMP WAS NOT TAKEN. ANOTHER DUMP WAS IN PROGRESS.
  NO LOCKS WERE REQUESTED TO BE FREED.
  THE SDWA WAS REQUESTED TO BE FREED BEFORE RETRY.

  THE REGISTER VALUES TO BE USED FOR RETRY:
  REGISTERS 0-7
  GR: 23B87D70 00000000 7C2F5250 7C2F79F8  7C2F7A10 A3BFDDE0 7C2F50A8 7C2F7428
  AR: 00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
  REGISTERS 8-15
  GR: 7C2F81E0 7C2F8098 23B87A76 7C2F5048  A3B82A10 7C2F7478 7C2F54F8 00000000
  AR: 00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
```

Figure 19. Formatted SYS1.LOGREC Entry Example (Part 2 of 5).

```
HEXADECIMAL DUMP

  HEADER
  +000    40831820    00000000    0099280F    08282538    | C.......R......|
  +010    8B038702    96720000                            |..G.O...        |

  JOBNAME
  +000    D6D4E5E2    40404040                            |OMVS            |

  SDWA BASE
  +000    00000C60    C40F4000    00000000    00000000    |...-D. .........|
  +010    00000000    00000000    5B200123    440F4000    |........$..... .|
  +020    7F2CDEC0    7F100018    0000009D    5B200123    |"..{".......$...|
  +030    00000013    00000006    7C2F81E0    7C2F8098    |........@.A\@..Q|
  +040    23B87A76    A3B847C4    7C2F7A10    7C2F7478    |..:.T..D@.:.@...|
  +050    A3B84E4E    0000009D    00000000    00000000    |T.++............|
  +060    00000000    00000000    075C1000    A3B84E60    |.........*..T.+-|
  +070    0002000D    23635001    070C0000    82D115D8    |......&.....BJ.Q|
  +080    0002000D    23635001    23B87D70    00000000    |......&...'.....|
  +090    7C2F5250    7C2F79F8    7C2F7A10    A3BFDDE0    |@..&@.'8@.:.T..\|
  +0A0    7C2F50A8    7C2F7428    7C2F81E0    7C2F8098    |@.
&Y @...@.A\@..Q|
  +0B0    23B87A76    7C2F5048    A3B82A10    7C2F7478    |..:.@.&.T...@...|
  +0C0    7C2F54F8    00000000    00000000    00000000    |@..8............|
  +0D0    00000000    00000000    00000000    00000000    |................|
  +0E0    00000000    00000000    04040001    00000040    |............... |
  +0F0    23B82D0C    00F76394    00000000    048C0000    |.....7.M........|
  +100    00000000    00000000    00000000    00000000    |................|
  +110    00000000    00000000    00000000    00000000    |................|
  +120    000E06FB    C7C6E4C1    D7C6E2F1    C7C6E4C2    |....GFUAPFS1GFUB|
  +130    C3D5C6E2    C7C6E4C1    C7D3D9D9    00F76340    |CNFSGFUAGLRR.7. |
  +140    00000000    00000000    00000000    00000000    |................|
  +150    00000000    00000000    00000000    00000000    |................|
  +160    00000000    00000000    00000000    FFFF0006    |................|
  +170    00F76690    8040000E    0001000E    00000000    |.7... ..........|
  +180    00000000    00000000    00000000    0004A79D    |.............X.|
  +190    00FFA070                                        |....            |
```

Figure 19. Formatted SYS1.LOGREC Entry Example (Part 3 of 5).

```
VARIABLE RECORDING AREA (SDWAVRA)

+000    KEY: C9     LENGTH: 04
+002    00000000                                        |....            |

+006    KEY: CA     LENGTH: 10
+008    B2F59F22    C131AB83    8B038702    96720000    |.5..A..C..G.O...|
+018    KEY: C9     LENGTH: 04
+01A    00005B01                                        |..$.            |

+01E    KEY: CA     LENGTH: 2C
+020    00000000    00000000    00000000    00000000    |................|
+030    00000000    00000000    00000000    00000000    |................|
+040    00000000    00000000    00000000                |............    |

+04C    KEY: C9     LENGTH: 04
+04E    00005B03                                        |..$.            |

+052    KEY: CA     LENGTH: 04
+054    7F2E6000                                        |".-.            |

+058    KEY: C9     LENGTH: 04
+05A    00005B27                                        |..$.            |

+05E    KEY: CA     LENGTH: 04
+060    7C2F7A10                                        |@.:.            |

+064    KEY: C9     LENGTH: 04
+066    00005B28                                        |..$.            |

+06A    KEY: CA     LENGTH: 04
+06C    7C2F81E0                                        |@.A\            |
```

Figure 19. Formatted SYS1.LOGREC Entry Example (Part 4 of 5).

```
SDWA FIRST RECORDABLE EXTENSION (SDWARC1)
+000    C4C6F1F8    F5404040    40404040    40404040    |DF185           |
+010    40404040    40404040    40404040    F0F861F2    |            08/2|
+020    F061F9F9    E4E6F6F2    F5F0F840    0000009D    |0/99UW62508 ....|
+030    C9C7E6C6    D9C3E2C4    F5F6F9F5    02000001    |IGWFRCSD5695....|
+040    00000000    00000000    00000000    82D115D8    |............BJ.Q|
+050    C9C7E6C6    C5C6D740    7C2F5028    7C2F5250    |IGWFEFP @.&.@..&|
+060    00000000    00000000    00000000    00000000    |................|
+070    00000000    0000000E    008ECD08    58F0C028    |.............0{.|
+080    0A0D41C0    00011E6C    00F4AD00    0203607F    |...{...%.4....-"|
+090    440F4000    0000009D    5FB1FE50    0203607F    |.. ........&..-"|
+0A0    72592DC0    8040000E    0001000E    7F95F380    |...{. ......"N3.|
+0B0    00000000    0203607F    00000000    00000000    |......-"........|
+0C0    00000000    00000000    7C1D1FAF    0203607F    |........@.....-"|
+0D0    FF8FF963    7C3020B0    00000000    00000000    |..9.@...........|
+0E0    00000000    00000000    00000000    00000000    |................|
+0F0    01010022    00000000    00000000    00000000    |................|
+100    00000000    00000000    01010022    00000000    |................|
+110    00000000    00000000    00000000    00000000    |................|
+120    00000000    00000000    00000000    00000000    |................|
+130    00000000    00000000    00000000    00000000    |................|
+140    00000000    00000000    00000000    00000000    |................|
+150    00000000    00000000    7F95F380    00000000    |........"N3.....|
+160    00000000    00000000    7B421F00    00000000    |........#.......|
+170    00000000    00000000    00000000    00000000    |................|
+180    00000000    00000000    00000000    00000000    |................|
+190    00000000    00000000    00000000    00000000    |................|
+1A0    00000000    7C302008    00000000    00010000    |....@...........|
+1B0    00000000    00000000    00000000    00000000    |................|
+1C0    C1D8C6E3    40404040                            |AQFT            |

SDWA SECOND RECORDABLE EXTENSION (SDWARC2)
+000    00000000    00000000    00000000    00000000    |................|

SDWA THIRD RECORDABLE EXTENSION (SDWARC3)
+000    00000000    00000000    00000000    00000000    |................|
+010    00000000    00000000    00000000    00000000    |................|

ERRORID
+000    06FB0040    000E0004    A79D                    |... ....X.      |
```

Figure 19. Formatted SYS1.LOGREC Entry Example (Part 5 of 5).

Out-of-Space During Sync Processing

HFS writes a LOGREC record and a console message when it encounters the out-of-space condition during Sync processing. The file system is unusable after that point until it is extended. Any functions that are attempted against the file system, other than confighfs or unmount, when the file system is in this state, returns an error with a return code of ENOSPC (00000085x) and a reason code of 5Bxx0E37x. If the file system is unmounted at this point, any updates that were made to the file system since the last successful Sync are lost, and the file system reverts to the state it had after the last sync. HFS writes the following message to the LOGREC and console:

```
IGW022S HFS Dataset "<dataset name>" Out of Space during
  Sync processing. Error Loc:<location in Sync>
  RC = <return code> RSN = <reason code>

<dataset name> - HFS Dataset name.
<location in Sync> - Location of failure.
  = EXTEND - failure occurred attempting to extend the
       file system during Sync processing.
   <return code> = 00000014x
   <reason code> = 5B27C005x - no space was available
              or no extents were allowed.
         =5B27C007x - an installation exit
           failure occurred.
  =ARPN  -Sync processing ran out of pages in the
       file system at a point where it could not
       Extend the file system.
   <return code> = 00000024x
   <reason code> = 5B0C0101x - internal error 1
           = 5B0C0102x - internal error 2
```

If confighfs is used to successfully extend the file system, for example

```
confighfs -x <amount> <filesys>
```

and the extent was large enough to accommodate the pages required to complete the sync processing, confighfs reinvokes the sync function to complete its update, not loosing the updates made since the last successful sync. The HFS Out- of-Space error flag is reset and all file system functions will work properly again. It will no longer be necessary to unmount and remount the file system to use it further.

If the extend amount is not large enough to provide the space required to complete the sync process, confighfs will issue the following response:

```
Inadequate space added to HFS.  At least another <nn> tracks required.
```

Note: This fix ONLY applies for the case when the IGW022S message indicates an Error Loc: EXTEND value. If it indicates an Error Loc: ARPN value, it will continue to function as it did prior to this update.

HFS Return and Reason Codes

HFS return and reason codes are found:

- ABEND0F4 dump title

  Example:

  ```
  COMPID=DF185,CSECT=GFUAGADS+04E0,DATE=03/26/99,MAINTID= UW57854
   ,ABND=0F4,RC=00000024,RSN=5B9C0101
  ```

- LOGREC

Example:

```
SEARCH ARGUMENT ABSTRACT:
    PIDS/5695DF185 LVLS/DFSMSV1R5 RIDS/GFUAGADS PRCS/00000024
    PRCS/5B0C0101
```

The return code is the right-most byte of the return code word. The reason code is the reason code word. The return and reason codes are always issued as hexadecimal values.

When contacting the IBM Support Center, be prepared to provide:
- Return and reason codes
- Dump title or LOGREC information

HFS Common Return Codes

The following return codes are common to all HFS modules.

Table 99. HFS Common Return Codes

Return Code	Meaning
0 (X'00')	Normal completion.
4 (X'04')	Request completed, but for some reason there is information being returned about the results.
8 (X'08')	Warning. The request was not performed but the results are as expected by the caller. For example: An OPEN request was made, but the data set was already open.
12 (X'0C')	Invalid parameter detected for this request.
16 (X'10')	Request not performed – probable caller error. Possible cause: • The caller might have done things out of order
20 (X'14')	Environmental error has been detected. Possible causes: • Not authorized • Wrong key • Area size incorrect
24 (X'18')	An I/O error has been detected. This may be a recoverable situation.
28 (X'1C')	Media error. This is not recoverable.
32 (X'20')	Data set logical error.
36 (X'24')	Severe system error.

HFS Common Reason Codes

The following list of reason codes is provided to assist in resolving *user correctable* errors. It does not list the reason codes used by HFS to communicate with other IBM products. Contact the IBM Support Center if you receive HFS reason codes that are not in this list.

Reason Code	Description
X'C005'	No DASD space available
X'C007'	Installation exit failed extend

HFS Diagnostic Aids

Reason Code	Description
X'C015'	Data set logical error no secondary allocation

IMF Reason Codes

Reason Code	Description
X'5BxxA02B'	Internal error in IRF code. A logical error was detected while processing records within a directory's index page.
X'5BxxA02E'	Feedback area does not have enough space for IMF to return the found index records.
X'5BxxA030'	Duplicate index record found. Internal error in IRF code. A logical error was detected while processing records within a directory's index page.
X'5BxxA033'	During the search operation of an HFS index, an inconsistent index structure was detected.
X'5BxxA03B'	During the search operation of an HFS index, an index page could not be found.
X'5BxxA074'	Unexpected record type detected while reading the contents of an index page.
X'5BxxA076'	The root page was found to be invalid when that root page of an index was read from disk.
X'5BxxA414'	An index search failed to find a record.
X'5BxxA420'	Attempt to an insert record with key smaller than the previous record.

SSF Reason Codes

Reason Code	Description
X'0E020001'	Error occurred on SETLOCK obtain.
X'0E020002'	Error occurred on SETLOCK release.
X'0E0A0003'	GETMAIN of private storage for DSL failed.
X'18010000'	Private storage not available for ADSR.
X'18FF0000'	Common storage not available for RCGB.
X'1Exx0011'	Insufficient ECSA storage to build the CAT.
X'1Exx001D'	Load module not found.
X'22030001'	Trace not enabled.
X'22070007'	Trace parse options failed.
X'220E0000'	No private storage for TRST available.
X'22F30001'	No active trace options.
X'22FB0000'	No ECSA storage for TRAS available.
X'22FB0003'	No private storage for trace table.
X'25xx0015'	Nonzero return code from SETLOCK obtain.
X'25xx0016'	Nonzero return code from SETLOCK release.
X'25xx0017'	Unable to obtain ASCB address from MVS LOCASCB service.
X'25xx0018'	Nonzero return code from GETMAIN.
X'25xx0019'	Nonzero return code from FREEMAIN.
X'25xx001A'	Nonzero return code from DSPSERV DREFON.
X'25xx001B'	Nonzero return code from DSPSERV FIX.
X'25xx001C'	Nonzero return code from DSPSERV DREFOFF.
X'25xx001D'	Nonzero return code from DSPSERV FREE.
X'25xx0030'	Page fix failed.
X'25xx0031'	Page free failed.
X'25xx0032'	A change key for the storage failed.

Chapter 14. PDSE Diagnostic Aids

The following diagnostic aids are provided for PDSE:

- Two operator commands to help diagnose PDSE errors
- A description of the component trace facility for use with modules that manipulate PDSEs
- IPCS VERBEXIT SMSXDATA dump formatting
- A sample entry in SYS1.LOGREC for a PDSE problem
- Return codes that are common to all PDSE modules
- Reason codes that are common to all PDSE modules

Exception: PDSE does not issue messages. Messages issued when processing a PDSE are from the PDSE access method, or they are PDS compatible messages from the SAM interface modules.

Operator Commands for Analyzing and Repairing PDSE Problems

Although PDSE errors are not common, when errors do occur in the code for PDSEs, the results often extend beyond the job or user who encountered the error. This is a result of the complexity of the PDSE implementation, its use of common control blocks in CSA, and the execution of the majority of its code under the user's tasks. This creates a situation where a failure during the recovery from cancel or force can break or prevent access to a data set.

Two operator commands simplify the determination of what may be broken. In some cases, the repair of the PDSE structures or the selection of which element of the system should be terminated is also determined. The two commands are: the V SMS,PDSE,ANALYSIS command (referred to as the ANALYSIS command) and the V SMS,PDSE,FREELATCH command (referred to as the FREELATCH command).

These commands are commonly used when the processing for one or more PDSEs hangs.

Definitions

LATCH
A serialization object that PDSE processing uses to maintain the integrity of its control block structures. These latches are not GRS latches. Other than the `V SMS,PDSE` commands, there are no mechanisms that allow you to directly determine or modify the status of the PDSE latches.

DIB
A control block that is used by PDSE processing to represent a data set. It is the root from which all OPEN DCB information is stored.

HL1B
A control block that is used by PDSE processing to represent the serialization for the PDSE.

XCM
The subcomponent of PDSE processing that is responsible for communicating between systems using the XCF facilities.

© Copyright IBM Corp. 1979, 2002

HASH TABLE
A structure that can be used to allow for the lookup of various PDSE control blocks, including DIBs and HL1Bs.

LOCK
The internal mechanism that is used to serialize access to the PDSE internal structures.

DIRECTORY LOCK
A lock that is used to serialize the updates to the name and attribute directories of the PDSE. When this lock is held, it is impossible to perform any PDSE functions.

FORMATWRITE LOCK
A lock that is used to serialize the operations that are associated with extending a PDSE and formatting space within the PDSE. When the system holds this lock, it cannot open the PDSE or perform new-member create functions on it, unless it is already open on the same system.

ANALYSIS Command

Issue the ANALYSIS command only when you suspect that one or more users or jobs are having problems accessing a PDSE. This command and the FREELATCH command use a sampling algorithm that interrogates the state of the PDSE every hundredth of a second for the number of retries that the user specifies (the default is 1500 retries, which is approximately 15 seconds). Errors are reported only if the state of the PDSE does not change.

Command Format

V SMS,PDSE,ANALYSIS
[, DSNAME (*dsname*) **[, VOLSER**(*volser*) **]]**
[, Retries(*retries* | *1500*) **]**

Function

The ANALYSIS command detects a number of the most common problems that result in PDSE breakage. The analysis is performed for a single system. If information for more than one system is required, the ROUTE command should be used.

The following errors can be detected:

CONTROL BLOCK BROKEN
The system encountered an invalid control block in running this command.

H1LB Hash table latch held
Prevents the opening and closing of a subset of the PDSEs on this system.

DIB Hash Table latch held
Prevents the opening and closing of a subset of the PDSEs on this system.

HL1B latch held
Prevents locks for this data set from either being gotten or released. It also blocks lock negotiation with other systems.

HL1BPLCH latch held
Prevents locks for this data set from either being gotten or released. It also blocks lock negotiation with other systems.

DIB latch held
Prevents the selected PDSE from being opened or closed on this system.

Contention message to another system is outstanding
Indicates that another system is failing to perform its required role in the sharing of PDSEs across the sysplex. Whenever this error occurs, the most common problem source will be the system that has not responded to the message.

XCM latch held
When the XCM latch is not released, the system is unable to negotiate with other systems. Over time this prevents all other systems from accessing any PDSEs that this system has locked exclusive. The latch is only acquired by tasks that run in the SMXC address space.

LATCH NULLIFIED
The PDSE code has made the latch not available. The code usually does this when it closes the PDSE for the last time on a system. This is an unexpected situation.

LATCH BROKEN
The data structures for a latch have been overlaid.

DIRECTORY OR FORMAT WRITE LOCK HELD OR WAITING
Detects that a job has held or that a job waited for a Directory or Format Write lock for an excessive amount of time.

Operands

dsname When specified, causes the analysis to be performed for a particular PDSE. If the volser is omitted, the data set is found using the default system catalog.

volser Allows you to specify an uncataloged PDSE.

retries Allows you to control the amount of time for which the particular PDSE situation must remain static. The PDSE control blocks are examined every hundredth of a second for the number of retries specified. By default, the data set is examined 1500 times or for approximately 15 seconds before reporting any exceptional conditions. If no exceptional conditions are found, the command returns immediately after the first examination of the control blocks.

Output

This command always produces output in report format. The output has one line preceded with ″++″ for each piece of information returned. It returns information for a data set only when there is an exceptional condition for which recovery actions may be necessary.

When no exceptional conditions are found you get one of the following outputs depending on whether the PDSEs are currently open:

```
IGW031I PDSE ANALYSIS Start of Report
++ no PDSEs connected
 End of Report
```

```
IGW031I PDSE ANALYSIS Start of Report
++ no exceptional data set conditions detected
 End of Report
```

When exceptional conditions are found for one or more PDSEs, the output looks something like this example, where the error conditions vary based on what is found. The output identifies the PDSE by data set name and VSGT (internal representation). The VSGT includes the volser for the PDSE.

```
IGW031I PDSE ANALYZE Start of Report
---------data set name---------------------- ---vsgt----------
SYSPLEX.TEMP.PDSE                             01-XP0201-000104
++ Message to SYSTEM2 pending for 32 seconds
++ Unable to latch Dib:074FC000
   Holder(003A:007D2858)
PDSE ANALYZE End of Report
```

In this case two possible problems were encountered:

1. A negotiation message has been sent to another system and no response has been received.
2. An internal latch (DIB) is held.

See “Recommended Usage” on page 395 for details about the possible outputs and the actions you can choose to take for the various conditions detected.

FREELATCH Command

This command releases any latch that is reported by the ANALYSIS command to be held for too long.

Command Format

V SMS,PDSE,FREELATCH(*latchaddr* , *ASID* ,*tcbaddr*)
[, Retries(*retries* | *1500*)]

Function

The `V SMS,PDSE,FREELATCH` command releases any latch that the ANALYSIS command indicates is held and will not release.

If this command is used to release a latch held by a process that was still running, it could result in the breakage of the PDSE.

The latch is not released unless it is held by the ASID, tcbaddr, indicated in the command. The latch is released only if it is held by the same user for each of the retries.

Operands

latchaddr Address of the latch that is to be released.

ASID ASID of the holder of the latch.

tcbaddr Address of TCB that holds the latch.

When an SRB holds the latch, the address for the TCB actually points to a control block that represents the active SRB. The value will be above the 16 M line.

retries Allows you to control the amount of time for which the particular PDSE situation must remain static. The PDSE control blocks are examined every hundredth of a second. By default the data set is examined 1500 times or for approximately 15 seconds, before the latch will be released.

Output

When the latch is not held by the specified ASID:TCBADDR, you will get the following output:

```
++ FREE LATCH:llllllll  failed . . .
```

When latches are released, you get messages in the following form:

```
++  LATCH:llllllll  RELEASED
```

When you steal the DIB latch, the system might reset the high-reserved FSN to prevent the double use of FSNs. This should be transparent to the user. If the system resets the FSN, the following message is issued:

```
++ Resetting Reserved FSN for DIB:dddddddd
 From hhhhhhhhhhh1 To hhhhhhhhhhh2
 DataSet:PdseDsname
```

Recommended Usage

If you suspect that a PDSE is failing, issue the ANALYSIS command from each system where the PDSE has been reported as failing. Do not issue the FREELATCH command without preceding it with an ANALYSIS command.

These variables indicate the kinds of message information:

nnnnnnnn	Number of seconds a lock or latch is held.
aaaa	ASID (address space identifier) of the latch holder.
tttttttt	TCB address of the latch holder. If an SRB holds the latch, the address is greater than *01000000*.
llllllll	Address of the latch that is held.
ssssssss	System to which a message is directed.
hhhhhhhh	Address of the HL1B that is the control block that contains the locking information for a PDSE.
dddddddd	Address of the DIB that contains the data structure information for a PDSE.
jjjjjjjj	Job name for the apparent hold of a latch or lock.
stststst	Started task name for the apparent holder of a latch or lock. The SMXC is the locking address space for the PDSE. You should never force or cancel the SMXC.
nummsgs	Number of outstanding messages from this system.
n holders	Number of holders for a lock.
n writers	Number of writers for a lock.

You should see one of the following results:

++ No PDSEs connected

++ No exceptional data set conditions detected

This indicates that the problem is not one of the class of problems that this tool is designed to detect or fix. Obtain an SVC dump and contact your IBM service representative.

++ Message to *ssssssss* **pending for** *nnnnnnnn* **seconds**

This system has been waiting for another system to respond to it. Try running the ANALYSIS command on the other system. The problem is probably on the other system. This command is sometimes displayed with the DIB latch message. When the other system fails to respond during an OPEN, both messages will appear, but the problem is still on the other system and no attempt should be made to issue the FREELATCH command to free the latch on this system.

++ Unable to latch DIB:ddddddd
LATCH:*llllllll* **Holder (***aaaa: ttttttt* **)**
Holding JOB:*jjjjjjjj* **| Holding Started Task (***ststst* **|**
Unable to determine job or started task name for holder

This latch is held only during OPEN and should not be held for very long, unless the system is waiting for a response from another system. The latch blocks new OPENs and CLOSEs for the data set in question. You should first attempt to cancel and force the job or user that has the latch. If that does not work, issue a FREELATCH command for this data set. If that does not work, either re-IPL or stop using the data set.

++ Unable to latch HL1B:*hhhhhhhh*
LATCH: Holder (aaaa: ttttttt)
Holding JOB:*jjjjjjjj* **| Holding Started Task (***ststst* **|**
Unable to determine job or started task name for holder

This latch is held only during an attempt to serialize updates to the PDSE. The latch should never be held for very long because it blocks most updates and reads of the data set in question. You should first attempt to cancel and force the job or user that has the latch. If that does not work, issue a FREELATCH command for this data set. If that does not work, either re-IPL or stop using the data set.

++ Unable to latch HL1BPLCH:*hhhhhhhh*
LATCH:*llllllll* **Holder (***aaaa: ttttttt* **)**
Holding JOB:*jjjjjjjj* **| Holding Started Task (***ststst* **|**
Unable to determine job or started task name for holder

This latch is held only during an attempt to send messages to another system. The latch should never be held for very long, unless a message is being sent to another system. The latch blocks most updates and reads of the data set in question. If any messages to another system are outstanding, you should perform analysis and repair on that system. If no messages to other systems are outstanding, you should first attempt to cancel and force the job or user that has the latch. If that does not work, issue a FREELATCH command for this data set. If that does not work, either re-IPL or stop using the data set.

++ Unable to latch DIB:*llllllll* **Holders changing**
++ Unable to latch HL1B:*llllllll* **Holders changing**
++ Unable to latch HL1BPLCH:*llllllll* **Holders changing**

This is an informational message. It indicates that although the latch was held every time you tried to access the structures, the holder was dynamic. This indicates high activity on the data set but does not indicate any error. The occurrence of any of these messages is unlikely. They are here only for information.

++ Unable to obtain latch for DIB | HL1B HashTable:*llllllll* **Holder(***aaaa:ttttttt* **)**

This message indicates that a global latch that is used to locate the structures for the PDSE is frozen. You should attempt to cancel the job or user that is associated with the ASID. If the system still holds the latch, force the initiator. If the system still holds the latch, you should attempt the FREELATCH command for this latch.

++ Unable to obtain latch for DIB | HL1B HashTable:*llllllll* **Holders changing**

This message indicates that a global latch used to locate the structures for the PDSE is very busy, preventing successful completion of the analysis. Reissue the FREELATCH command until the analysis is successful.

++ Unable to latch XCM Holder(*aaaa:ttttttt* **)**

The SMXC address space obtains this latch only when XCF messages are sent to another system or when messages are received from another

system. This indicates that a global latch that is not associated with a particular data set is frozen. You should attempt to issue the FREELATCH command for this latch.

++ Unable to latch XCM Holders changing

This message indicates that a global latch used to serialize messaging between systems is very busy, preventing successful completion of some analysis. You should reissue the FREELATCH command until the analysis is successful.

++ Latch DIB | HL1B | HL1BPLCH Nullified
++ Latch DIB | HL1B | HL1BPLCH Broken

This message indicates that the specified latch either is broken or has been nullified. Both cases indicate that the data structures for the PDSE are compromised. You should attempt to have all users unallocate the PDSE; this might clean up the data structures.

++ Lock GLOBAL | LOCAL DIRECTORY | FORMATWRITE SHARED | EXCLUSIVE
Held for at least *nnnnnnnn* **seconds**
HL1B:*hhhhhhhh* **HOLDER(***aaaa:tttttttt***)**
Holding JOB:*jjjjjjjj* **| Holding Started Task:***stststst* **|**
Unable to determine job or started task name for holder

This message indicates that the system has held a directory or format write lock for a long time. The indicated amount of time begins with the first ANALYSIS command that detected the waiting lock request.

If the lock is GLOBAL, it has been granted for the sysplex. If there is contention on this resource, you might want to cancel the job that holds the lock.

If the lock is held locally, this system is waiting for another system to release the lock, so that it can be granted. If there is no message to another system, you might need to cancel this job. If there is a message to another system, issue the ANALYSIS command on that system and resolve the problem on that system.

If the command cannot detect the lock holder, you might be able to fix the problem by canceling all users of the data set.

++ *n holders* **Additional holders of DIRECTORY |FORMAT WRITE Lock**
++ *n writers* **waiting for DIRECTORY | FORMAT WRITE LOCK Shared | Exclusive**

These messages give an indication of the amount of activity against a data set when a lock has been held for an excessive amount of time.

++ *nummsgs* **unresponded messages to** *ssssssss*

This is a summary of messages sent to other systems that have been outstanding for too long. If you do not find any other errors, run the ANALYSIS command on the other system.

++ Broken PDSE structure —
Bad HL3B | LRE encountered

Internal structures of the PDSE are broken. Attempt to quiesce the use of the PDSE. This frees all existing PDSE structures and might resolve the PDSE problem.

PDSE Component Trace

DFSMSdfp provides a trace service for use with all functions that manipulate PDSEs. A special trace table is created in each address space in which PDSEs are used and tracing is selected.

You can use SYS1.PARMLIB member CTncccxx to specify trace options for the TRACE CT command and to turn off tracing. CTncccxx must exist at IPL for tracing to be available on the system. The default is CTISMS00. For more information on using CTncccxx to specify tracing options, see ″CTncccxx″ in *z/OS MVS Initialization and Tuning Reference*.

Trace also allows external writers (WTR) to capture trace buffers. Capturing trace buffers increases diagnostic capability and decreases the possibility of losing trace data.

Complete syntax and usage information for TRACE can be found in *z/OS MVS System Commands*.

TRACE Command for PDSE Component Trace

The syntax of the TRACE command for tracing the PDSE component follows.

```
TRACE      ,WTRSTART=membername [WRAP|NOWRAP ] ] ]
           CT [,WTRSTOP=jobname ] ]
           [CT [,ON ] { {COMP=SYSSMS } [,SUB=(sub[sub],...)]
           [,PARM=CTISMSxx] } ]
           [CT [,nnnK ]{                                    } ]
           [CT [,nnnnM] {                                    } ]
           [CT [,OFF] {                                      } ]
```

The default **SYSSMS** trace table size is 72 KB.

PARM=CTISMS*xx* identifies the SYS1.PARMLIB member that contains the tracing options.

The *nnn***K** and *nnnn***M** keywords can be used to specify trace table sizes of 16–999 KB or 1–2047 MB, respectively.

You are then prompted to specify the trace options to be in effect. You must respond with the REPLY command, using the following syntax.

```
REPLY      id
           [,JOBNAME=(jobnamelist)]
           ASID[,=(asidlist.)]
           [,OPTIONS=(namename...)]
           [,WTR=procname|DISCONNECT]
           [,END]
```

Exception:

- The *id* value is the identification number, 0-99, specified on the prompting message.
- You may list up to 16 jobs or ASIDs to be used as filters in tracing.

- The REPLY may be continued on multiple lines; the END must be given as the final parameter to identify the end of the REPLY.

The TRACE and REPLY commands are fully described in *z/OS MVS System Commands*.

PDSE Trace Options

The valid options for use with this trace facility follow.

Table 100. Valid Trace Facility Options

Trace Event	Description
ENTRY	All module entries.
EXIT	Module exits with nonzero return codes only.
EXITA	All module exits.
CALL	Equivalent to specifying both ENTRY and EXIT.
RRTN	Entries and exits of recovery routines.
CB	Control block changes.
SPECIAL	Entries and exits of commonly shared functions.
COMP=(*component- list*)	Specifies major functional areas of DFSMSdfp used to manipulate PDSEs.
SUBCOMP=(*component- list*)	Specifies minor functional areas within the specified major areas.

The following terms are used to designate the functional areas of PDSE management.

Table 101. Functional Areas of PDSE Management

COMP	SUBCOMP	
BMF	BDS	LMC
	CDS	SRS
	IOC	VBM
CDM	ADC	
	LAC	
	RAC	
	DSC	
CLM	BTR	HLS
	BRL	LIC
	CLS	LRS
	ENS	SLS
	GLS	XCM
IMF	—	
SSF	ARM	SAC
	CHF	
	DMP	
	ERS	VSM
	RCS	

Tip:

- You may also specify **ALL** rather than a list for both the **COMP** or **SUBCOMP** keywords.

- The **SUBCOMP** keyword can be specified without specifying the **COMP** keyword.
- Any of the options listed above except for **ALL**, **COMP**, and **SUBCOMP** can be individually turned off by prefixing them with **NO** (for example, **NOSSF**, **NOIMF**).

The following command will turn off the trace facility:

TRACE CT,OFF,COMP=SYSSMS

PDSE CTRACE Writer Options

You can trace without or with a CTRACE writer.

Tracing Without a CTRACE Writer Data Set

When the trace buffer is filled, it wraps and starts overwriting the oldest trace records with the newest trace records. Therefore, only the most recent records are available, and the trace buffer size is the limiting factor. Whether you get an ABEND dump, or take a console dump, the dump contains the trace buffers with the most current data, and all the data in the trace buffer can be formatted by the CTRACE formatter.

Tracing With a CTRACE Writer Data Set

As the trace buffer fills up, it is spooled to the CTRACE writer queue. When the buffer is full, a second buffer is obtained, allowing the first buffer to finish being spooled to the CTRACE writer. When the second buffer is full, tracing wraps back to the first buffer and begins overwriting the oldest records.

The second buffer is not contiguous to the first buffer, and therefore is not formatable by IPCS CTRACE formatter. The consequence is that you cannot view the most recent trace records if they are in the second buffer. After the change to SSF trace, the latest records are spooled to the CTRACE writer queue, therefore the CTRACE writer data set contains all the trace records.

If you take a console dump, the trace buffer is not spooled to the CTRACE writer queue. Therefore, the CTRACE writer data set does not contain the most recent trace records. This is important because the second buffer is not contiguous to the first buffer, and therefore is not formatable by IPCS CTRACE formatter. This means that you cannot view the most recent trace records if they are in the second buffer. After the SSF trace change, the trace formatter will be pointed at the last buffer. This should resolve the problem, since the other buffer (if there is one), has already been spooled to the CTRACE writer queue.

Recommendations:

- If you are using a CTRACE writer data set, specify a small buffer size, such as 500 K, which results in more frequent spooling of the trace records to the CTRACE writer queue.
- If you are not using a CTRACE writer data set, specify a large buffer so that it contains as many trace records as possible to avoid overwriting the trace records you might need to see.

  Identify the JOBNAME or ASID for which you want the large buffer sizes. A global trace applies to the entire system and can cause excessive paging activity.
- If you are planning to use a console dump to get your trace records, specify a large buffer size (for example 16 M), and do not specify a CTRACE writer data set. This avoids the second buffer noncontiguous problem when attempting to format the trace records in the dump. Identify the JOBNAME or ASID for which you want the large buffer sizes. A global trace applies to the entire system and can cause excessive paging activity.

Producing a CTRACE:

1. Optional (when using CTRACE writer): Create a member in SYS1.PROCLIB to allocate CTRACE writer data sets. Run the following example job to create a member called CTWDASD, which allocates one data set for the CTRACE writer to write to. Change the data set name as appropriate. Use more space if available (suggest CYL (15,5)).

```
//PROCLIB  JOB ,
//      TIME=(,30),MSGCLASS=Z,MSGLEVEL=(1,1)
//*
//* CREATE PROC MEMBER FOR TRACE WRITER
//*
//STEP0 EXEC PGM=IEBGENER,REGION=1024K
//SYSPRINT DD SYSOUT=*
//SYSUT2 DD DSN=SYS1.PROCLIB(CTWDASD),DISP=SHR
//SYSUT1 DD DATA,DLM=EE
//WTRJCL  PROC
//*
//*   PROC FOR WRITER
//*
//CTWDASD  EXEC PGM=ITTTRCWR,TIME=5
//SYSPRINT  DD SYSOUT=A
//* if you have the dasd, allocate more space
//TRCOUT01  DD DSN=IBMUSER.TRACE.WRITER,
//          UNIT=SYSDA,SPACE=(CYL,(5,2)),
//          DISP=(,CATLG)
//*TRCOUT02  DD DSN=IBMUSER.TRACE.WRITER1,
//*          UNIT=SYSDA,SPACE=(CYL,(5,2)),
//*          DISP=(,CATLG)
EE
//SYSIN DD DUMMY
/*
```

2. Optional (when specifying options by means of PARMLIB member): Create a PARMLIB member to specify your trace options:

```
//CTISMS01 JOB ,
//   TIME=(,30),MSGCLASS=A,MSGLEVEL=(1,1)
//*
//*  WTR(CTWDASD)
//*
//STEP0 EXEC PGM=IEBGENER,REGION=1024K
//SYSPRINT DD SYSOUT=*
//SYSUT2 DD DSN=SYS1.PARMLIB(CTISMS01),DISP=SHR
//SYSUT1 DD *
TRACEOPTS ON
     WTR(CTWDASD)
     OPTIONS=('ENTRY,EXIT,EXITA,SPECIAL,CB')
/*
//SYSIN DD DUMMY
/*
```

3. Optional (only to use CTRACE writer): Start the trace writer which allocates the trace data sets (so that the trace buffers are written, rather than wrapped):

   TRACE CT,WTRSTART=CTWDASD

   You get the following response:

```
10.43.51           IEE839I ST=(ON,0016K,00016K) AS=ON  BR=OFF EX=ON
MT=(ON,024K)
   ISSUE DISPLAY TRACE CMD FOR SYSTEM AND COMPONENT TRACE STATUS

10.43.53           ICH408I JOB(MSTJCL00) STEP(ASCH    )
LOGON/JOB INITIATION - USER AT TERMINAL          NOT RACF-DEFINED
10.43.53           IRR012I  VERIFICATION FAILED. USER PROFILE NOT FOUND

10.43.54           AHL906I THE OUTPUT BLOCK SIZE OF    23476 WILL BE USED
FOR OUTPUT
```

```
   DATA SETS:
   IBMUSER.TRACE.WRITER
10.43.54          ITT110I INITIALIZATION OF CTRACE WRITER CTWDASD
COMPLETE.
```

Activate trace with a PARMLIB member to specify options:

TRACE CT,nnnK,COMP=PFS,PARM=CTISMS01

where CTISMS01 has all the options you intend to trace, and '*nnnK*' is the trace buffer size.

Or

Activate trace and specify options with response:

TRACE CT,500K,COMP=PFS

Tips:

- 500 K is the recommended trace table size if the CTRACE writer is used.
- Use a larger size if CTRACE writer is not used; for example, 4 M.
- Buffer size range is from 72 K to 64 M.
- Smaller buffer size causes frequent spooling to CTRACE writer queue.
- Larger buffer size causes more paging.

Reply with the trace options:

Rxx,JOBNAME=OMVS,OPTIONS=
(ENTRY,EXIT,EXITA,SPECIAL,CB,COMP=(PFS)),wtr=ctwdasd,end

4. The following message indicates that you entered the command correctly:

```
10.53.21          ITT038I ALL OF THE TRANSACTIONS REQUESTED VIA THE
TRACE CT COMMAND WERE SUCCESSFULLY EXECUTED.
10.53.21          IEE839I ST=(ON,0016K,00016K) AS=ON  BR=OFF EX=ON
MT=(ON,024K)
   ISSUE DISPLAY TRACE CMD FOR SYSTEM AND COMPONENT TRACE STATUS
```

5. To display what you have just setup:

   DISPLAY TRACE,COMP=PFS

   The following is displayed:

```
10.51.51          IEE843I 10.51.50  TRACE DISPLAY 077
     SYSTEM STATUS INFORMATION
 ST=(ON,0016K,00016K) AS=ON  BR=OFF EX=ON  MT=(ON,024K)
 COMPONENT     MODE BUFFER HEAD SUBS
 ------------------------------------------------------------
 PFS           ON   0072K
               ASIDS      *NONE*
               JOBNAMES   *NONE*
               OPTIONS    ENTRY,EXITA
               WRITER     CTWDASD
```

6. Run your test cases.
7. Take a console dump to include the trace buffers that have not yet been written to the trace data sets. If a dump results as part of the re-create, this step is not necessary.

   DUMP COMM=(your dump title)

8. Reply with the OMVS address space name and dataspace name:

R xx,SDATA=(CSA,SQA,LSQA,PSA,RGN,TRT),CONTR

R xx,JOBNAME=OMVS,DSPNAME=’OMVS’.HFSDSP01,END

9. Terminate or Disconnect Trace

   Disconnecting the trace retains all of your current trace options. Terminating the trace requires that you respecify the trace options if you restart the trace. Disconnect if you would like to redirect trace data to a new CTRACE writer data set.

   Terminate the trace:

   TRACE CT,OFF,COMP=PFS

   Or

   Disconnect the trace:

   TRACE CT,500K,COMP=PFS (Where 500 K is any size you choose)

   Respond with: **R #,WTR=DISCONNECT,END**

10. Terminate the trace writer:

    TRACE CT,WTRSTOP=CTWDASD,FLUSH

    FLUSH writes the spooled trace data. Note that trace records not already spooled to the CTRACE writer queue are not written to the CTRACE writer data set.

11. Send both the dump and the trace data sets to IBM Level-2 support

    A dump is necessary, either because the recreate took a dump or you took one from the console to get the trace buffers that were not flushed to DASD.

12. To view the trace under IPCS

```
To view the writer data set:
 set defaults (option 0) dsname('ibmuser.trace.writer')
 under commands (either option 4 or 6)
 CTRACE comp(syssms) full local

To view trace buffers in dump, assuming dump is sys.dumpxx:
set defaults (option 0) dsname('sys1.dumpxx')
under commands (either option 4 or 6)
CTRACE comp(syssms) full local
```

Formatting the PDSE Component Trace

The IPCS CTRACE command is used to format and view the information in the component trace table.

The MVS DUMP command can be used to produce an SVCDUMP-type dump containing the trace buffer data. See *z/OS MVS IPCS Commands* for a complete description of this service.

IPCS VERBEXIT SMSXDATA Dump Formatting

IPCS VERBEXIT SMSXDATA supports dump formatting of multiple DFSMS components. Control blocks are formatted as hexadecimal data.

An IPCS verb exit can be invoked during an IPCS session or from a batch job. The syntax of the IPCS VERBEXIT SMSXDATA subcommand is as follows.

VERBEXIT	**SMSXDATA**	**'[COMP**(*component-list*) | **ALL**)] [**GLOBAL** | **NOGLOBAL**] [**MAP** | **NOMAP**] [**DSPACE** | **NODSPACE**] [**DETAIL** | **EXCEPTION**] [**CURRENT** | **ASID**(*asid*) | **JOBNAME**(*jobname*)]**'**

The keywords are described below:

COMP(*component-list* **| ALL)**
You can specify a component or list of components whose control structures are to be formatted. These components are identified by the following terms:

AMA Access method adapter

BMF Buffer management facility

CDM Common data manager

CLM Common lock manager

DFM Distributed FileManager

IMF Index management facility

SSF System services facility

The default value **ALL** will select all of the above components.

GLOBAL | NOGLOBAL
This keyword requests formatting of global control structures for the selected components.

MAP | NOMAP
This keyword requests formatting of a module map of the load modules.

DETAIL | EXCEPTION
These keywords are defined as follows:

DET Requests formatting of all control structures for the selected components.

EXC Validates all control blocks for the selected components and formats *only* those control blocks that are detected to be invalid for the selected components. Validation consists of control block header verification.

DSPACE | NODSPACE

This keyword requests formatting of all BMF data space pages. It does not format individual control blocks.

CURRENT | ASID(*asid***) | JOBNAME(***jobname***)**
These options are used as follows:

ASID(*asid-number***)**
asid-number is a decimal number that represents the ASID to select for dump access.

CUR Selects each active address space (for example, dispatched on a CPU when the dump was taken) for dump access.

JOB *jobname* represents the jobname and associated ASID to select for dump access.

See *z/OS MVS Dump Output Messages* for explanations of the dump output messages issued by SMSXDATA.

PDSE Records Written to SYS1.LOGREC

This section shows an example of a formatted SYS1.LOGREC entry for a PDSE-related software failure.

PDSE symptom records are used primarily to identify incidents associated with PDSE X'0F4' abends. When PDSE symptom records occur without an X'0F4' abend, use the symptom strings to search for matching problems in the problem reporting database. If no fix exists, contact the IBM Support Center. Append the keyword prefix RSN to the symptom data for the return code that contains nonzero data in all four bytes and include this PDSE reason code keyword in the search.

Figure 20 shows a formatted SYS1.LOGREC entry example.

```
TYPE:  SYMPTOM RECORD    REPORT: SOFTWARE EDIT REPORT   DAY YEAR
                        REPORT DATE: 266  92
SCP:   VS 2 REL 3           ERROR DATE:    217  92
                  MODEL:  9021           HH MM SS.TH
                  SERIAL: 120546   TIME:   09:45:18.96
SEARCH ARGUMENT ABSTRACT:
  PIDS/566528464 LVLS/DFPV3R30 RIDS/IGWBSDMI PRCS/0000000C
  PRCS/2009001B
SYSTEM ENVIRONMENT:
  CPU MODEL:  9021               DATE:  217  92
  CPU SERIAL: 120546             TIME:  09:45:18.96
  SYSTEM:     M040                BCP:   MVS
  RELEASE LEVEL OF SERVICE ROUTINE:      JBB4422
  SYSTEM DATA AT ARCHITECTURE LEVEL:     10
  COMPONENT DATA AT ARCHITECTURE LEVEL:  10
  SYSTEM DATA:  00000000 00000000    |........|
COMPONENT INFORMATION:
  COMPONENT ID:             566528464
  COMPONENT RELEASE LEVEL:  330
  PID NUMBER:       5665XA3
  PID RELEASE LEVEL:        V3R30
  SERVICE RELEASE LEVEL:      \ {
  DESCRIPTION OF FUNCTION:  PDSE DSPACE SUPPORT
  PROBLEM ID:               IGW00002
  SUBSYSTEM ID:             SMS
PRIMARY SYMPTOM STRING:
  PIDS/566528464  LVLS/DFPV3R30   RIDS/IGWBSDMI   PRCS/0000000C
  PRCS/2009001B   #
SYMPTOM           SYMPTOM DATA EXPLANATION
-------           ----------  -----------
PIDS/566528464    566528464
LVLS/DFPV3R30     DFPV3R30
RIDS/IGWBSDMI     IGWBSDMI
PRCS/0000000C     0000000C
PRCS/2009001B     2009001B
THE SYMPTOM RECORD DOES NOT CONTAIN A SECONDARY SYMPTOM STRING.
FREE FORMAT COMPONENT INFORMATION:
  KEY VALUE IS NULL. FREE FORMAT IS TERMINATED.
HEX DUMP OF RECORD:
 HEADER
```

Figure 20. Formatted SYS1.LOGREC Entry (Part 1 of 2).

```
+000 4C831800 00000000 0092217F 09451896 ³<C.......K."...O│
+010 A2120546 90210000                   ³S.......        ³
 SYMPTOM RECORD
+000 E2D9F9F0 F2F1F1F2 F0F5F4F6 FFFFCA5B ³SR9021120546...$³
+010 A61799CF 09608101 40404040 40404040 ³W.R..-A.        ³
+020 4040D4F0 F4F04040 4040F5F7 F5F2D1C2 ³  M040    5752JB³
+030 C2F4F4F2 F2400080 00000000 00000000 ³B4422 ..........³
+040 F1F00030 00640070 005100D4 00000000 ³10.........M....³
+050 005C0125 00000000 00000000 00000000 ³.*..............³
+060 00000000 00000000 00000000 00000000 ³................³
+070 E2D9F2F1 F1F0F5F6 F6F5F2F8 F4F6F400 ³SR2110566528464.³
+080 F3F3F040 4000E051 C0000005 F5F6F6F5 ³330  .\.{...5665³
+090 E7C1F340 E5F3D9F3 F0404040 D7C4E2C5 ³XA3 V3R30   PDSE³
+0A0 40C4E2D7 C1C3C540 E2E4D7D7 D6D9E340 ³ DSPACE SUPPORT ³
+0B0 40404040 40404040 40404040 00000000 ³            ....³
+0C0 00000000 C9C7E6F0 F0F0F0F2 E2D4E240 ³....IGW00002SMS ³
+0D0 40404040 D7C9C4E2 61F5F6F6 F5F2F8F4 ³    PIDS/5665284³
+0E0 F6F44040 D3E5D3E2 61C4C6D7 E5F3D9F3 ³64  LVLS/DFPV3R3³
+0F0 F0404040 D9C9C4E2 61C9C7E6 C2E2C4D4 ³0   RIDS/IGWBSDM³
+100 C9404040 D7D9C3E2 61F0F0F0 F0F0F0F0 ³I   PRCS/0000000³
+110 C3404040 D7D9C3E2 61F2F0F0 F9F0F0F1 ³C   PRCS/2009001³
+120 C2404040 7B000020 11000000 08000000 ³B   #...........³
+130 00000000 00000020 06000000 44040C00 ³................³
+140 008125BA 68000000 00000000 0000FD41 ³.A..............³
+150 58000000 00070E00 00000000 00070C00 ³................³
+160 00810231 0E075C30 0082AEF4 34000000 ³.A....*..B.4....³
+170 00000000 00070E00 00000000 00000000 ³................³
+180 00                                  ³.              ³
```

Figure 20. Formatted SYS1.LOGREC Entry (Part 2 of 2).

PDSE Return and Reason Codes

PDSE return and reason codes are found:

- ABEND0F4 dump title

  Example:

  ```
  COMPID=DF115,CSECT=IGWISRCH+05C8,DATE=04/12/99,MAINTID= UW58377
   ,ABND=0F4,RC=00000024,RSN=145AA033
  ```

- LOGREC

  Example:

  ```
  SEARCH ARGUMENT ABSTRACT:
      PIDS/566528464 LVLS/DFSMSV1R5 RIDS/IGWBSDMI PRCS/000000 0C
      PRCS/2009001B
  ```

The return code is the right-most byte of the return code word.

The reason code is the reason code word.

The return and reason codes are always issued as hexadecimal values.

When contacting the IBM Support Center, be prepared to provide:

- Return and reason codes
- Dump title or LOGREC information.

  Prerequisite: PDSE information APAR II04594 lists the documentation required before creating a PDSE APAR.

PDSE Common Return Codes

The following return codes are common to all PDSE modules.

Table 102. PDSE Common Return Codes

Return Code	Meaning
0 (X'00')	Normal completion
4 (X'04')	Request completed, but for some reason there is information being returned about the results.
8 (X'08')	Warning. The request was not performed but the results are as expected by the caller. For example: An OPEN request was made, but the data set was already open.
12 (X'0C')	Invalid parameter detected for this request.
16 (X'10')	Request not performed – probable caller error. Possible cause: • The caller might have done things out of order
20 (X'14')	Environmental error has been detected. Possible causes: • Not authorized • Wrong key • Area size incorrect
24 (X'18')	An I/O error has been detected. This may be a recoverable situation.
28 (X'1C')	Media error. This is not recoverable.
32 (X'20')	Data set logical error.
36 (X'24')	Severe system error.

PDSE Common Reason Codes

The following list of reason codes is provided to assist in resolving *user correctable* errors. It does not list the reason codes that are used by PDSE to communicate with other IBM products. Contact the IBM Support Center when you receive PDSE reason codes that are not in this list.

Reason Code	Description
X'C001'	The BMF data space is out of storage.
X'C002'	Exhausted ECSA storage.
X'C003'	Exhausted user private storage.
X'C004'	Exhausted DREFED autodata storage.
X'C005'	No DASD space available.
X'C006'	Unable to convert VSE VTOC.
X'C007'	Installation exit failed extend.
X'C008'	I/O error on a PDSE directory.
X'C009'	I/O error on a PDSE member.
X'C00A'	DADSM I/O error on a VTOC.
X'C00E'	Media error on a PDSE directory.
X'C00F'	Media error on a PDSE member.
X'C012'	Data set logical error on a PDSE directory.
X'C013'	Data set logical error on a PDSE member.

PDSE Diagnostic Aids

Reason Code	Description
X'C015'	Data set logical error no secondary allocation.
X'C016'	A request is already in progress.
X'C020'	A MVCL length was exceeded.

BMF Reason Codes

Reason Code	Description
X'05xx001A'	There are still users connected to this data set.
X'05xx0031'	DADSM obtain format-1 DSCB failed.
X'05xx0032'	DADSM obtain format-3 DSCB failed.
X'05xx0046'	Invalid DSCB extents.
X'05xx0049'	SETLOCK obtain failed for the local lock.
X'05xx0051'	The data set is still connected during data set destroy.
X'07xx0050'	LSCB processed by recovery.
X'09xx0013'	SETLOCK obtain failed.
X'09xx0015'	SETLOCK release failed.

CDM Reason Codes

Reason Code	Description
X'01xx5781'	Caller is not authorized to specify the bypass RACF option.
X'01xx5784'	Data set connect failed because of share options.
X'01xx5788'	The name specified on input does not exist in the attribute directory.
X'01xx579F'	Nonzero return code from RACHECK.
X'01xx57A7'	Nonzero return code from free block.
X'01xx57BA'	The name specified on input does not exist in the name directory.
X'01xx57C6'	The input name to be destroyed was the primary name of the file, but the caller did not specify the destroy file option.
X'01xx57CE'	The input name already exists in the directory, and the caller did no specify the destroy alias option.
X'01xx57CF'	The input name already exists as the primary name.
X'01xx57D1'	Caller is trying to create a PDSE that already exists.
X'01xx57D2'	The input connect intent specifies an invalid value.
X'01xx57D3'	Unable to delete the data set because it's currently connected on this system.
X'01xx57D4'	Unable to delete the data set because it's currently connected on some other system in the Parallel Sysplex®.
X'01xx57D8'	The BMF connect for T0COPY failed. The data set was extended while the connection was being made.
X'01xx57D9'	An HFS data set is being connected with the HFS option being specified.
X'01xx57DA'	An HFS data set is being connected for output, but the data set is already connected for output.
X'01xx5B5D'	Caller must wait until no output connections exist before the request to connect for dump will be satisfied.
X'01xx5B5E'	Caller must wait until a prior dump connection finishes before the request to connect for output will be satisfied.
X'01xx5B5F'	Caller must wait until a prior restore connection finishes before any request to connect will be satisfied.
X'01xx5B60'	The dump connection failed because some output connection occurred before the dump finished.
X'01xx5B61'	The call to disconnect the data set with the partial release option failed because other connections still exist.
X'01xx5B62'	Caller attempted to insert a duplicate alias without specifying replace.

Reason Code	Description
X'01xx5B64'	The data set is not an HFS data set.
X'0Dxx5608'	Caller buffer too big, greater than 180 pages.
X'0Dxx560A'	Out of space on DASD for restore.
X'0Dxx5B5F'	Caller must wait until a prior restore connection finishes before any request to connect will be satisfied.
X'0Fxx5141'	Connect intent requested is not valid.
X'0Fxx5142'	Current use of file and file's share options do not allow connection.
X'0Fxx5144'	File does not exist.
X'0Fxx5145'	Cannot be destroyed because connections still exist.
X'0Fxx5148'	Last disconnect and file was pending destroy.
X'0Fxx5149'	Not all members in this list could be connected.
X'0Fxx514A'	File could not be connected as requested.
X'0Fxx514E'	File not destroyed.
X'0Fxx514F'	Member is already connected.
X'0Fxx5151'	Request contained duplicate names for the same file.
X'0Fxx5159'	Not enough information for restore.
X'0Fxx515E'	This file already has its checkpoint bit set.
X'0Fxx5160'	Two or more file IDs in the list represent the same file.
X'0Fxx5161'	At least one file in the list already has its checkpoint bit set.
X'0Fxx5162'	There is an enqueue held by another system for this PDSE.
X'0Fxx5163'	The file protect flag is on and the connect intent is for output.
X'0Fxx5164'	Member lock held within this system.
X'0Fxx5165'	Member lock held on another system.
X'0Fxx5166'	Create name request for member failed.
X'0Fxx5169'	Maximum number of members in PDSE exceeded.
X'0Fxx516A'	Duplicate MLT is not allowed in PDSE.
X'0Fxx516C'	Unable to create a unique MLT.
X'0Fxx5B62'	Caller attempted to insert a duplicate alias without specifying replace.
X'1Cxx52D1'	No space available in private to create the DWA.
X'1Cxx52DB'	I/O error. Error found when trying to read or write data pages.
X'1Cxx52E6'	Input user buffer is too small.
X'1Cxx52E7'	BMF extend failed. Data set extend failed.
X'1Cxx52F9'	Update record is longer than the original record.
X'1Cxx52FA'	Update record is shorter than the original and the pad option is not specified.
X'1Cxx5302'	Segmented record is not completed.
X'1Cxx5304'	The record is longer than specified record size.
X'1Cxx5B64'	The data set is not an HFS data set.
X'1Dxx4E22'	Authorization check failed.
X'1Dxx4E29'	The input data supplied caused a storage protection error.
X'1Dxx4E2A'	Input parameters are key 5 but the caller's PSW key is not 5.
X'1Dxx4E2B'	Could not allocate private space for FPL and associated data copies.
X'1Dxx4E35'	Unable to get private storage for the RPB.
X'1Dxx4E36'	Unable to get private storage for the SOR pool.
X'1Dxx4E37'	Unable to get private storage for the small FPL pool.
X'1Dxx4E38'	Unable to get private storage for the medium FPL pool.
X'1Dxx4E39'	Unable to get private storage for the large FPL pool.
X'1Dxx4E4A'	An attempt to delete a VSM storage pool failed.
X'1Dxx4E4B'	An attempt to free a block failed.
X'1Dxx4E4C'	An attempt to free storage failed.
X'1Dxx4E4F'	Space for copying the user data to key 5 could not be obtained.
X'1Dxx4E50'	There was no SOR pool and a create attempt failed.
X'1Dxx4E51'	An attempt to free the RPB failed.
X'1Dxx4E52'	The SOR pool could not be deleted.
X'1Dxx4E53'	A SOR block could not be returned to the pool.

PDSE Diagnostic Aids

Reason Code	Description
X'1Dxx4E54'	Unable to free a SOR.
X'1Dxx4E5A'	Unable to create an AIB pool in private storage.
X'1Dxx4E5B'	Unable to get private space for an AIB.
X'1Dxx4E5C'	Unable to delete the AIB pool.
X'1Dxx4E5D'	Unable to free an AIB pool element.
X'1Dxx4E5E'	Could not free an AIB.
X'1Dxx4E6D'	A request for local storage failed.
X'1Dxx4E6E'	A request to delete a pool failed.
X'5Cxx5F51'	Destroy is not allowed for root directory.
X'5Cxx5F53'	Directory is not empty.
X'5Cxx5F54'	Connections still exist for the directory.
X'5Cxx5F56'	Invalid name length specified.
X'5Cxx5F59'	File type is neither data nor directory.
X'5Cxx5F5A'	Output intent is specified for mounted read-only file system.
X'5Cxx5F5D'	The file is already connected.
X'5Cxx5F63'	The name looked up is a subdirectory and is already connected.
X'5Cxx5F69'	Old name not found in source directory.
X'5Cxx5F6B'	Subdirectory pointed to by the new name is already connected.
X'5Cxx5F6E'	Subdirectory pointed to by new name is not empty.
X'5Cxx5F6F'	No rename to be done since source and target are the same, and old and new names are the same.
X'5Cxx5F75'	The new name points to a subdirectory and the old name does not.
X'5Cxx5F76'	The old name points to a subdirectory and the new name does not.

CLM Reason Codes

Reason Code	Description
X'12xx8A15'	A create data set request failed because the same data set is currently open on other systems.
X'12xx8F1C'	A create data set request failed because the same data set is currently open on other systems.
X'12xx8F1D'	A create data set request failed because the same data set is currently open on other systems.
X'13xx0706'	SETLOCK call failed to obtain the local lock.
X'13xx0707'	SETLOCK call failed to release the local lock.
X'13xx0727'	A resume was issued for an invalid or nonexistent TCB.
X'13xx0728'	A schedule SRB was issued for an invalid ASCB.
X'13xx3643'	Could not get MVS local lock.
X'13xx3644'	MVS resume failed.
X'58xx1123'	Installation defined maximum XCF group limit has been reached.
X'58xx1124'	Installation defined maximum XCF member limit has been reached.

FAMS Reason Codes

Reason Code	Description
X'28xxnnnn'	nnnn is the specific FAMS reason code for this instance. All reason codes issued by FAMS are translated into a IGW01 message that is issued to the task. Convert the last two bytes (X'nnnn') of the reason code from hexadecimal to a decimal value and refer to that IGW01nnns message in *z/OS MVS System Messages, Vol 8 (IEF-IGD)*, *z/OS MVS System Messages, Vol 9 (IGF-IWM)*, and *z/OS MVS System Messages, Vol 10 (IXC-IZP)*.

For messages with the prefix IGW01, the type codes indicate the severity of the detected error, as follows:

I Information: Return code 0
W Warning: Return code 4
E Error: Return code 8
T Ending: Return code 12
S Severe: Return code 16

For example: Take the last 2 bytes of reason code RSN28260096. Converting X'0096' to decimal gives you 150. Prefixing it with IGW01 results in message IGW01150T.

IMF Reason Codes

Reason Code	Description
X'14xx9C41'	No storage available.
X'14xx9C43'	The IMIB was not initialized because a request for ECSA storage was not successful.
X'14xx9C45'	The IUB was not initialized because a request for private storage was not successful.
X'14xx9C46'	The backup IMIB was not initialized because a request for ECSA storage was not successful.
X'14xx9C47'	The ICE pool was not initialized because a request for private storage was not successful.
X'14xx9C4C'	FREEMAIN request not successful for the IUB.
X'14xx9C4E'	FREEPOOL request not successful.
X'14xx9C67'	The IUB pool was not initialized because a request for private storage was not successful.
X'14xx9C68'	The ICAR pool was not initialized because a request for private storage was not successful.
X'14xx9C69'	The BPL pool was not initialized because a request for private storage was not successful.
X'14xx9C6C'	The IASB was not initialized because a request for private storage was not successful.
X'14xx9C74'	The IRE pool was not initialized because a request for private storage was not successful.
X'14xxA033'	An index page is damaged. Some requests failed.
X'14xxA035'	VSM is out of ICE pool space.
X'14xxA03A'	Could not obtain private storage for the ICAR.
X'14xxA03E'	Unexpected ABEND during IMF processing.
X'14xxA076'	Content of a root page is damaged.
X'14xxA08C'	Out of DASD storage for index pages.
X'14xxA08D'	I/O error during read.
X'14xxA08E'	I/O error during write.

SSF Reason Codes

Reason Code	Description
X'0E020001'	Error occurred on SETLOCK obtain.
X'0E020002'	Error occurred on SETLOCK release.
X'0E0A0003'	GETMAIN of private storage for DSL failed.
X'18010000'	Private storage not available for ADSR.
X'18FF0000'	Common storage not available for RCGB.
X'1A940200'	PDSE support not available.

PDSE Diagnostic Aids

Reason Code	Description
X'1Exx0011'	Insufficient ECSA storage to build the CAT.
X'1Exx001D'	Load module not found.
X'22030001'	Trace not enabled.
X'22070007'	Trace parse options failed.
X'220E0000'	No private storage for TRST available.
X'22F30001'	No active trace options.
X'22FB0000'	No ECSA storage for TRAS available.
X'22FB0003'	No private storage for trace table.
X'25xx0015'	Nonzero return code from SETLOCK obtain.
X'25xx0016'	Nonzero return code from SETLOCK release.
X'25xx0017'	Unable to obtain ASCB address from MVS LOCASCB service.
X'25xx0018'	Nonzero return code from GETMAIN.
X'25xx0019'	Nonzero return code from FREEMAIN.
X'25xx001A'	Nonzero return code from DSPSERV DREFON.
X'25xx001B'	Nonzero return code from DSPSERV FIX.
X'25xx001C'	Nonzero return code from DSPSERV DREFOFF.
X'25xx001D'	Nonzero return code from DSPSERV FREE.
X'25xx0030'	Page fix failed.
X'25xx0031'	Page free failed.
X'25xx0032'	A Change Key for the storage failed.

Chapter 15. SAM Diagnostic Aids

The following major diagnostic aids are provided for the sequential access method (SAM):

- Extended format I/O routine diagnostic information
- The format of the SAM interrupt control block (ICB) that is used for chained scheduling (Tape and unit record).
- A procedure for retrieving additional information in SYNAD routine (DASD). For tracing of access method code that is called during OPEN, CLOSE, and EOV, see “Using Error Records for Debugging” on page 248

  For fields that SAM adds to the IOB, see “IOB” on page 335.

If a message or abend code is related to the failure, refer also to:

- *z/OS MVS System Messages, Vol 1 (ABA-AOM)*
- *z/OS MVS System Messages, Vol 2 (ARC-ASA)*
- *z/OS MVS System Messages, Vol 3 (ASB-BPX)*
- *z/OS MVS System Messages, Vol 4 (CBD-DMO)*
- *z/OS MVS System Messages, Vol 5 (EDG-GFS)*
- *z/OS MVS System Messages, Vol 6 (GOS-IEA)*
- *z/OS MVS System Messages, Vol 7 (IEB-IEE)*
- *z/OS MVS System Messages, Vol 8 (IEF-IGD)*
- *z/OS MVS System Messages, Vol 9 (IGF-IWM)*
- *z/OS MVS System Messages, Vol 10 (IXC-IZP)*
- *z/OS MVS System Codes*

Extended Format I/O Routine Diagnostic Information

The Extended Format I/O Routine processes data residing in sequential extended-format data sets. It is called by the System Data Mover (SDM) whenever it determines that the data set being processed is an extended format data set. The extended format data set may be in the compressed format or non-compressed format.

SDM passes the Extended Format I/O Routine an internal parameter list describing the function requested. On return from the Extended Format I/O Routine, the return and reason codes are returned in register 15 and register 0, respectively. They are also stored in the parameter list at offsets X'18' and X'1C'. If the return and reason codes from the Extended Format I/O Routine indicate that the called system service encountered an error, the error return and reason codes from the called service are stored in the diagnostic fields in the parameter list at offsets X'20' and X'24'.

When the Extended Format I/O Routine returns an error to SDM, the error return and reason codes from the Extended Format I/O Routine may be displayed in system message ADR910E. If so, contact the IBM Support Center for diagnostic assistance.

Table 103 lists the Extended Format I/O Routine return and reason codes.

Table 103. Extended Format I/O Routine

Return Code	Reason Code	Description
0 (X'00')	0 (X'00')	The requested operation was performed successfully.

© Copyright IBM Corp. 1979, 2002

Table 103. Extended Format I/O Routine (continued)

Return Code	Reason Code	Description
4 (X'04')	Reason codes listed below.	The requested operation failed.
	2 (X'02')	The internal parameter list passed is not valid for one of the following reasons: • The required eye-catcher is not valid • The required Extended Format I/O Routine token is zero
	3 (X'03')	The Extended Format I/O Routine token passed in the internal parameter list is not valid. The token eye-catcher is not valid.
	4 (X'04')	The function code in the internal parameter list is not valid for one of the following reasons: • A zero function code is specified • A read function is requested while open for output • A write function is requested while open for input
	5 (X'05')	An invalid request caused the open function to fail. One of the following required fields was zero in the internal parameter list: • SDM token • number of stripes • volume count • volume information
	6 (X'06')	At least one of the volumes of the target extended format data set does not meet the hardware requirements for an extended format data set. Possible user error. Ensure that the hardware requirements for an extended format data set are met when using a DFSMSdss™ or DFSMShsm function.
	7 (X'07')	The Extended Format I/O Routine called SDM to perform a service. SDM encountered a severe error which prevented it from completing any I/O requests successfully.
	8 (X'08')	The Extended Format I/O Routine called CVAFDIR to perform a service. CVAFDIR encountered an error. During close processing, an attempt to update information in the data set label for the target data set via CVAFDIR failed.
	9 (X'09')	During a write function, it was determined that no blocks of length specified as the block size would fit on the target track.
	10 (X'0A')	During the last write (buffer count=0), an error was detected attempting to write the remaining data to the target data set.
	11 (X'0B')	During a write function, it was determined that the length of one of the records found in the source data set exceeded the block size of the target data set.

Table 103. Extended Format I/O Routine (continued)

Return Code	Reason Code	Description
	13 (X'0D')	During a write function, it was determined that the target data set is not allocated with enough space to hold all of the source tracks. This is an out-of-extent situation.
	14 (X'0E')	A write function failed because it determined that a zero block size had been provided for the target data set.
	15 (X'0F')	The Extended Format I/O Routine called Compression Services to perform a service. Compression Services encountered an error. Although Compression Services is required, it is not able to process the data set.
	16 (X'10')	The Extended Format I/O Routine called DASDCALC to perform a service. DASDCALC encountered an error. DASDCALC is required to determine the physical block size of the target data set.
	17 (X'11')	During open, DASDCALC services was called to determine the physical block size of the target data set. Although it returned successfully, it could not derive a block size.
	18 (X'12')	A write function failed because it determined that no compression dictionary token had been provided for the compressed format data set.
	19 (X'13')	A write function failed because it determined that no RECFM had been provided for the compressed format data set.
	20 (X'14')	An invalid request caused the open function to fail. Reblocking was requested, but either RECFM, LRECL, or block size was zero.
	21 (X'15')	An invalid request caused the open function to fail. Reblocking was requested with RECFM of U or V(B)S. Possible user error. Ensure that the reblock option is not specified in a DFSMSdss function if the data set RECFM is U or V(B)S. Reblocking is not supported for these RECFMs.
	22 (X'16')	An invalid request caused the open function to fail. Source data set is compressed format and target is non-extended format. However, track-packing was not requested.
	23 (X'17')	An invalid request caused the open function to fail. Data set is multivolume but the number of stripes is greater than one. A multivolume extended format data set cannot consist of more than one stripe.
	24 (X'18')	A user record was decompressed to a length other than the length that was expected.
	25 (X'19')	An error was returned from the DECOMPRESS call to Compression Services.

Table 103. Extended Format I/O Routine (continued)

Return Code	Reason Code	Description
	26 (X'1A')	An error was encountered in a source track. During reconstruction of a record, the next segment was neither a middle or last segment.
	27 (X'1B')	A compressed record was found in the data set, but the dictionary token is not valid.
	28 (X'1C')	A user block was not found in the track.

Interrupt Control Block (ICB)

The ICB, used by SAM, contains channel programs and status information when using chained scheduling. Chained scheduling is used only on tape and unit record devices. The ICB is pointed to by the DCBIOBA field of the DCB.

Table 104. SAM Interrupt Control Block (ICB)—Description and Format

Offset Dec(Hex)	Bytes and Bit Pattern	Field Name	Description
0 (X'00')	*	ICB	SAM interrupt control block.
0 (X'00')	1	ICBNFLG1	Flag byte.
	X'80'	ICBPRTOV	PRTOV has occurred.
	X'40'	ICBWRITE	Write operation in process.
	X'20'	ICBREAD	Read operation in process.
	X'10'	ICBUPDAT	Block is to be updated.
	X'08'	ICBBKSPC	ICB being used for backspace, CNTRL, NOTE, or POINT.
	X'04'	ICBSPAN	This record is a spanned record.
	.... ..x.		Reserved.
	X'01'	ICBFIRST	This is first ICB on chain.
1 (X'01')	3	ICBNICBB	Address of next ICB on chain.
4 (X'04')	4	ICBNECB	Event control block.
8 (X'08')	1	ICBFLAG1	Flag byte.
	X'80'	ICBDATCH	Data chaining used in channel program.
	X'40'	ICBCMDCH	Command chaining used in channel program.
	X'20'	ICBERRTN	Error routine is in control.
	X'10'	ICBRPSTN	Device is to be repositioned.
	X'08'	ICBCYCCK	Cyclic redundancy check (tape only).
	X'04'	ICBIOERR	I/O error has occurred.
	X'02'	ICBUNREL	This I/O request is unrelated (nonsequential).
	X'01'	ICBRSTRT	Restart address in ICB to be used.
9 (X'09')	1	ICBFLAG2	Flag byte.
	X'80'	ICBHALT	Halt I/O has been issued by SVC purge routine.
	X'40'	ICBSENSE	Issue sense command after device end occurs.
	X'20'	ICBPURGE	ICB has been purged; allow I/O to quiesce.
	X'02'	ICBSTATO	Device end status has been ORed with channel end status; graphics device.
	X'01'	ICBPNCH	Turned on by QSAM when error recovery is to be provided for the 2540 card punch.
10 (X'0A')	1	ICBSENS0	First sense byte.
11 (X'0B')	1	ICBSENS1	Second sense byte.

Table 104. SAM Interrupt Control Block (ICB)—Description and Format (continued)

Offset Dec(Hex)	Bytes and Bit Pattern	Field Name	Description
12 (X'0C')	1	ICBECBCC	Completion code for this I/O request.
13 (X'0D')	3	ICBECBPB	Address of ECB to be posted on I/O completion.
16 (X'10')	1	ICBFLAG3	Error routine flag byte.
17 (X'11')	7	ICBCSW	Low order seven bytes of CSW at channel end.
24 (X'18')	4	ICBSTART	Address of channel program.
24 (X'18')	1	ICBSIOCC	Bits 2 and 3 = condition code from SIO.
28 (X'1C')	2	ICBINCAM	Value used to increment block count on tape.
28 (X'1C')	1	ICBCRDCC	Optical reader; data check error count.
29 (X'1D')	1	ICBCRILC	Optical reader; incorrect length error.
30 (X'1E')	1	ICBINDIC	Special count indicators.
	X'80'	ICBVOLFL	End of volume with read/write error.
	.xxx xxxx		Reserved.
31 (X'1F')	1		Reserved.

Additional SYNAD Information (BSAM and QSAM)

For a PDSE, extended format data set, or USS file, additional information might be provided to the SYNAD routine. The following procedure enables you to retrieve this information:

- Obtain the address of the status indicators.
  - For QSAM, the address is located in bits 8-31 of register 0.
  - For BSAM, the DECB address is located in bits 8-31 of register 0. The address of the status indicators is located at offset 16 (X'10') in the DECB.
- At offset 28 (X'1C') is a one byte field which may contain a diagnostic code stored by SAM to further describe the error.

A nonzero value in this field usually indicates that the problem was due to a non-hardware related error.

For certain user errors, it may be possible for you to easily determine the problem and make the appropriate modifications to your program.

Table 105 is a list of diagnostic codes.

Table 105. SAM Diagnostic Codes

Diagnostic Code	Description
1 (X'01')	Bad parameter list passed to asynchronous completion routine by SMS system service (PDSE).
2 (X'02')	Program check encountered in asynchronous completion routine (PDSE).
3 (X'03')	Error return code returned from SMS system service when processing PDSEs.
4 (X'04')	Invalid token encountered by POINT.[1]
5 (X'05')	Unsuccessful attempt to position to a member via POINT (PDSE).

Table 105. SAM Diagnostic Codes (continued)

Diagnostic Code	Description
6 (X'06')	Data set open for output, but the token passed to POINT was not for the current member (PDSE).[1]
7 (X'07')	Outstanding I/O was in progress when POINT issued.[1]
8 (X'08')	Token passed to POINT would have caused POINT to position beyond the end of the data set.[1]
9 (X'09')	Unable to get appropriate locks during POINT processing (PDSE).
10 (X'0A')	Attempt to POINT to a nonexistent member (PDSE).[1]
11 (X'0B')	A padding error encountered (extended format data set).
12 (X'0C')	Error return code returned from a system service when processing extended format data sets.
13 (X'0D')	An I/O error was encountered in a shadow IOB (compressed format data set).
14 (X'0E')	A READ/WRITE issued against a compressed format data set whose DCB specifies NOTE/POINT (MACRF=P) was for a block whose user RBN value exceeds the 3-byte token limit of 16 777 215 (compressed format data set).
15 (X'0F')	An error was encountered while processing a USS file.

Note:

1. Probable user error.

If you plan to contact IBM service, the following information may be useful in diagnosing the problem.

- One byte diagnostic code at offset 28 (X'1C') from status indicators.
- Two words at offset 56 (X'38') and 60 (X'3C') from status indicators.

Chapter 16. Storage Management Subsystem Diagnostic Aids

The following major diagnostic aids are provided for the Storage Management Subsystem (SMS):

- A procedure to isolate the source of an SMS message-indicated failure to an area either within SMS or elsewhere, in a *non-DFSMS* component
- A description of the SMS IPCS dump formatting verb exit (SMSDATA) and examples of SMSDATA usage
- Information about the use of the SMS trace facility
- A description of the trace facility for use with modules that manipulate USS files and IPCS formatting with SMSXDATA
- A sample entry in SYS1.LOGREC for an SMS problem
- A sample SMS SUMDUMP showing the intermodule communications area (IMC), the recovery communications area (ERRCA), and a partial SMS procedure trace table (ERPTT)
- SMS subcomponent return code and reason code tables

Storage Management Subsystem Message-Indicated Failure Source Identification

Use this section to isolate the probable source of a message-indicated failure to an area within SMS or to one of the services external to DFSMS that SMS uses. The following procedure may be necessary if SMS issues a failure-related message that identifies a *non-DFSMS* component.

1. If a failure-related SMS message identifies a module whose name begins with **IGD**, the source of the failure is probably within SMS.
2. If a failure-related SMS message identifies a module whose name begins with anything other than **IGD**, use the message to module table in the appendix of the following sources to determine the failure-related component:
   - *z/OS MVS System Messages, Vol 1 (ABA-AOM)*
   - *z/OS MVS System Messages, Vol 2 (ARC-ASA)*
   - *z/OS MVS System Messages, Vol 3 (ASB-BPX)*
   - *z/OS MVS System Messages, Vol 4 (CBD-DMO)*
   - *z/OS MVS System Messages, Vol 5 (EDG-GFS)*
   - *z/OS MVS System Messages, Vol 6 (GOS-IEA)*
   - *z/OS MVS System Messages, Vol 7 (IEB-IEE)*
   - *z/OS MVS System Messages, Vol 8 (IEF-IGD)*
   - *z/OS MVS System Messages, Vol 9 (IGF-IWM)*
   - *z/OS MVS System Messages, Vol 10 (IXC-IZP)*

© Copyright IBM Corp. 1979, 2002

Storage Management Subsystem IPCS Dump Formatter (SMSDATA)

The SMS IPCS verb exit (SMSDATA) is intended for the use of diagnosticians who are working with the IBM Support Center to resolve an SMS-related problem.

This section describes the following aspects of the SMS IPCS verb exit (SMSDATA):
- The SMSDATA functions
- The control blocks that SMSDATA can format
- SMSDATA's invocation syntax and options
- SMSDATA formatting examples.

SMSDATA Verb Exit Functions

The SMSDATA verb exit performs the following functions:
- Validates control block chains in the SMS address space.
- Formats the control blocks in the SMS address space.
- Formats the trace table in the SMS address space.
- Formats the control blocks associated with the SMS automatic data areas.

Validating Control Block Chains

When SMSDATA examines a chain of control blocks, it validates the chain pointers. That is, when SMSDATA finds a zero chain pointer, and a nonzero pointer is expected, it issues an informational message. SMSDATA only validates the control block chains necessary to process a caller's request; it does not automatically validate unreferenced control blocks.

Formatting Control Blocks

The SMSDATA verb formats the control blocks and uses IPCS to write the formatted information to the output data set. Table 106 describes the control blocks in a dump that can be formatted.

Table 106. Control Blocks Formatted by SMSDATA

Control Block	Macro Description
ACOHD	Object Table Header.
AGD	Aggregate Group Definition
BCD	Base Configuration Definition.
DCD	Data Class Definition.
DRD	Drive Definition.
ERMAP	SMS Module Map.
ERTRE	SMS Trace Table Data Header.
ERTRN	SMS Trace Table Entry.
ERTRT	SMS Trace Table Control Area.
ICMRT	Intersystem Communications Root.
ICMDS	Communications Data Set Header.
LBD	Library Definition.
MCD	Management Class Definition.
OPSCR	Configuration Root.
OPSRT	Operational Services Root.
ERPTT	SMS Procedure Flow Trace Table.
ERRCA	SMS Error Recovery Communications Area.
SCD	Storage Class Definition.
SGD	Storage Group Definition.
SSIAT	SMS Address Space Vector Table.
IMC	SMS Intermodule Communications Area.
SSISP	SMS Automatic Data Area Block Header.
SSISS	SMS Storage Segment Header.
SSIVT	SMS Vector Table.
VLD	Volume Definition.

Invoking the Storage Management Subsystem IPCS Dump Formatter (SMSDATA)

Invoke the SMS IPCS Dump Formatter (SMSDATA) from the IPCS environment. The syntax follows.

SMSDATA	**[MAP | NOMAP]** **[CONFIG{(ACTIVE | ALL)}]** **[FORMAT(SUMMARY | ALL | ACS | DATACLAS |** **MGMTCLAS | STORCLAS | STORGRP | VOLUMES |** **BCD | ERTRE | ERTRT | ICMDS | ICMRT | OPSCR |** **OPSRT | SSIAT | SSIVT)]** **[TRACE]** **[ASID(***asid number***)]** **[SELECT{(ALL | ACSINT | ACSPRO | CATG | CDSC |** **CONFA | CONFC | CONFR | CONFS | DCF |** **DISP | DPN | ERR | IDAX |** **MODULE | MSG | OPCMD | RCD | SCHEDP | SCHEDS |** **SMSSJF | SMSSSI | TVR |** **VOLREF | VTOCA | VTOCC |** **VTOCD | VTOCL | VTOCR)}]** **[START(***ddd,hh:mm:ss***)]** **[STOP(***ddd,hh:mm:ss***)]** **[TADDR(***ERTRE address***)]** **[STORAGE(***IMC address,IMC asid***,[DATA |** **FLO | IMC])]**

If you do not select any options, SMSDATA defaults to **MAP** and **FORMAT(SUMMARY)**. If you select only the **NOMAP** parameter, SMSDATA defaults to **FORMAT(SUMMARY)**.

The SMSDATA verb keywords and abbreviations are described in the following list:

MAP | NOMAP
Specifies whether the SMS module map (ERMAP) is to be formatted.

CONFIG(ACTIVE | ALL)

CON(ACTIVE | ALL)
Requests formatting of the SMS control blocks related to an SMS configuration in the storage management address space (SMAS). You may select either the *active configuration* or request *all* configurations in the SMAS. The following control blocks are related to an SMS configuration:

ACOHD
AGD
AGN
BCD
DCD
DRD
LBD
MCD
OPSCR
SCD
SGD
VLD

FORMAT

FOR

Requests formatting of SMS control blocks, either individually or in groups. The two available groups consist of control blocks *related to the active configuration*, and control blocks that are *not related to the configuration*, as cited in the following two lists.

Note: If you specify both the **FORMAT** and **CONFIG** options, SMSDATA only formats the control block(s) once, under the "Configuration" section of the SMSDATA output.

SUMMARY

Requests formatting of the control blocks that are *not related to the configuration*.

ALL

Requests formatting of *all* control blocks, whether or not they are configuration related.

Select the control blocks *related to the active configuration* by specifying the applicable parameter(s) from the following list.

ACS	Requests formatting of the ACS routine object table headers (ACOHD).
DATACLAS	Requests formatting of the data classes in the configuration (DCD).
MGMTCLAS	Requests formatting of the management classes in the configuration (MCD).
STORCLAS	Requests formatting of the storage classes in the configuration (SCD).
STORGRP	Requests formatting of the storage groups in the configuration (SGD).
VOLUMES	Requests formatting of the volume definitions in the configuration (VLD).

Select the control block(s) that are *not related to the configuration* by specifying the applicable parameter(s) from the following list:

ERTRE
ERTRT
ICMDS (COMMDS)
ICMRT
OPSRT
SSIAT
SSIVT

TRACE

TR

Requests formatting of the SMS trace table.

Note: If **TRACE** is specified or implied and no other limiting parameters are supplied (for example, **SELECT, ASID**), then all of the trace records are formatted.

ASID(*asid number***)**
Restricts formatting of trace records to those in which the home ASID is equal to the *asid number* you select (specified in decimal). Specifying **ASID** implies the **TRACE** option.

SELECT

SEL
Restricts formatting of trace records to those associated with one or more trace events selected from the following list. The valid trace events are the same as those that can be specified in SYS1.PARMLIB or on the SETSMS operator command. Specifying **SELECT** implies the **TRACE** option.

```
ACSINT      CONFS       MSG         VOLREF
ACSPRO      DCF         OPCMD       VTOCA
ALL         DISP        RCD         VTOCC
CATG        DPN         SCHEDP      VTOCD
CDSC        DSTACK      SCHEDS      VTOCL
CONFA       ERR         SMSSJF      VTOCR
CONFC       IDAX        SMSSSI
CONFR       MODULE      TVR
```

These options are described under “Trace Event Options” on page 431.

START(*DDD,HH:MM:SS***)**

ST(*DDD,HH:MM:SS***)**
Restricts formatting of trace records to those created at or after the specified time. Specifying **START** implies the **TRACE** option.

STOP(*DDD,HH:MM:SS***)**

SP(*DDD,HH:MM:SS***)**
Restricts formatting of trace records to those created at or before the specified time. Specifying **STOP** implies the **TRACE** option.

TADDR(*ERTRE address***)**

TD(*ERTRE address***)**
Allows the user to specify explicitly the location of the SMS trace table. If **TADDR** is not specified, SMSDATA locates the trace table using the globally owned SMS control blocks. Specifying **TADDR** implies the **TRACE** option.

STORAGE(*IMC address,IMC ASID***,DATA | FLO | IMC)**

STO(*IMC address,IMC ASID***[,DATA | FLO | IMC])**
The **STORAGE** option requests formatting of the control blocks that make up the automatic data areas used by the SMS modules. You must supply the address of the root control block (IMC) as the first positional parameter. If you do not supply the IMC ASID parameter, SMSDATA defaults to the SMAS ASID. The **DATA**, **FLO**, and **IMC** parameters are optional and nonpositional; you can select more than one. If you do not specify any of these parameters, SMS does not produce any output.

DATA Requests formatting of the automatic data areas that were active when the dump was taken.

FLO Requests formatting of the procedure flow trace table attached to the IMC. This table contains the names of the modules in the order they received control, and any applicable return and reason codes. Because the table is wraparound, it is most useful for determining module flow immediately before an error.

IMC Requests formatting of all of the control blocks associated with the IMC whose address was specified. These control blocks are:

ERPTT SMS procedure flow trace table.

ERRCA SMS error recovery communications area.

IMC SMS intermodule communications area.

SSISP SMS automatic data area block header (only the fixed-length portion).

SSISS SMS storage segment header (only the fixed-length portion).

When formatted using the **IMC** parameter, these control blocks receive no special formatting considerations as they do when formatted using the **DATA** or **FLO** options.

SMSDATA Formatting Examples

This section provides examples of invoking the SMS IPCS verb exit under ISPF and the output it produces. For information about how to use the verb exit syntax and options, see *z/OS ISPF Reference Summary*.

Formatting the SSI Vector Table

You may use the **FORMAT** option to format the SSIVT. The following command would display the output panel shown in Figure 21.

```
VERBEXIT SMSDATA 'FORMAT(SSIVT)'
```

```
IPCS OUTPUT STREAM ----------------------------------------- LINE 0 COLS 1 78
COMMAND ===>                                                SCROLL ===> CSR
  ****************************** TOP OF DATA ***********************************
  --------------------- Storage Management Subsystem Data ----------------------
  ----------------- Storage Management Subsystem Control Blocks ----------------
    IGDSSIVT:  01DCBF00
       +0000  ID: IGDSSIVT       LEN: 0100    VERSION: 0000
       +000C  SMS_MODULE_MAP: 01DC4000       SMS_ASID: 0000000E
       +0014  SMS_ASCB: 00F7FE00
       +0018  LINKAGE_INDEX: 00000B00        CONSOLE_ID: 00000000
       +0020  SYSTEM_ID: SYSTEM1       INTERVAL: 0000000F
       +002C  ACDS: SYS1.STAGE2.ACDS
       +0058  COMMDS: SYS1.STAGE2.COMMDS
       +0084  PARMLIB_SUFFIX: 00       OPERATOR_COMMAND_FLAGS: 9000
       +0088  PROMPT_OPTIONS: 20000000       TRACE_TABLE_PTR: 01DCBE80
       +0090  AUTHORIZATION_OPTIONS: 00000000
       +0094  SMS_ADDRESS_SPACE_STATUS: C0000000   RESTARTS: 00000000
       +009C  ADDRESS_SPACE_RESTART: 808FE438
       +00A0  SMS_AS_VECTOR_TABLE: 01E00FA8
       +00A4  IGDSSIP2_SPACE_SWITCH: 00000B01
       +00A8  IGDSSIP2_NO_SPACE_SWITCH: 00000B00   SEQ_PREFIX: IGD#
       +00B0  SEQ_NUM: 00000085
```

Figure 21. SMSDATA Formatting of the SSIVT Control Block

Formatting the Storage Management Subsystem Trace Table

You may use the **TRACE** option to format the SMS trace table. You may specify the **ASID**, **SELECT**, **START**, and **STOP** options to selectively format trace records. You may specify the **TADDR** option to explicitly specify the address of the trace table (ERTRE).

The following command formats all SMS trace table records that were the result of an **RCD** event for ASID 16. (Note that the command requires a decimal ASID number, but the formatted record shows the hexadecimal ASID number.) The formatted output is shown in Figure 22.

```
  VERBEXIT SMSDATA 'TRACE ASID(16) SELECT(RCD)'
```

```
IPCS OUTPUT STREAM ------------------------------------------- LINE 0 COLS 1 78
COMMAND ===>                                                   SCROLL ===> HALF
 ****************************** TOP OF DATA ***********************************
 --------------------- Storage Management Subsystem Data ----------------------
 ------------------- Storage Management Subsystem Trace Table -----------------

 7F6EC000 RTRN00000000 9C43013B8FD79102 00000100 E02 H-0010 P-000E S-0010 8003
          Trace Events        : RCD
          Recording Module    : IGDERRC2
          Header Identifier   : SYMREC FAILED! - RETURN/REASON CODES FOLLOW:
          Return Code         : 00000010
          Reason Code         : 00000F04

 7F6EED40 RTRN00000001 9C4301405285E402 000002D0 E02 H-0010 P-000E S-0010 8003
          Trace Events        : RCD
          Recording Module    : IGDMCSIS
          Control Block Name  : ADSR
          Control Block Data  :
           +00000000: E2D9F3F0 F9F0F1F4 F0F3F8F2 FFFF94B7   *SR3090140382....*
           +00000010: 9C430140 52702202 40404040 40404040   *... ....        *
           +00000020: 4040E2E8 E2E3C5D4 F140F5F7 F5F2D1C2   *  SYSTEM1 5752JB*
           +00000030: C2F2F2F2 F3400080 00000000 00000000   *B2223 ..........*
```

Figure 22. SMSDATA Formatting of the SMS Trace Table

SMSDATA Trace Record Format

Each trace record displayed in can be divided logically into two sections: The header section, consisting of a single *unlabeled* line, and the data section, consisting of *labeled* areas. These two sections correspond to the *fixed* and *variable* portions of the trace table entry (ERTRN control block), respectively.

SMSDATA Trace Record Header: In the *unlabeled* header, the first field is the address of the trace entry. Using the second trace record at address X'7F6EED40' as an example, the contents of the header are as follows:

RTRN00000001
The control block identifier for the trace entry, followed by a fullword sequence number. The formatter uses the sequence number to determine if gaps in the table exist and to skip over such gaps during formatting.

9C4301405285E402
The CPU clock value associated with the entry. It is a 64-bit binary number and is interpreted in the same manner as the CPU clock.

000002D0
The total length of the entry.

E02
This field contains the record type and the record identifier. The record type should always be either 'E' (for error) or'N' (for nonerror). You may use the identifier to determine a particular trace point within a module when more than one trace point exists. In this case the identifier is X'02'.

H-0010, P-000E, S-0010
The home, primary, and secondary ASIDs at the time the entry was written.

8003
The first two digits of this field indicate the format of the variable (data) portion of the record and should always be X'80'. The last two digits indicate the number of keyed items contained in the variable portion, in this case three.

SMSDATA Trace Record Data: In the *labeled* data, the first line identifies the trace event(s) associated with the trace record. The other areas describe the remaining variable trace data.

The trace record data contains keys (not displayed in the preceding example) describing the type of data contained in the entry. SMSDATA uses this key and other information in the trace record to determine the module that recorded the trace information, and to identify the variable data in the second trace record of the example as an ADSR control block. The ADSR is displayed in both hexadecimal and EBCDIC notation (as displayed in a standard dump) under the label “Control Block Data”.

Formatting the Storage Management Subsystem Intermodule Communications Area

Because SMS code is reentrant, storage is obtained automatically upon entry from an external caller. This storage is chained from the SMS intermodule communications area (IMC). You may use the **STORAGE** option of SMSDATA (with the IMC address and ASID), to format these data areas. The **STORAGE** option has three optional, nonpositional parameters: **DATA**, **FLO**, and **IMC**. You can specify these options individually or in any combination to SMSDATA in a single invocation.

You may use the **DATA** option to format the areas of storage used by the modules during processing. See Figure 23 on page 429 for an example of the **DATA** option.

You may use the **FLO** option to format the SMS procedure flow trace table. This is a wraparound trace table that captures the flow of control between modules and their return and reason codes. This trace table enables you to determine module flow immediately prior to an error. See Figure 24 on page 430 for an example of the **FLO** option.

You may use the **IMC** option to format all of the control blocks related to the automatic data area. This option is most useful when you suspect a problem in the management of the automatic data areas. The output produced for this option is similar to that shown in Figure 21 on page 426, although a different set of control blocks is displayed.

Formatting Automatic Data Areas

You may specify the **STORAGE** '**DATA**' parameter to format all automatic data areas, including the current area. The following command displays the output panel shown in Figure 23:

```
VERBEXIT SMSDATA 'STORAGE(7FFE3000,DATA)'
```

```
IPCS OUTPUT STREAM ------------------------------------------ LINE 0 COLS 1 78
COMMAND ===>                                                 SCROLL ===> CSR
********************** ******** TOP OF DATA ***********************************
--------------------- Storage Management Subsystem Data ----------------------
------------------ Storage Management Subsystem Autodata Areas ---------------

Autodata for Module: IGDCNS00
     Module Address: 01C46C10  Date: 06/14/87  FMID: HDP3310   PTF: NONE
     Address: 7FFE8028
     Length : 000002B0
     Register Save Area:
       Prev : 00005FB0
       Next : 00000000
         0-3  7FFFC2B0 7FFE8148 7FFE81DC 81C2ED28
         4-7  00C2B4F0 00C2B058 01DCBF00 FD000000
         8-B  008FBED0 808FB9A0 01C6374F 7FFE8048
         C-F  81C62750 -------- 81C62A08 7FFE82C8
     Autodata:
       +00000048: 81C62832 00000000 FD000008 0000000C   *.F.............*
       +00000058: 00000040 008FCE24 008FCE00 008FE270   *... ..........S.*
       +00000068: 008EDFF8 FD000000 008FBED0 808FB9A0   *...8............*
       +00000078: 01C6374F 7FFE8048 81C62750 7FFE82C8   *.F.......F.....H*
       +00000088: 81C62B80 00000000 FD000008 00000000   *.F.............*
```

Figure 23. SMSDATA Formatting of an SMS Automatic Data Area

As it formats a data area, SMSDATA attempts to determine which SMS module is associated with the storage. If successful, SMSDATA extracts module-related information from the SMS module map (ERMAP) and prints it with the formatted storage. Occasionally, SMSDATA might be unable to determine the owning module. If this occurs, SMSDATA does not print the module-related information, but prints the contents of the calling program's register 15, if possible.

The information displayed in Figure 23 is as follows:

ADDRESS
The address of the SSISP that contains the module's automatic data area.

LENGTH
The length of the SSISP.

REGISTER SAVE AREA
This is the module's register save area. The value for register 13 is shown as "--------" because register 13 does not exist within a register save area.

AUTODATA

The automatic data area used by the owning module. The offsets are relative to the SMS base register used and can be used when analyzing the assembler code. For example, if the owning module issues an X'5860B06C' instruction, then, using Figure 23 on page 429, register 6 would be loaded with the fullword at offset X'6C' into the automatic data area, which contains X'FD000000'. You should be especially careful when interpreting automatic data areas. SMS can use the same location for multiple purposes, so data may not be preserved.

Formatting the Storage Management Subsystem Procedure Flow Trace Table

You may specify the **STORAGE** '**FLO**' parameter to format the SMS procedure flow trace table.

The following command would display the output panel shown in Figure 24:

```
VERBEXIT SMSDATA 'STORAGE(7FFE3000,FLO)'
```

```
IPCS OUTPUT STREAM ------------------------------------------ LINE 0 COLS 1 78
COMMAND ===>                                                  SCROLL ===> CSR
 ****************************** TOP OF DATA ***********************************
 --------------------- Storage Management Subsystem Data ----------------------

 ---------------- Storage Management Subsystem Module Flow Trace --------------
  Call   -- IGDSSI01
  Call   -- IGDERCRT
    Return- IGDERCRT        Return Code: 00000000  Reason Code: 00000000
  Call   -- IGDERADD
    Return- IGDERADD        Return Code: 00000000  Reason Code: 00000000
  Call   -- IGDERADD
    Return- IGDERADD        Return Code: 00000000  Reason Code: 00000000
  Call   -- IGDOPSI1
    Return- IGDOPSI1        Return Code: 00000000  Reason Code: 00000000
  Call   -- IGDERDEL
 *Trace  -- IGDERDEL /                          ID: 02  TOD: 9C4301405285E402
    Return- IGDERDEL        Return Code: 00000000  Reason Code: 00000000
  Call   -- IGDENVN1
    Return- IGDENVN1        Return Code: 00000000  Reason Code: 00000000
  Call   -- IGDICMI0
    Return- IGDICMI0        Return Code: 00000000  Reason Code: 00000000
  Call   -- IGDERTRI
```

Figure 24. SMSDATA Formatting of the SMS Procedure Trace Table

Return entries are indented in the formatted SMS procedure flow trace table for readability.

**Trace* entries indicate that a trace record was written to the SMS trace table.

The TOD value at the end of the record is the CPU clock value associated with the record, which you can use to identify a specific trace entry.

The ID value identifies the trace point that was invoked. This is useful when a module contains multiple trace points.

Storage Management Subsystem Trace Facility

This section provides descriptions of the following items:
- How to create and format an SMS trace table
- SMS trace table entries
- Valid trace event options

The Storage Management Subsystem Address Space Trace Table

SMS provides a trace table for you to use when working with the IBM Support Center on a problem. The IBM Support Center may request a copy of the SMS trace table. You can obtain this by dumping the contents of the SMS address space to a SYS1.DUMP*nn* data set by using the DUMP operator command at the time of the failure. The dump options must minimally contain **RGN**, **CSA**, and **SQA**.

The SMS trace table can then be displayed using the IPCS SMSDATA verb. See “Storage Management Subsystem IPCS Dump Formatter (SMSDATA)” on page 420 and “Formatting the Storage Management Subsystem Trace Table” on page 427 for details.

For the SMS address space trace table to be created and active, tracing must be specified by the SETSMS command or by means of member IGDSMS*nn* in SYS1.PARMLIB. For details, see *z/OS DFSMSdfp Storage Administration Reference*. Entries in the trace table consist of certain 'events' that occurred during SMS processing. Areas of SMS processing that can be traced are defined in the next section titled Trace Event Options.

Entries in the trace table can be of two types: ERROR or NONERROR. If a trace table exists, an 'E' record is always written to the table. An 'N' record is written only if you selected that area of SMS processing for tracing and TYPE(ALL) is specified by using the SETSMS operator command or the IGDSMS*nn* member in SYS1.PARMLIB.

Trace Event Options

The following options are valid to specify with the SELECT keyword of the SMSDATA verb.

Table 107. Valid Options with the SELECT Keyword for the SMSDATA Verb

Trace Event	Description
ACSINT	ACS services interfaces.
ACSPRO	Perform ACS processing.
ALL	All of the other options.
CATG	SMS catalog services.
CDSC	Control data set changes.
CONFA	Activate a new configuration.
CONFC	Configuration changes.
CONFR	Return data from an active configuration.
CONFS	Configuration services.
DCF	Device control facility.
DISP	SMS disposition processing.
DPN	SMS device pool name selection subsystem interface.

Table 107. Valid Options with the SELECT Keyword for the SMSDATA Verb (continued)

Trace Event	Description
DSTACK	SMS data stacking service.
ERR	SMS error recovery and recording services.
IDAX	SMS interpreter/dynamic allocation exit.
MODULE	All SMS module entries/exits.
MSG	SMS message services.
OPCMD	SMS operator commands.
RCD	SMS recording services and fast VTOC/VVDS access.
SCHEDP	SMS scheduling services (prelocate catalog orientation).
SCHEDS	SMS scheduling services (system selection).
SMSSJF	SJF interfaces.
SMSSSI	SSI interfaces.
TVR	SMS tape volume record update facility.
VOLREF	SMS VOLREF services.
VTOCA	VTOC/data set services (add a volume to a data set).
VTOCC	VTOC/data set services (create a new data set).
VTOCD	VTOC/data set services (delete existing data set).
VTOCL	VTOC/data set services (allocate existing data set).
VTOCR	VTOC/data set services (rename existing data set).

Storage Management Subsystem Records Written to SYS1.LOGREC

This section shows an example of a formatted SYS1.LOGREC entry for an SMS-related software failure.

If SMS issues message IGD306I to report a failure, it also writes a software failure symptom record to SYS1.LOGREC. To print these records, use the EREP service aid and request a detailed edit report for software records. You cannot select individual records for printing. For details about using EREP, see *EREP User's Guide* and *EREP Reference*. If you have a dump associated with the IGD306I message, you can also use the IPCS LOGDATA verb to format and print the associated symptom records. Figure 25 shows a sample printed record.

Note: The contents of the PROBLEM ID field in the software symptom record match the problem ID field in the corresponding SMS failure-related IGD306I message. This enables you to identify the failure-related SYS1.LOGREC entry.

```
TYPE:  SYMPTOM RECORD       REPORT:  SOFTWARE EDIT REPORT           DAY YEAR
                                                        REPORT DATE: 204  87
SCP:   VS 2 REL 3                                        ERROR DATE: 197  87
                            MODEL:   3084                           HH MM SS.TH
                            SERIAL:  124440                   TIME: 02:58:56.64

SEARCH ARGUMENT ABSTRACT:

    PIDS/5695DF101 LVLS/DFP130#ESA RIDS/IGDERDMP PCSS/IGDERDMP01
    VALU/CIGDERDMP PRCS/0000000008 REGS/GR00020013 MS/IGD306I

SYSTEM ENVIRONMENT:

    CPU MODEL:   9672              DATE:   265  95
    CPU SERIAL:  000049            TIME:   02:58:56.64
    SYSTEM:      SYSTEM1           BCP:    MVS

    RELEASE LEVEL OF SERVICE ROUTINE:      HBB5520
    SYSTEM DATA AT ARCHITECTURE LEVEL:     10
    COMPONENT DATA AT ARCHITECTURE LEVEL:  10

    SYSTEM DATA:  00000000 00000000    |........|

COMPONENT INFORMATION:

    COMPONENT ID:             5695DF101
    COMPONENT RELEASE LEVEL:  130
    PID NUMBER:               5695DF1
    PID RELEASE LEVEL:        V1R30
    SERVICE RELEASE LEVEL:    NONE
    DESCRIPTION OF FUNCTION:  ERROR SERVICE-SMS ER
    PROBLEM ID:               IGD00001
    SUBSYSTEM ID:             SMS

PRIMARY SYMPTOM STRING:

    PIDS/5695DF101  LVLS/DFP130#ESA RIDS/IGDERDMP   PCSS/IGDERDMP01
    VALU/CIGDERDMP  PRCS/0000000008 REGS/GR00020013 MS/IGD306I

    SYMPTOM            SYMPTOM DATA     EXPLANATION
    ---------------    -------------    -----------
    PIDS/5695DF101     5695DF101        COMPONENT IDENTIFIER
    LVLS/DFP130#ESA    DFP130#ESA       PROGRAM PRODUCT RELEASE LEVEL
    RIDS/IGDERDMP     IGDERDMP         ROUTINE IDENTIFIER
    PCSS/IGDERDMP01    IGDERDMP01       SOFTWARE STATEMENT
    VALU/CIGDERDMP     IGDERDMP         ERROR RELATED CHARACTER VALUE
    PRCS/0000000008    0000000008       RETURN CODE
    REGS/GR00020013    00020013         GENERAL PURPOSE REGISTER
    MS/IGD306I         IGD306I          MESSAGE IDENTIFIER

THE SYMPTOM RECORD DOES NOT CONTAIN A SECONDARY SYMPTOM STRING.
FREE FORMAT COMPONENT INFORMATION:

    KEY = 0057     LENGTH = 000015 (000F)

    +000    C5D9C4D4    D7C5D9C1    C4C4C4C3    C6D9F2      |ERDMPERADDDCFR2 |

    KEY = 0039     LENGTH = 000031 (001F)

    +000    D9C5C3D6    E5C5D9E8    40C5D5E5    C9D9D6D5    |RECOVERY ENVIRON|
    +010    D4C5D5E3    40D5D6E3    40C1C3E3    C9E5C5      |MENT NOT ACTIVE |
```

Figure 25. Sample SMS SYS1.LOGREC Entry (Part 1 of 2)

```
    KEY = 0011     LENGTH = 000008 (0008)

    +000    C9C7C4C5    D9D9C3C1                            |IGDERRCA        |

    KEY = 0013     LENGTH = 000013 (000D)

    +000    C5D9D9C3    C16DD3D9    D9E3C5E7    E3          |ERRCA.LRRTEXT   |

    KEY = 0012     LENGTH = 000024 (0018)

    +000    00000000    00000000    00000000    00000000    |................|
    +010    00000000    00000000                            |........        |

    KEY = 0013     LENGTH = 000012 (000C)

    +000    C5D9D9C3    C16DC3D9    C5C1E3C5                |ERRCA.CREATE    |

    KEY = 0012     LENGTH = 000156 (009C)

    +000 LENGTH(0144) ==> ALL BYTES CONTAIN X'00'.
    +090    00000000    00000000    00000000                |............    |

HEX DUMP OF RECORD:

  HEADER
    +000    4C831800    00000000    0087197F    02585664    |.C.......G......|
    +010    66124440    30840000                            |... .D..        |

  SYMPTOM RECORD
    +000    E2D9F3F0    F8F4F1F2    F4F4F4F0    FFFF94B7    |SR3084124440..M7|
    +010    9D061664    5A20F030    40404040    40404040    |......0.        |
    +020    4040E2E8    E2E3C5D4    F140F5F7    F5F2D1C2    |  SYSTEM1 5752JB|
    +030    C2F2F2F2    F3400080    00000000    00000000    |B2223 ..........|
    +040    F1F00030    00640070    008000D4    00000000    |10.........M....|
    +050    011F0154    00000000    00000000    00000000    |................|
    +060    00000000    00000000    00000000    00000000    |................|
    +070    E2D9F2F1    F1F0F5F6    F6F5F2F8    F4F6F200    |SR2110566528462.|
    +080    F3F1F040    D5D6D5C5    40404040    F5F6F6F5    |310 NONE    5665|
    +090    E7C1F340    E5F3D9F1    F0404040    C5D9D9D6    |XA3 V3R10   ERRO|
    +0A0    D940E2C5    D9E5C9C3    C560E2D4    E240C5D9    |R SERVICE—SMS ER|
    +0B0    40404040    40404040    40404040    00000000    |            ....|
    +0C0    00000000    C9C7C4F0    F0F0F0F1    E2D4E240    |....IGD00001SMS |
    +0D0    40404040    D7C9C4E2    61F5F6F6    F5F2F8F4    |    PIDS/5665284|
    +0E0    F6F24040    D3E5D3E2    61C4C6D7    F37BF1F0    |62  LVLS/DFP3.10|
    +0F0    7BE7C140    D9C9C4E2    61C9C7C4    C5D9C4D4    |.XA RIDS/IGDERDM|
    +100    D7404040    D7C3E2E2    61C9C7C4    C5D9C4D4    |P   PCSS/IGDERDM|
    +110    D7F0F140    E5C1D3E4    61C3C9C7    C4C5D9C4    |P01 VALU/CIGDERD|
    +120    D4D74040    D7D9C3E2    61F0F0F0    F0F0F0F0    |MP  PRCS/0000000|
    +130    F0F0F840    D9C5C7E2    61C7D9F0    F0F0F2F0    |008 REGS/GR00020|
    +140    F0F1F340    D4E261C9    C7C4F3F0    F6C94040    |013 MS/IGD306I  |
    +150    40404040    0057000F    C5D9C4D4    D7C5D9C1    |    ....ERDMPERA|
    +160    C4C4C4C3    C6D9F200    39001FD9    C5C3D6E5    |DDDCFR2....RECOV|
    +170    C5D9E840    C5D5E5C9    D9D6D5D4    C5D5E340    |ERY ENVIRONMENT |
    +180    D5D6E340    C1C3E3C9    E5C50011    0008C9C7    |NOT ACTIVE....IG|
    +190    C4C5D9D9    C3C10013    000DC5D9    D9C3C16D    |DERRCA....ERRCA.|
    +1A0    D3D9D9E3    C5E7E300    12001800    00000000    |LRRTEXT.........|
    +1B0    00000000    00000000    00000000    00000000    |................|
    +1C0    00000000    13000CC5    D9D9C3C1    6DC3D9C5    |.......ERRCA.CRE|
    +1D0    C1E3C500    12009C00    00000000    00000000    |ATE.............|
    +1E0 LENGTH(0144) ==> ALL BYTES CONTAIN X'00'.
    +270    000000                                          |...             |
```

Figure 25. Sample SMS SYS1.LOGREC Entry (Part 2 of 2)

Locating the Intermodule Communications Area (IMC)

The intermodule communications area (IMC) is a vital link to diagnostic information. If you are not using the SMSDATA verb exit, you can locate the IMC in the summary portion (SUMDUMP) of a dump. You should use the IPCS/PRDMP SUMDUMP verb to format all dumps. This permits you to extract significant failure-related information from the IMC and other data areas.

To locate the IMC in a SUMDUMP, scan the EBCDIC portion for the eye-catcher **IGDSSIIM**. See Figure 26 on page 436 for an example of a SUMDUMP.

Storage Management Subsystem Storage Dump Contents

SVC dumps that are initiated by SMS are accompanied by an **IGD300I** message. SMS provides the following data areas and control blocks in SVC dumps that it initiates:

- The CVT
- The JESCT
- The JESCEXT
- The SSIB
- The SMS vector table (SSIVT)
- The SMS trace table header (ERTRE)
- The SMS module map (ERMAP)
- All SMS information related to automatic data areas (in the IMC)
- All related address spaces (including primary and secondary if in cross-memory mode)
- Storage subpool 230.

Note: The LPA, PSA, SQA, SUM, SWA, and TRT SDUMP parameters are always included in SVC dumps initiated by SMS.

Sample of a Storage Management Subsystem SUMDUMP

Figure 26 is a portion of a sample SMS SUMDUMP showing the intermodule communications area (IMC: identified by the EBCDIC eye-catcher **IGDSSIIM**), the recovery communications area (ERRCA: identified by the EBCDIC eye-catcher **IGDERRCA**), and the procedure trace table (ERPTT: identified by the EBCDIC eye-catcher **IGDERPTT**).

```
----- SUMLSTA RANGE -------- ASID 0010.

7F70A000   C9C7C4E2 E2C9C9D4 0001A368 00000000   7F7172D8 7F70A300 7F70A300 7F70A6B8  *IGDSSIIM...........Q...........*
7F70A020   7F70AA70 7F70FA90 7F70F000 7F70F000   00000000 00000000 00000000 000000E6  *..........0...0...............W*
7F70A040   00F7E280 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.7S...........................*
7F70A060   00000000 00000000 7F70A300 00000004   7F70F434 01BF53E4 81C9875A 01BF58B2  *................4....U.I......*
7F70A080   7F70FDCC 7F70FE70 81C80FF8 7F70FAB0   81C811C6 7F70A000 00000000 00000000  *.........H.8.....H.F...........*
7F70A0A0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *...............................*
********   LINES FROM 7F70A0C0 TO 7F70A0E0 ARE THE SAME AS THE ABOVE LINE
7F70A0E0   00000000 00000000 00000000 00000000   01DCBE80 00000000 00000000 00000000  *...............................*
7F70A100   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *...............................*
********   LINES FROM 7F70A120 TO 7F70A2E0 ARE THE SAME AS THE ABOVE LINE
7F70A2E0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *...............................*


----- SUMLSTA RANGE -------- ASID 0010.

7F70A300   C9C7C4C5 D9D9C3C1 03B80000 00000000   00000000 008C740C 10000000 7F7172D8  *IGDERRCA......................Q*
7F70A320   7F70A300 7F70A000 00000000 00000000   90000000 00000000 00000000 C5D9D9D6  *.........................ERRO*
7F70A340   D9E7C9E3 01BE87A2 0000FFFF 00C18B18   7F70F098 00000002 00000002 008C75E4  *RXIT.........A....0...........U*
7F70A360   7F70F14C 00F7E280 00C18B18 00FD76D8   7FFFD230 7F70F1C8 7F70F048 81BE81A8  *..1..7S..A.....Q..K...1H..0.....*
7F70A380   7F70F048 81BE82B4 81C82118 00000000   00000000 00000000 00000000 00000000  *..0......H....................*
7F70A3A0   00000000 00000000 C9C7C4E2 E2C9D9E3   40D9C5C3 D6E5C5D9 E8404040 40404040  *........IGDSSIRT RECOVERY       *
7F70A3C0   11040000 C9C7C4C5 D9C4D9D9 7F70A300   00000001 81C833A8 7F70A300 FFFF02C1  *....IGDERDRR.........H.........A*
7F70A3E0   C9C7C4E2 E2C9D9D9 7F70A300 00000002   81C998F8 7F70F14C FFFF01C1 C9C7C4C4  *IGDSSIRR.........I.8..1....AIGDD*
7F70A400   E2D7C5D9 7F70A300 00000003 01C80398   008C740C FFFF00C1 C9C7C4D4 C3E2C5D9  *SPER.........H.........AIGDMCSER*
7F70A420   7F70A300 00000004 81BF77D8 7F70FD28   000000C1 C9C7C4E2 E2C9D9C4 7F70A300  *...........Q.......AIGDSSIRD....*
7F70A440   00000005 81C993D0 7F712950 FFFF10C9   00000000 00000000 00000000 00000000  *.....I.........I..............*
7F70A460   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *...............................*
********   LINES FROM 7F70A480 TO 7F70A5A0 ARE THE SAME AS THE ABOVE LINE
7F70A5A0   00000000 00000000 00000000 00000000   01DCBE80 00000000 00000000 00000000  *...............................*
7F70A5C0   008C5B40 C9C7C4C5 D9C4D4D7 75010100   00000000 00000000 00000000 00000000  *... IGDERDMP..................*
7F70A5E0   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *...............................*
```

Figure 26. SMS Sample SUMDUMP—Eye-catchers in EBCDIC Section (Part 1 of 2)

```
----- SUMLSTA RANGE -------- ASID 0010.

7F719D00                     C9C7C4C5 D9D7E3E3   42400000 00000000 00000000 00000000  *        IGDERPTT. .............*
7F719D20   7F71A4C8 7F71DF28 00000000 00000000   00000000 00000000 00000000 00000000  *...H..........................*
7F719D40   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *...............................*
********   LINES FROM 7F70A0C0 TO 7F70A0E0 ARE THE SAME AS THE ABOVE LINE
7F719F40   00000000 00000000 C3C5D9C5 E2E30000   00000000 00000000 00000000 00000000  *........CEREST................*
7F719F60   00000000 00000000 C3C5D9C3 D9E30000   00000000 00000000 00000000 00000000  *........CERCRT................*
7F719F80   00000000 00000000 D9C5D9C3 D9E30000   00000000 00000000 00000000 00000000  *........RERCRT................*
7F719FA0   00000000 00000000 C3C5D9D9 C3F30000   00000000 00000000 00000000 00000000  *........CERRC3................*
7F719FC0   00000000 00000000 D9C5D9D9 C3F30000   00000000 00000000 00000000 00000000  *........RERRC3................*
7F719FE0   00000000 00000000 C3C5D9D4 E2C70000   00000000 00000000 00000000 00000000  *........CERMSG................*
7F71A000   00000000 00000000 C3D4C3E2 C3D40000   00000000 00000000 00000000 00000000  *........CMCSCM................*
7F71A020   00000000 00000000 C3C5D9C1 C4C40000   00000000 00000000 00000000 00000000  *........CERADD................*
7F71A040   00000000 00000000 D9C5D9C1 C4C40000   00000000 00000000 00000000 00000000  *........RERADD................*
7F71A060   00000000 00000000 C3D4C3E2 C3F20000   00000000 00000000 00000000 00000000  *........CMCSC2................*
7F71A080   00000000 00000000 D9D4C3E2 C3F20000   00000000 00000000 00000000 00000000  *........RMCSC2................*
7F71A0A0   00000000 00000000 C3C5D9C4 C5D30000   00000000 00000000 00000000 00000000  *........CERDEL................*
7F71A0C0   00000000 00000000 D9C5D9C4 C5D30000   00000000 00000000 00000000 00000000  *........RERDEL................*
7F71A0E0   00000000 00000000 D9D4C3E2 C3D40000   00000000 00000000 00000000 00000000  *........RMCSCM................*
7F71A100   00000000 00000000 C3D4C3E2 C3D50000   00000000 00000000 00000000 00000000  *........CMCSCN................*
7F71A120   00000000 00000000 C3C5D9C1 C4C40000   00000000 00000000 00000000 00000000  *........CERADD................*
7F71A140   00000000 00000000 D9C5D9C1 C4C40000   00000000 00000000 00000000 00000000  *........RERADD................*
7F71A160   00000000 00000000 C3D4C3E2 C3F20000   00000000 00000000 00000000 00000000  *........CMCSC2................*
7F71A180   00000000 00000000 D9D4C3E2 C3F20000   00000000 00000000 00000000 00000000  *........RMCSC2................*
7F71A1A0   00000000 00000000 C3C5D9C4 C5D30000   00000000 00000000 00000000 00000000  *........CERDEL................*
7F71A1C0   00000000 00000000 D9C5D9C4 C5D30000   00000000 00000000 00000000 00000000  *........RERDEL................*
7F71A1E0   00000000 00000000 D9D4C3E2 C3D50000   00000000 00000000 00000000 00000000  *........RMCSCN................*
7F71A200   00000000 00000000 C3D4C3E2 C9D90000   00000000 00000000 00000000 00000000  *........CMCSIR................*
7F71A220   00000000 00000000 C3C5D9C1 C4C40000   00000000 00000000 00000000 00000000  *........CERADD................*
7F71A240   00000000 00000000 D9C5D9C1 C4C40000   00000000 00000000 00000000 00000000  *........RERADD................*
7F71A260   00000000 00000000 C3C5D9C4 C5D30000   00000000 00000000 00000000 00000000  *........CERDEL................*
7F71A280   00000000 00000000 D9C5D9C4 C5D30000   00000000 00000000 00000000 00000000  *........RERDEL................*
7F71A2A0   00000000 00000000 D9D4C3E2 C9D90000   00000000 00000000 00000000 00000000  *........RMCSIR................*
7F71A2C0   00000000 00000000 C3C5D9D9 E3D90000   00000000 00000000 00000000 00000000  *........CERRTR................*
7F71A2E0   00000000 00000000 C3E2E2C9 D9D90000   00000000 00000000 00000000 00000000  *........CSSIRR................*
7F71A300   00000000 00000000 C3C5D9D9 C3C40000   00000000 00000000 00000000 00000000  *........CERRCD................*
7F71A320   00000000 00000000 C3C5D9D9 C3F30000   00000000 00000000 00000000 00000000  *........CERRC3................*
7F71A340   00000000 00000000 C3E2E2C9 D7C30000   00000000 00000000 00000000 00000000  *........CSSIPC................*
7F71A360   00000000 00000000 C3C5D9C1 C4C40000   00000000 00000000 00000000 00000000  *........CERADD................*
7F71A380   00000000 00000000 D9C5D9C1 C4C40000   00000000 00000000 00000000 00000000  *........RERADD................*
7F71A3A0   00000000 00000000 C3C5D9C4 C5D30000   00000000 00000000 00000000 00000000  *........CERDEL................*
7F71A3C0   00000000 00000000 D9C5D9C4 C5D30000   00000000 00000000 00000000 00000000  *........RERDEL................*
7F71A3E0   00000000 00000000 D9E2E2C9 D7C30000   00000000 00000000 00000000 00000000  *........RSSIPC................*
7F71A400   00000000 00000000 E3E2E2C9 D9D94040   40404040 40404040 40404040 40400100  *........TSSIRR               ..*
7F71A420   9C649624 CC086440 D9C5D9D9 C3F30000   00000000 00000000 00000000 00000000  *........RERRC3................*
7F71A440   00000000 00000000 D9C5D9D9 C3C40000   00000000 00000000 00000000 00000000  *........RERRCD................*
7F71A460   00000000 00000000 C3C5D9C4 D4D70000   00000000 00000000 00000000 00000000  *........CERDMP................*
7F71A480   00000000 00000000 C3E2E2C9 D7C30000   00000000 00000000 00000000 00000000  *........CSSIPC................*
```

Figure 26. SMS Sample SUMDUMP—Eye-catchers in EBCDIC Section (Part 2 of 2)

Storage Management Subsystem Return and Reason Codes

Use the SELECT specification of SMSDATA as an aid in determining which of the following return and reason code tables are applicable to a failure.

ACS Translator Return Codes

The ACS translator provides a return code in its output listing. 108 describes the conditions that cause the return code.

Table 108. SMS—ACS Translator Return Codes

Return Code	Meaning
0 (X'00')	Successful.
8 (X'08')	Invalid parameter list; the reason code is in register 0.
12 (X'0C')	The ACS routine was not translated because of severe user errors; the listing contains error messages.
16 (X'10')	Abend error in translator.
20 (X'14')	Translator error; the reason code is in register 0.

ACS Executor Return Codes

The ACS executor provides a return code in its output listing. Table 109 describes the conditions that cause the return code.

Table 109. SMS—ACS Executor Return Codes

Return Code	Meaning
0 (X'00')	Successful.
8 (X'08')	Invalid parameter list; the reason code is in register 0.
12 (X'0C')	ACS routine set a nonzero exit code; register 0 contains the nonzero exit code.
16 (X'10')	Abend error by executor. Dump of virtual storage taken. Reason code is set to 300.
20 (X'14')	Error in executor module or ACS routine invalid. Register 0 contains the reason code and is set to 300.
24 (X'18')	An error code is received during parsing. Check for syntax errors. Reason codes insignificant.

Storage Management Subsystem Initialization Return Codes

The SMS initialization subcomponent provides a return code in register 15. Table 110 describes the conditions that cause the return code.

Table 110. SMS Initialization Return Codes

Return Code	Meaning
0 (X'00')	Successful. If router return code is zero, then the function return code will always be error free.
4 (X'04')	Warning.
8 (X'08')	Unknown PC routine requested.
12 (X'0C')	SMAS unavailable.
16 (X'10')	Error condition.
20 (X'14')	Abend occurred.
24 (X'18')	SMS SSIB could not be found.
26 (X'1A')	Recovery environment does not exist.
28 (X'1C')	No wait for restart and unavailable SMAS.
30 (X'1E')	Program check. Restart is not possible.

Storage Management Subsystem Interface Return Codes

The SMS interface provides a return code in register 15. These return codes apply to the following SMS services:

- ACS services
- Catalog services
- Construct access services
- Device Pool Select
- Disposition services
- Interpreter/dynamic allocation exit (IDAX)
- Operational services
- Scheduling services
- Tape Volume Record Update
- VOLREF services
- VTOC/data set services (VDSS)

Table 111 describes the conditions that cause the return code.

Table 111. SMS Interface Return Codes

Return Code	Meaning
0 (X'00')	Successful.
4 (X'04')	The subfunction completed the request, but detected an unusual condition; examine the reason code for details.
8 (X'08')	The subfunction was unable to complete the request.
12 (X'0C')	Invalid SSOB extension format.
16 (X'10')	An indeterminate error occurred, from which the subfunction was unable to recover.
20 (X'14')	The caller requested not to be put into a wait (SSSAWAIT), but a wait is required because SMS is restarting.

Storage Management Subsystem Reason Codes

Subsystem Interface Reason Codes

Reason Code	Description
0000	Successful.
0004	Invalid SSSAID field.
0008	Invalid SSSAVER field.
0012	Unknown SMS subfunction requested.
0016	SMS does not support the specified SSI function code.
0020	Caller not authorized to perform requested SMS function.
0300	An SMS abend occurred.
0301	Data set not eligible for allocation on SMS-managed volume.
0302	Incompatible storage class and volume.
0303	MGMTCLAS may not be specified for non-SMS data sets.
0304	Use of referenced storage group not allowed by ACS routine.
0305	Storage class assigned to data set which references a non-SMS data set.
0306	Error during SMS processing, for example, SPAF or SJF error; the error was logged. If the message indicates an error in IEFSJRTE has occurred, check the JES JOBLOG for an error message that further explains the problem.
0307	Error in an ACS installation exit.

Reason Code	Description
0308	Not authorized to the specified SMS class.
0309	An SMS data set is not allowed within the scope of a JOBCAT/STEPCAT.
0310	Neither management class nor storage class can be specified for a data set that is not eligible to be managed by SMS.
0313	DSNTYPE cannot be specified for data set that is not managed.
0314	DSNTYPE specified is not supported for the specified data class.
0315	Invalid label specified for new mountable allocation.
0317	DSNTYPE of PIPE may only be specified with PATH.
0318	No storage groups of the same type as the referenced data set assigned to the referencing data set.
0320	z/OS UNIX system services is not installed on this system.
0399	Unable to add the recovery routine.
0400	Unable to obtain ASCB pointer.

ACS Services Reason Codes

Reason Code	Description
1001	The allocation request was rejected by the ACS installation exit or the ACS routine. If ACS routine is invoked by OAM, one of the ACS routines returned a nonzero exit code.
1011	A nonexistent SMS class was returned.
1012	The storage group routine assigned a nontemporary data set to a storage group which is not of type POOL.
1013	No storage group was assigned by the storage group routine.
1014	The explicitly specified SMS class does not exist.
1015	RACINIT failed.
1016	An error was detected by the ACS executor.
1017	Invalid expiration date.
1018	Invalid retention period.
1019	The TIME macro failed.
1020	The authorization check for management class failed.
1021	The authorization check for storage class failed.
1022	The authorization check for both management class and storage class failed.
1023	Nonzero return code from construct access services.
1024	Storage group routine assigned an object to a storage group which is not of type object or object backup.
1025	RACHECK failed; see SYS1.LOGREC record key 45 and key 6 for RACHECK return and reason codes.
1026	RACDEF failed.
1027	&SIZE or &MAXSIZE overflow.
1028	Both pool and tape storage groups derived, only pool assigned.

ACS Executor Reason Codes

Reason Code	Description
2001	The ACERO control block ID is incorrect.

Reason Code	Description
2002	The ACERO control block version is incorrect.
2003	The ACERW control block ID is incorrect.
2004	The ACERW control block version is incorrect.
2005	The ACOHD control block ID is incorrect.
2006	The ACOHD control block version is incorrect.
2007	The data set name is blank.
2008	Invalid data set name; missing qualifier; two qualifier separators found adjacent (for example, A..BD).
2009	Invalid data set name; the qualifier is longer than the maximum allowable.
2010	Incorrect ACCT_JOB field; the field length sums are greater than the maximum allowable.
2011	Incorrect ACCT_STEP field; the field length sums are greater than the maximum allowable.
2012	The data set organization field contains an invalid value.
2013	The data set record organization field contains an invalid value.
2014	The data set type field contains an invalid value.
2015	The execution mode field contains an invalid value.
2016	Parameter error in ACERO; the value of the day for the year is greater than 366.
2019	LABEL field contains an invalid value.
2020	DSNTYPE field is invalid.
2021	Member name is blank.
2022	Member name is invalid; missing qualifier.
2023	Member name is invalid; qualifier is too long.
2024	Data set name is invalid, for example, '.ABC.DEF'
2030	Error in an ACS routine; invalid operation code.
2031	Error in an ACS routine; invalid type of operand.
2032	Error in an ACS routine; invalid subscript for DSN.
2033	Error in an ACS routine; invalid value for SET.
2034	Error in an ACS routine; invalid fragment for write.
2035	Error in an ACS routine; invalid read/only variable.
2036	Error in an ACS routine; end of data before end of ACS routine.
2037	Error in an ACS routine; invalid table in which to find variable or literal.
2038	Error in an ACS routine; unknown compare operand.
2040	Error in an ACS routine; unknown mask.
2041	Error in an ACS routine using the STCK instruction; the epoch bit was invalid.
2042	Error in an ACS routine; unexpected clock time.
2043	Error in an ACS routine; the work area for the FILTER macro was too small.
2500	Error in the ACS executor; execution was unsuccessful because of add error recovery.

ACS Translator Reason Codes

Reason Code	Description
3050	Invalid control block ID.
3051	Invalid control block version.
3052	Invalid listing line length.
3053	Invalid listing page length.

Storage Management Subsystem Diagnostic Aids

Reason Code	Description
3100	Translation unsuccessful because of errors in the source routine.
3101	Error adding local error routine.
3102	Error when opening source data set.
3103	Error when closing source data set.
3104	I/O error on source data set.
3105	Error when opening listing data set.
3106	Error when closing listing data set.
3107	I/O error on listing data set.
3108	Unable to load I/O routine IGDACTIO.
3109	Unable to delete I/O routine IGDACTIO.
3110	The source data set DD was not specified.
3111	Invalid LRECL for the listing data set.
3112	Invalid use of member name with a sequential data set.
3113	No member name specified for a PDS.
3114	A nonexistent member name was specified for a PDS.
3115	An open for BLDL failed.
3116	BLDL failed.
3118	An unexpected STACK EMPTY occurred; unable to pop the label.
3119	An unexpected STACK FULL occurred; unable to push the label.
3121	The parser recognized a STACK OVERFLOW.
3127	Invalid DSORG for the source data set.
3128	Invalid DSORG for the listing data set.
3129	Error when reading the JFCB for the source data set.
3130	Error when reading the JFCB for the listing data set.
3131	RACF abend; not authorized to the source data set.
3132	RACF abend; not authorized to the listing data set.
3133	Invalid token grammar.
3134	TOKNR recognized a STACK OVERFLOW.
3135	Invalid routine type.
3136	Invalid ITEXT operation code.
3137	A branch was expected, but was not found.
3138	Unresolved label reference.
3150	X'37' ABEND; out of space.
3201	Unknown semantic action was encountered in IGDACTST.
3202	Unknown semantic action was encountered in IGDACTFL.
3203	Unknown semantic action was encountered in IGDACTIF.
3204	Unknown semantic action was encountered in IGDACTPE.
3205	Unknown semantic action was encountered in IGDACTSW.
3206	Unknown semantic action was encountered in IGDACTEX.
3207	Unknown semantic action was encountered in IGDACTWM.
3208	Unknown semantic action was encountered in IGDACTCN.
3209	Unknown semantic action was encountered in IGDACTCP.
3210	Unknown semantic action was encountered in IGDACTCO.
3211	Unknown semantic action was encountered in IGDACTPA.
3212	Unknown semantic action was encountered in IGDACTCB.
3213	Unknown semantic action was encountered in IGDACTCB.
3301	Unknown literal input from the parser to ACTFL.
3302	Unexpected call to ACTFL during INCLUDE/EXCLUDE processing.
3310	Unexpected input from the parser to ACTPE.
3323	Unexpected TOKEN_CODE from the parser.
3324	Unexpected VALUE_TYPE from the parser.
3325	Unexpected relational operator from the parser.
3326	Unexpected VALUE_TYPE from the parser.

Catalog Services Reason Codes

Reason Code	Description
4000	Locate was not performed; the volume is not SMS-managed.
4001	Catalog locate processing error.
4002	Locate failed for the data set referenced within the scope of JOBCAT/STEPCAT.
4003	Non-SMS-managed data set resides on SMS-managed volumes.
4004	A non-SMS-managed data set was found, but an SMS-managed volume serial was specified.
4006	Volume serials specified are not all in the same tape library.
4007	Volume serials specified are both tape and DASD.
4050	Both the SPAF and SJF tokens were specified; they are mutually exclusive.
4051	The locate was not performed; special, uncataloged system data set.
4052	SJF processing error.
4053	JES3 spool access facility (SPAF) processing error.
4054	Locate failure; the data set was not found.
4055	SMS-managed data set resides on non-SMS-managed volume(s).
4056	Error from LCS CBRXVOL.
4100	The LABEL JCL parameter indicates input processing only and DISP=MOD is specified. This is not valid.
4101	An LCS error occurred while trying to obtain device pool names.
4107	The volume list (as specified by the user or obtained from the catalog) contains a combination of DASD and library tape volume serials.
4108	The volume list (as specified by the user or obtained from the catalog) contains a combination of DASD and non-DASD volume serials, or the volume may be offline.
4109	The volume list (as specified by the user or obtained from the catalog) contains a combination of library and non-library tape volume serials.
4110	The volume list (as specified by the user or obtained from the catalog) contains a combination of SMS-managed and non-SMS-managed DASD volume serials.
4900	Error from BPXXSYS(STA) call for z/OS UNIX system services data set.
4901	Null path name specified.

Construct Access Services Reason Codes

Reason Code	Description
5000	Invalid parameter list.
5001	Insufficient space was passed by the calling program to contain the requested element.
5002	The SMS class or resource was not found in the active configuration (includes storage class, management class, data class, storage group and volume definition).
5003	No volumes are associated with the storage group.
5004	Volume verification; all volumes are defined to SMS, but do not belong to the same storage group.

Reason Code	Description
5005	Invalid user of provided data area.
5006	The volume list is invalid or inconsistent with the active configuration.
5007	The volume definition in the active configuration is associated with a nonexistent storage group.
5008	The count is required, but was not provided.
5009	Unable to acquire sufficient space in the user's address space.
5010	Volume verification; all volumes in the list are not defined to SMS.
5011	The volume verification list contains both volumes defined to SMS and undefined volumes.
5012	Extended interface; invalid subfunction specified.
5013	Caller requested specific release, and item does not exist in that release.
5014	Caller requested specific release, and length specified is invalid.
5015	Caller specified a release level value that is invalid. (It exceeds the maximum, is not allowed for the requested function, or the count requested is greater than 1.)

Configuration Services Reason Codes

Reason Code	Description
6001	Function code out of range.
6002	Invalid access token.
6003	Data set not accessed.
6004	Data set not formatted.
6005	Data set accessed in read mode.
6006	Invalid type specified by the caller.
6007	Error detected by error recovery.
6008	COMMDS must be accessed in update mode.
6009	CDS in memory is bad (for example, due to an abend); reaccess or refresh is required.
6010	CDS cannot be named ″ACTIVE″.
6030	Data set not found.
6031	Caller has insufficient authority.
6032	CI size is invalid (not 4K).
6033	Access token not initialized.
6034	Unable to perform access.
6035	Access mode is read; data set is empty.
6036	Invalid data set type.
6037	A required resource for accessing the data set is unavailable.
6038	A list ID field is invalid.
6039	DIV was unable to process the VSAM SHAREOPTIONS.
6040	Data set previously updated.
6041	Unable to perform save.
6042	Data set size has changed (COMMDS).
6043	Reserve required and not held.
6050	Data set is invalid. Run ISMF VALIDATE.
6051	Unable to perform unaccess.
6052	CDS is in 8-name mode and the SMSplex is in 32 name mode; conversion was not requested.
6053	The CDS is in 32-name mode and the SMSplex is in 8-name mode.
6054	Save failed; another system converted the configuration or COMMDS to 32-name mode.

Reason Code	Description
6058	Device services failed to return UCB for DD.
6059	Data set size changed Indicates high access activity.
6060	Unable to perform reserve.
6061	Unable to perform release.
6062	SVC99 error; error detected by SMS.
6063	SVC99 error; dynamic allocation or unallocation.
6064	Unable to find UCB to perform reserve or release.
6065	SVC99 parameter list is invalid.
6066	Environmental error prevented dynamic allocation or the operation in progress.
6067	DIV failure; undeterminable failure.
6068	DIV failure; system service failed (called by DIV). Problem could be caused by insufficient space to extend the data set. See the system console for IEC messages that might be related to the error.
6069	DIV failure; I/O error. See the system console for IEC messages that might be related to the error.
6070	No matching elements found.
6071	Element not found.
6072	Element was replaced.
6073	Invalid item count; must be 1.
6074	Invalid item length.
6080	Data set is in use.
6081	ENQ failed for data set.
6082	DEQ failed for data set.
6090	ACS routine not found.
6091	Data set is invalid.
6092	BCDEFLEN is zero.
6093	BCDEFLEN is bad.
6094	Extension length is zero.
6095	Extension length is bad.
6096	Extension offset is zero.
6097	Extension offset is bad.
6098	Extension offset is out of ascending order.
6099	Extension lengths plus definition length do not add up to total length specified in header.
6101	Function not allowed for the active configuration.
6102	Subsystem not active; no active configuration available.
6103	Error detected by SSI.
6104	Access to active configuration not allowed in update mode.
6110	System or system group name not found.
6111	System name or system group already exists.
6112	System/system group name list is full.
6113	No base configuration information found.
6114	System name type specified is invalid
6130	Data set is not a linear data set.
6131	Access failed; the data set type is incorrect. Either a COMMDS was found when an ACDS or SCDS was requested, or an SCDS or ACDS was found when a COMMDS was requested.
6700	Object or object backup storage group connected to system group.
6701	Object or object backup storage group connected to more than one system.
6702	Optical drive connected to system group.
6703	Optical drive connected to more than one system.
6750	Invalid DS separation profile name.
6751	Invalid OVERFLOW value.

Reason Code	Description
6752	Invalid extend SG name.
6753	Internal SMS parm list IGDIOPL contains incorrect information.
6754	DS separation profile record format is not fixed.
6755	Open of DS separation profile failed.
6756	Space constraint relief must also be specified when maximum volume count is specified.
6757	Invalid maximum volume count value.
6780	Separation profile unexpected end of file. Profile was modified during SMS configuration activation.
6781	Separation profile number of sep groups mismatch. Profile was modified during SMS configuration activation.
6782	Separation profile number of sep group DSNs mismatch. Profile was modified during SMS configuration activation.
6783	Error accessing data set separation profile. See message IGD06031I.
6801	RECORG is invalid.
6802	RECFM is invalid.
6803	Carriage control is invalid.
6804	LRECL is invalid.
6805	Key length is invalid.
6806	Key off is invalid.
6807	Retention period is invalid.
6808	Expiration day/date is invalid.
6809	Primary space is invalid.
6810	Secondary space is invalid.
6811	Directory blocks space is invalid.
6812	Volume count is invalid.
6813	CI size is invalid.
6814	CI free percent is invalid.
6815	CA free percent is invalid.
6816	VSAM share options is invalid.
6817	Name is invalid for management class.
6818	Minimum days on primary is invalid.
6819	Retain days archival is invalid.
6820	Expiration format is invalid.
6822	Backup frequency is invalid.
6823	Backup versions is invalid.
6824	Backup retention is invalid.
6825	Backup retention/no primary is invalid.
6826	Maximum retention is invalid.
6827	Name is invalid for storage class.
6828	Millisecond response is invalid.
6829	Sequential kilobyte response is invalid.
6830	Name is invalid for storage group.
6831	Storage group type is invalid.
6832	VIOMAX is invalid.
6833	VIOUNIT is invalid.
6834	System names are invalid.
6835	System statuses are invalid.
6836	Thresholds are invalid.
6837	Default management class is invalid.
6838	Volser is invalid.
6839	SMS status is invalid.
6840	MVS status is invalid.
6841	Name length is invalid.
6842	Unknown storage group for volume.

Reason Code	Description
6843	Storage group is type VIO, OBJECT, or OBJECT BACKUP or TAPE.
6844	Volser length not equal to 6.
6845	Duplicate system names found.
6846	Allocation unit less than 1.
6847	Scale factor (AVGREC) invalid.
6848	Availability is invalid.
6849	Direct response is invalid.
6850	Direct bias is invalid.
6851	Sequential response is invalid.
6852	Sequential bias is invalid.
6853	Level 1 days is invalid.
6854	Command/automatic migrate is invalid.
6855	Number of GDG elements is invalid.
6856	Data set deleted; backup version is invalid.
6857	Administrator/end user backup is invalid.
6858	RECORG and RECFM are mutually exclusive.
6859	Autobackup system is invalid.
6860	Autodump system is invalid.
6861	Automigrate system is invalid.
6862	Reason code is no longer used.
6863	Reason code is no longer used.
6864	Reason code is no longer used.
6865	Minimum days on primary is required.
6866	Level 1 days required.
6867	Backup versions is required.
6868	Backup versions (data set delete) is required.
6869	Backup frequency is required.
6870	Backup retention is required.
6871	Retain days only backup is required.
6872	MCPREL and MCPRECN are mutually exclusive.
6880	DSNTYPE is invalid.
6881	RECORG and DSNTYPE are mutually exclusive.
6882	Backup destination class is invalid.
6683	Expiration action is invalid.
6884	Retain days archive is invalid.
6885	Name is invalid.
6886	Expiration date is invalid.
6887	Destination is invalid.
6888	Data set prefix is invalid.
6889	Instruction DSN is invalid.
6890	Instruction member is invalid.
6891	Data set name is invalid.
6892	Data set member invalid.
6893	Destination is required.
6894	Output prefix is required.
6895	Instruction DSN is required.
6896	One DSN is required.
6897	Guaranteed backup frequency is invalid.
6898	Guaranteed backup frequency is required with AUTOBACKUP=Y.
6899	Invalid block size limit value.
6900	Time Since Creation value is invalid.
6901	Time Since Last Use value is invalid.
6902	Periodic value is invalid.
6903	Group ID is not unique.

Reason Code	Description
6904	Object Cycle Start is invalid.
6905	Object Cycle End is invalid.
6906	Initial Access is invalid.
6907	Library name is invalid.
6908	Library type is invalid.
6909	Drive name is invalid.
6910	Drive type is invalid.
6911	Time Since Creation and Time Since Last Use fields are mutually exclusive.
6912	Time Since Creation and Periodic fields are mutually exclusive.
6913	Time Since Last Use and Periodic fields are mutually exclusive.
6914	Periodic Monthly and Periodic Quarterly fields are mutually exclusive.
6915	Periodic Monthly and Periodic Yearly fields are mutually exclusive.
6916	Periodic Quarterly and Periodic Yearly fields are mutually exclusive.
6917	Library status is invalid.
6918	Drive status is invalid.
6919	Group qualifier is invalid.
6920	Duplicate library names in one storage group.
6921	Storage Group INTERVAL MIGRATION cannot be selected unless AUTO MIGRATION has also been selected.
6922	Copy technique is invalid in management class.
6923	Accessibility is invalid in storage class.
6924	Library name length is invalid in storage group.
6925	Library name is invalid in storage group.
6926	Primary space management requested without auto migration in storage group.
6927	Library name length is nonzero or library name found in a storage that is not of type OBJECT, OBJECT BACKUP, or TAPE.
6941	Entry default volume use attribute is invalid in library.
6942	Eject default is invalid in library.
6943	Library ID in library configuration database is invalid in library.
6944	Library device type is invalid in library.
6945	Entry default unit name is invalid in library.
6946	Media type is invalid in data class.
6947	Recording technique is invalid in data class.
6948	Compaction type is invalid in data class.
6949	Default entry data class is invalid in library.
6950	Management class name length is invalid in aggregate group.
6951	Management class name is invalid in aggregate group.
6952	SES cache length is invalid in base configuration.
6953	SES cache name is invalid in base configuration.
6954	Cache length is invalid in base configuration.
6955	Cache name is invalid in base configuration.
6956	SES cache count is invalid in base configuration.
6957	SES cache pointer is zero in base configuration.
6958	Pointer to aggregate backup parameters is 0 in management class.
6959	Aggregate backup name is invalid in management class.
6960	Aggregate backup name length is invalid in management class.
6961	Destination name is invalid in management class.
6962	Destination name length is invalid in management class.
6965	Indication of number of aggregate backup versions is invalid in management class.
6966	Indication of retention period for only version is invalid in management class.

Reason Code	Description
6967	Indication of retention period for extra versions is invalid in management class.
6968	Copy serialization option is invalid in management class.
6969	Retain only version is invalid in management class.
6970	Retain extra versions is invalid in management class.
6971	Number of versions to be backed up is invalid in management class.
6972	Destination pointer is zero.
6973	Incorrect number of destinations.
6974	Destination length is incorrect.
6975	Destination name is incorrect.
6976	Destination name length is incorrect.
6977	Destination name is incorrect.
6978	Time is incorrect.
6979	Netview class is incorrect.
6980	File encryption is incorrect.
6981	Transmit accompany files incorrect.
6982	Transmit control information is incorrect.
6983	Number of copies is not specified in aggregate group.
6984	Number of copies is invalid in aggregate group.
6985	Aggregate copy technique is invalid in management class.
6990	SES cache list is not in ascending order in base configuration.
6991	Cache set name is invalid in storage class.
6992	Cache set name length is invalid in storage class.
6993	Direct CF weight is invalid in storage class.
6994	Sequential CF weight is invalid in storage class.
6995	SES cache index is invalid in base configuration.
6996	Cache set pointer is zero in base configuration.

Disposition Processing Reason Codes

Reason Code	Description
7000	Invalid parameter list.
7001	Failure in catalog define of roll-in GDG.
7002	Unable to find the passed DDNAME.
7004	No volumes provided to catalog a data set on a managed, mountable volume.
7900	Null path name provided.
7901	Call to BPXXSYSC(UNL) failed to open for z/OS UNIX system services data set.
7902	Invalid disposition specified for z/OS UNIX system services.
7998	Enhanced volume list required to catalog a data set on a managed, mountable volume.
7999	Processing not supported for data sets on managed, mountable volumes with disposition other than CATLG.

Intersystem Communications Reason Codes

Reason Code	Description
10000	Operator specified cancel.
10001	Operator specified suspend.
10002	Operator specified terminate.
10003	Operator specified not to convert the COMMDS to 32 name mode.
10050	Internal COMMDS unavailable.
10051	Invalid user return area.
10052	Null configuration is active.
10053	COMMDS was switched.
10054	Device services failed to return UCB for DD.
10055	Unable to find UCB for access.
10100	Invalid IGDICMRT control block.
10101	Invalid current IGDICMTK control block.
10150	COMMDS validation: Invalid activation level.
10151	COMMDS validation: Invalid update count.
10152	COMMDS validation: Invalid volume count.
10153	COMMDS validation: Invalid volume length.
10154	COMMDS validation: invalid volume system data length.
10155	COMMDS validation: invalid system data length.
10200	Invalid IGDICMS0 function code.
10201	Invalid IGDICMC1 function code.
10202	Invalid IGDICMAC function code.
10203	Invalid IGDICMU1 function code.
10250	SMS resource is unavailable.
10251	SMS resource not serialized.
10300	Intersystem communications abend occurred.
10301	Null COMMDS accessed.
10302	Null COMMDS specified.
10303	Operational services ACDS activation failure.

Interpreter/Dynamic Allocation Reason Codes

Reason Code	Description
11000	Internal IDAX logic error.
11001	Invalid input condition; the SJF token is always required.
11007	Device Pool Select SSI error.
11008	Request was not for retrieve.
11009	Job or step request.
11010	SJF failure occurred with return code 0.
11011	SJF failure occurred with return code 4.
11012	Request was not for find.
11013	REFDD related request.
11014	SJF error.
11015	Indeterminate unallowable error.
11016	LCS CRBXVOL service failed.
11101	DSNTYPE cannot be specified with RECORG.

Message Services Reason Codes

Reason Code	Description
12000	The cell eye-catcher was not found.
12001	An invalid value was specified for FREEMSG.
12002	Unable to stack local recovery routine.
12003	The MCSCH.VERSION is incorrect.
12004	The MCSRS.VERSION is incorrect.
12005	The MCSMCBID field is incorrect.
12006	The MCSMCBVR field is incorrect.
12007	The MSGPTR is invalid.
12008	An unexpected abend occurred.
12009	Invalid MLTYPE code.
12010	The message length was too long.
12011	The message chain eye-catcher was not found.
12012	Unable to create recovery environment.
12020	The message ID does not exist in the message index.
12021	The subpool ID is invalid.
12022	The variant number is invalid.
12023	The REUSE value is invalid.
12024	The PREFIX value is invalid.
12025	The INCLUD value is invalid.
12026	The required input parameters were not specified.
12027	The SSSA control block ID is invalid.
12028	The SSSA control block version is invalid.
12029	REUSE and START_NEW_CHN are mutually exclusive.
12030	SSSARMSG is set to prevent returning messages.
12031	The SSSA is invalid.
12032	CPOOL abended.
12033	The message definition does not exist in any message module.
12040	The value for SMS construct or group name is too long.
12041	INCLUD = N and PREFIX = Y are mutually exclusive.
12042	The required variable parameters were not specified.
12043	Invalid TEXT type code.
12044	Invalid DATA type code.
12060	The descriptor code could not be determined.
12061	HRDCPY is invalid for MLWTO message.
12062	Invalid WTO type code in the message ID.
12063	A multiline message was provided for a WTOR request.
12064	A WTO abend occurred.
12065	A TPUT abend occurred.
12066	A RECORD abend occurred.
12067	The number of message lines exceeded the maximum.
12068	The pointer to the message is zero.
12069	Failed to pass check necessary for SVC35 after switching state and key.

Operational Services Reason Codes

Reason Code	Description
13000	Not authorized to activate a configuration (RACF).
13001	Too many request elements are on the queue.

Reason Code	Description
13002	The requested configuration no longer exists.
13003	An invalid request element was found on the queue.
13004	The active configuration was not found on the configuration chain.
13005	Too many configurations are on the configuration chain.
13006	Invalid configuration.
13007	Invalid parameter list.
13008	Invalid configuration root.
13009	No active configuration.
13010	Unexpected return code from SAF.
13011	Operator denied authorization to activate a configuration.
13012	Event audit failed.
13013	Status for update library on drive must be online or offline.
13014	Library or drive not found. Update failed.
13015	EXTRACT LIBRARY/DRIVE failed. Unexpected return/reason code from module IGDOPSD1.
13016	LIBRARY/DRIVE UPDATED failed. Unexpected return/reason code module IGDOPS10
13017	Library definition exists, no drive definitions exist, module IGDOPST1.

Scheduling Services Reason Codes

Reason Code	Description
14000	SMS resource connectivity conflict (causes the job to fail).
14001	A subfunction other than SSSAPLCO or SSSASCHD was specified.
14002	The saved scheduling information inconsistent with the current SMS configuration.
14003	A new list of system names has been determined.
14004	The job has no SMS resource requirements.
14005	The returned system list is restricted because of unavailable resources.
14006	JES3 spool access facility (SPAF) read error.
14007	A SPAF token was not specified.
14008	SMS resources are temporarily unavailable (reschedule the job).
14009	The SPAF subsystem name was not specified.
14010	Volume not in active configuration.
14012	An SMS service other than scheduling services has failed; the failing service return code is in SSSABSAC or SSSACSAC; the failing service reason code is SSSABSAR or SSSACSAR.

Subsystem Initialization Reason Codes

Reason Code	Description
15000	PSW supervisor state.
15001	PSW problem state.
15010	Initialization failed because of an MVS service.
15011	Initialization failed because of an SMS error.
15012	SMS is not defined to MVS as a subsystem.
15013	Module IGDSSI00 has already been activated.

Reason Code	Description
15014	Invalid SSIVT.
15015	Invalid SSIAT.
15016	SMS address space verification failed.
15017	Member IGDSMSxx does not exist.
15018	AXSET failed in IGDSSI01.
15019	LXRES failed in IGDSSI01.
15020	ETCON failed in IGDSSI01.
15021	ETCRE failed in IGDSSI01.
15022	Missing keyword was claimed, but no keywords were missing.
15023	Did not build the message.
15024	Did not build the message.
15025	Did not build the message.
15026	SMS initialization failed.
15027	IEEMB888 failed.
15028	IEEMB887 failed.
15029	Syntax error.
15100	Length of macro IHADFVT is bad.

VOLREF Services Reason Codes

Reason Code	Description
16000	The pointer to the CPL of the referencing data set is 0.
16001	The pointer to the CPL of the referenced data set is 0.
16002	The referencing data set disposition field contains an invalid value.
16003	The referenced data set disposition field contains an invalid value.
16004	Unable to build the recovery environment for VOLREF services.
16005	VOLREF services internal error.
16006	Failure in scheduling information processing for VOLREF services.
16007	The subfunction code was not the VOLREF services function code; valid codes are SSSAVOLR and SSSACPSW.
16008	SSSAASJ1 is 0.
16009	Neither the SJF token nor the CPL pointer is specified for the referenced DD.
16010	SJF retrieval request failed.
16011	SJF update request failed.
16012	Both the SJF and the SPAF tokens are 0.
16013	VOLREF services did not do any processing for this invocation.
16014	Both the SJF and the SPAF tokens are specified.
16015	Unexpected return code from the invoked routine.
16016	Delete of VOLREF services error recovery routine failed.
16017	A non-SMS-managed data set referenced an SMS-managed data set (VOLREF services).
16018	Catalog services failure.
16019	A construct access services failure occurred.
16020	VOLREF services internal error; DDFLAG value is invalid; valid values are 1 and 2.
16021	VOLREF services error recovery routine failed because no VOLREF services global data control block (IGDVRFGD) exists.
16022	The referenced data set is not cataloged (VOLREF services).
16023	The referencing data set is not cataloged (VOLREF services).
16024	The SPAF job token is 0.

Reason Code	Description
16025	The SPAF scheduling information token is 0.
16026	CLEANUP is on, SSSAASRA is 0.
16027	VOLREF services internal error; both the SCHEDULE and the ALLOCATE flags are set on in module IGDVRF00.
16028	VOLREF services did not do any processing for a new reference.
16029	VOLREF services internal error; the CLEANUP, SCHEDULE, and ALLOC flags are all 0 in module IGDVRF00.
16030	VOLREF services internal error; invalid SJF function code; valid codes are 0 = CLEANUP, 1 = RETRIEVE, 2 = UPDATE.
16031	No SMS SWBs were copied to the target DD because the source DD has no SMS SWBs.
16032	Processing did not complete for scheduling information because the referenced data set is new or temporary.
16033	Missing the required SJF token for a NEW/MOD data set.
16034	Message service failure.
16035	SJF access function flag or access field flag is invalid; valid function combinations are 1 = UPDATE/UPDATE and CLEANUP, 2 = RETRIEVE/RETRIEVE and CLEANUP, 3 = CLEANUP only; access field flag cannot be 0.
16036	The pointer to the global data block from the SW parameter list is 0.
16037	The pointer to SJF local storage is 0 (in the VOLREF services global data block).
16038	An SJF request to terminate failed.
16039	VOLREF services internal error; the referenced data set has no storage class and the SMS flag is off.
16040	Error; CTGWAVCT = 0 from IGDCAT00 when the return code = 0.
16041	VOLREF internal error. The function code is not acceptable to IGDVRFS1; valid functions are UPDATE or RETRIEVE.
16042	No SJF token was specified.
16043	UPDATE and RETRIEVE are mutually exclusive.
16044	An SJF update failed because SJF retrieve failed.
16045	VOLREF internal error; SJF REQNUM greater than SJF array DIM or less than 1.
16046	Add scheduling records failed.
16047	SMS internal error; SMS class not found for an SMS volume.
16048	SSSAASJ2 is 0.
16049	The caller is neither ALLOCATION nor JES3.
16050	The control block visual ID is not IGDVRMSW.
16051	The pointer to the SJACP is 0.
16052	An unexpected visual ID exists for control block IGDVRMGD; valid values are zeros or IGDVRMGD.
16053	ACS does not allow the storage group which was derived from the referenced data set.
16054	An ACS routine failed.
16055	IGDIDMAR failed.
16056	SSSA5SGN is 0 from an ACS routine.
16057	VOL=REF to GDG is not allowed.
16100	SSSAAVOL and SSSAMVL are mutually exclusive.
16101	Internal error from IDAX.
16102	Error returned by LCS.
16103	LCS indicated volume record not found for managed-mountable data set.
16104	No SJF token provided for referencing data set.
16999	VOLREF services failed because of an undetermined error.

VTOC/Data Set Services (VDSS) Reason Codes

Reason Code	Description
17001 (X'4269')	Duplicate data set name on volume.
17002 (X'426A')	No room in VTOC or VTOC index for DEFINE request.
17003 (X'426B')	Permanent I/O error during DELETE/RENAME request.
17006 (X'426E')	Average block length greater than 65535.
17012 (X'4274')	User not authorized to define data set.
17014 (X'4276')	Space request = 0.
17015 (X'4277')	Invalid function code.
17016 (X'4278')	Invalid request type.
17017 (X'4279')	TRKCALC (track calculation) failure.
17018 (X'427A')	DEVNAME (device name) not passed.
17036 (X'428C')	Directory space specified is greater than primary space.
17037 (X'428D')	DADSM installation exit rejected the request with a return code of 8.
17038 (X'428E')	The DADSM installation exit rejected the request and provided return code 4.
17040 (X'4290')	An error occurred in DADSM processing.
17042 (X'4292')	Expiration date has been overridden.
17044 (X'4294')	Invalid parameter list supplied by caller.
17045 (X'4295')	Space not specified for creation of data set.
17046 (X'4296')	Unexpected return code from DADSM.
17051 (X'429B')	Primary space requested is greater than 65535 tracks.
17053 (X'429D')	Invalid DADSM parameter or volume list.
17054 (X'429E')	DSCB not found on volume for DELETE/RENAME request.
17055 (X'429F')	DADSM password failure.
17056 (X'42A0')	Duplicate data set name on volume; RENAME request failed.
17057 (X'42A1')	Unexpired purge date; DELETE request failed.
17058 (X'42A2')	UCB unavailable for DELETE or RENAME request.
17059 (X'42A3')	Unable to mount volume for DELETE or RENAME request.
17060 (X'42A4')	DELETE or RENAME failed, data set is open.
17061 (X'42A5')	Insufficient RACF authority for DELETE or RENAME request.
17062 (X'42A6')	Invalid DADSM status code.
17065 (X'42A9')	Invalid DADSM return code.
17066 (X'42AA')	Unable to update last volume indicator.
17067 (X'42B0')	Convert CNTPCVNT TTR failed.
17068 (X'42B1')	I/O error while writing EOF mark.
17070 (X'42B3')	Data set allocated as extended format.
17071 (X'42B4')	Data set not allocated as extended format.
17072 (X'42B5')	Extended format requirement could not be met.
17073 (X'42B6')	Allocation as non-extended format will be attempted.
17074 (X'42B7')	Guaranteed space volume list is invalid.
17075 (X'42B8')	Data set DSORG must be RS for data set to be extended format.
17076 (X'42B9')	Non-SMS data sets cannot be extended format.
17077 (X'42C0')	An error has occurred in the storage group/volume selection.
17080 (X'42C3')	Data set not eligible for VSAM extended format.
17100 (X'42CC')	Unexpected catalog error.
17101 (X'42CD')	Duplicate name exists in the catalog.
17102 (X'42CE')	Catalog error while defining non-VSAM data set.
17103 (X'42CF')	Catalog error while defining VSAM data set.
17104 (X'42D0')	Catalog error while renaming data set.
17105 (X'42D1')	Catalog error while deleting data set.
17106 (X'42D2')	Catalog locate error.

Reason Code	Description
17107 (X'42D3')	Catalog error while updating volume list.
17110 (X'42D6')	Referenced data set not found in catalog.
17111 (X'42D7')	Invalid parameter list to IGDVTSCT; system error.
17112 (X'42D8')	Expiration date reset by catalog.
17115 (X'42DB')	Alternate index* DEFINE failed, base cluster not SMS.
17116 (X'42DC')	Locate for data component name failed.
17118 (X'42DE')	Data set referred to in LIKE= is neither a non-VSAM data set nor a VSAM cluster name.
17119 (X'42DF')	Null RDA.
17120 (X'42F0')	Bad return code from FAMS
17160 (X'43CC')	Data set is compressible.
17161 (X'43C0')	Bad return code from compression services.
17162 (X'43C1')	Bad return code from compression services.
17163 (X'43C2')	Data set characteristics do not allow compression.
17164 (X'43C3')	Compression not allowed by catalog management.
17165 (X'43C4')	Multivolume temporary data set cannot be extended format.
17200 (X'4330')	More volumes in TIOT than in catalog.
17201 (X'4331')	The SMS construct definition was not found.
17202 (X'4332')	Unexpected return code from construct access services.
17203 (X'4333')	The SMS volume definition was not found.
17204 (X'4334')	Unable to ENQ on the data set name.
17205 (X'4335')	Volumes specified are not in same storage group for a guaranteed space request.
17206 (X'4336')	Not enough volumes exist with sufficient space for the data set.
17207 (X'4337')	No accessible volumes for the data set with sufficient space.
17208 (X'4338')	Volume allocation was unable to obtain a list of allocated UCBs.
17209 (X'4339')	Sysplex Cache manager failure.
17210 (X'433A')	Dynamic unallocation of volume during data set CREATE failed.
17211 (X'433B')	An invalid volume list was specified.
17212 (X'433C')	Unable to retrieve DDNAME during data set CREATE.
17213 (X'433D')	Unable to dynamically allocate volume during data set CREATE.
17214 (X'433E')	Unable to dynamically allocate data set during data set CREATE.
17215 (X'433F')	No space specified for creation of VSAM data set.
17216 (X'4340')	Extend storage group used during EOV processing.
17217 (X'4341')	Unable to use volume for a guaranteed space request.
17218 (X'4342')	Volume list not passed for DEFINE request.
17219 (X'4343')	Unable to continue DEFINE request.
17220 (X'4344')	Greater than 59 volumes specified.
17221 (X'4345')	More than one volume requested for DEFINE temporary VSAM data set dsn.
17222 (X'4346')	Server is not available.
17223 (X'4347')	Overflow Storage Group used for initial allocation.
17224 (X'4348')	No storage groups returned by ACS services for a DEFINE request.
17225 (X'4349')	The SMS volume definition was not found for a guaranteed space request.
17226 (X'434A')	Insufficient number of volumes in eligible storage group(s).
17230 (X'434E')	Parameter list for VSAM EOV synch is invalid.
17231 (X'434F')	JFCB for VSAM EOV request indicate non-VSAM.
17232 (X'4350')	Data set name for VSAM EOV synch request could not be found.
17234 (X'4352')	VSAM EOV synch request failed because volume allocation failed.
17235 (X'4353')	VSAM EOV delete request failed because volume supplied was inconsistent with JFCB.
17236 (X'4354')	VSAM EOV delete request failed because volume deallocation failed.
17239 (X'4357')	VSAM EOV synch completed with no action taken.

Reason Code	Description
17240 (X'4358')	VSAM EOV synch request failed because the call to allocation to count the number UCBs failed.
17241 (X'4359')	VSAM EOV synch request failed because the call to update the TIOT failed.
17242 (X'435A')	VSAM EOV delete request failed because the call to update the TIOT failed.
17249 (X'4361')	VSAM EOV extend request failed because a volume is provided for non-guaranteed space request.
17250 (X'4362')	Cannot stack local recovery routine.
17251 (X'4363')	SWA manager LOCATE mode failure.
17260 (X'436C')	Storage group/volume not enabled to SMS.
17261 (X'436D')	Invalid volume list passed to old SMS data set allocation.
17262 (X'436E')	The volume could not be allocated.
17263 (X'436F')	Invalid INCLUDE list passed.
17271 (X'43A7')	Tertiary volumes used.
17272 (X'4378')	Volume selection failure for insufficient space.
17273 (X'4379')	Allocation failed for all volumes selected for the data set.
17274 (X'437A')	Volumes specified for a guaranteed space request do not belong to an eligible storage group allocation for the data set.
17275 (X'437B')	No eligible storage group has enough space for a “best fit” request.
17276 (X'437C')	DISP=MOD request would exceed maximum volume count of 59.
17277 (X'437D')	Non-VSAM volume selection failure status message.
17279 (X'437F')	A required pointer is zero.
17280 (X'43A0')	Alternate index DEFINE request with no volume serial provided.
17301 (X'4395')	Volume(s) not enabled or online for DELETE or RENAME request.
17302 (X'4396')	Unable to retrieve DD name for DELETE request.
17303 (X'4397')	Unable to allocate data set for DELETE request.
17304 (X'4398')	Non-SMS-managed data set on SMS-managed volumes.
17305 (X'4399')	The data set was not deleted or renamed; the volumes are a combination of SMS and non-SMS-managed.
17306 (X'439A')	Invalid dynamic allocation parameter list passed to dynamic allocation during DELETE request.
17307 (X'439B')	Invalid volume list or volume list pointer passed for DELETE or RENAME request.
17308 (X'439C')	DELETE or RENAME failed; LOCATE failed.
17309 (X'439D')	Dynamic unallocation error during DELETE request.
17310 (X'439E')	GETMAIN failed for DADSM VOLIST during delete.
17311 (X'439F')	SMS-managed volumes were specified for DELETE or RENAME of a non-SMS-managed data set.
17350 (X'43C6')	An invalid parameter list was supplied by the caller of VTOC data set services.
17351 (X'43C7')	Space parameter is not within value range.
17352 (X'43C8')	Unexpected reason code received from SJF.
17353 (X'43C9')	SMS-managed volumes specified for non-SMS request.
17354 (X'43CA')	Invalid return code from ACS services.
17355 (X'43CB')	Invalid space parameter in partial DSCB.
17357 (X'43CD')	GDG RECLAIM request failed.
17358 (X'43CE')	Volume section failed; volume(s) could not be selected after 5 attempts.
17359 (X'43CF')	The password specified is ignored.
17360 (X'43D0')	RETPD or EXPDT are ignored for a temporary data set.
17361 (X'43D1')	Non-SMS-managed volume specified for SMS-managed VSAM EXTEND request.
17362 (X'43D2')	Locate failed for SMS-managed or VSAM data set during EXTEND processing.

Reason Code	Description
17364 (X'43D4')	The data set was not authorized for the specified expiration date; the expiration date was set to the maximum allowed.
17365 (X'43D5')	An extent reduction request failed.
17366 (X'43D6')	Invalid return code from device information services.
17367 (X'43D7')	Data set organization conflicts with record format.
17368 (X'43D8')	Error in UCBSCAN.
17369 (X'43D9')	Error in EDTINFO.
17370 (X'43DD')	Error in UCBLOOK.
17371 (X'43F0')	Undefined VIO device type.
17372(X'43DC')	Data set separation could not be provided.
17373 (X'43DD')	Unexpected failure during data set separation processing.
17400 (X'43F8')	Data set referenced in LIKE= request is not cataloged.
17401 (X'43F9')	Unexpected return code from OBTAIN during LIKE processing.
17402 (X'43FA')	Allocation to read PDS directory for LIKE request failed.
17403 (X'43FB')	Unable to enqueue on the referenced data set.
17404 (X'43FC')	I/O error while trying to read directory of data set referenced in LIKE= request.
17405 (X'43FD')	Unable to determine device characteristics.
17406 (X'43FE')	Unable to dequeue the referenced data set.
17407 (X'43FF')	Catalog error while trying to locate data set referenced in LIKE= request.
17408 (X'4400')	Error return code from SJF while attempting to get data class.
17409 (X'4401')	Failure in data set properties merge.
17410 (X'4402')	LIKE=data set not on a DADSM volume.
17411 (X'4403')	SSSA request bits not set.
17412 (X'4404')	Internal error in copying FVT.
17413 (X'4405')	CPL prefix not found.
17414 (X'4406')	Request for unsupported function.
17415 (X'4407')	FVT not found.
17416 (X'4408')	Internal error messages found.
17417 (X'4409')	Volume set not allowed.
17418 (X'440A')	Error copying from caller key.
17419 (X'440B')	Error copying to caller key.
17420 (X'440C')	Internal error.
17421 (X'440D')	CWA not found.
17422 (X'440E')	FVT external header ID mismatch.
17423 (X'440F')	CPL external header ID mismatch.
17424 (X'4410')	Best fit interference not supported.
17430 (X'4416')	Tape data set allocation referenced a VSAM data set with LIKE.
17431 (X'4417')	Unable to determine directory blocks.
17432 (X'4418')	Inconsistent data set structure.
17433 (X'4419')	Allocation of tape data set failed.
17501 (X'4487')	Open failed for a USS file.
17502 (X'4488')	Create of a USS file failed.
17503 (X'4489')	Close failed for a USS file.
17510 (X'4496')	DSNTYPE PIPE specified without PATHNAME.
17511 (X'4496')	No PATHNAME specified.
17512 (X'4496')	Invalid DSNTYPE specified.
17800 (X'4588')	The SMS construct definition was not found.
17801 (X'4589')	Error return code from CNS.
17802 (X'458A')	The volumes specified by the caller are not in the same storage group for a guaranteed space request.
17803 (X'458B')	Not enough volumes with sufficient space for the data set.
17804 (X'458C')	No accessible volumes for the data set with sufficient space.
17806 (X'458E')	Allocation failed for all volumes selected for the data set.

Reason Code	Description
17807 (X'458F')	VSAM volume selection failure status message.
17808 (X'45A0')	Relax guaranteed space.
17813 (X'4595')	Bad return code from ACS.
17814 (X'4596')	Bad return code from DASDCALC.

Fast VTOC/VVDS Access Reason Codes

Reason Code	Description
04	Mismatch between format-1 DSCB and VVR/NVR.
05	FREEMAIN error (SVC 5).
08	Error accessing the VTOC (CVAFFILT READ).
12	Error accessing the VVDS (VVDS Manager).
32	Parameter list error (eye-catcher, length, UCB address, or function requested).
64	Major error (ABEND).
120	GETMAIN error (SVC 120).
139	CVAF error (SVC 139).

Device Control Facility Reason Codes

Reason Code	Description
18000	Unsupported device.
18001	An SJF error occurred.
18002	A construct access services error occurred.
18003	Invalid CASD parameter list.
18004	An error recovery and recording error occurred.
18005	The storage class was not found.
18006	Invalid SSSCB offset.
18007	The UCB was not found.
18008	Invalid DCFDS parameter list.
18009	No more subsystem to process.
18010	An AOMSERV error occurred.
18011	A 3990 storage subsystem could not be found.
18012	UCB address is 0.
18013	CMM error occurred.
18020	No device counts were returned by AOMSERV.
18021	No statistics were returned by AOMSERV.
18022	No SSDP pointer was returned by AOMSERV.
18023	The TIME macro failed.
18024	Invalid day from TIME macro.
18025	Invalid control block ID; expected IGDDCFCS.
18026	Invalid control block ID; expected IGDDCFCD.
18027	Unexpected return code.
18028	Unable to stack recovery routine.
18029	Abend occurred: no token returned.
18030	Abend occurred: token returned.
18031	SMS address space not available.
18032	Create recovery environment failed.

Reason Code	Description
18033	Add recovery routine failed.
18034	Zero divisor detected.
18035	Device table index is 0.
18036	Invalid search UCB (pointer is zero).
18037	SSID is forced offline.
18038	New hit ratio is in error. The read counts for the subsystem are so large that a meaningful cache hit ratio cannot be derived.
18039	Invalid input to IGDDCFCA.
18050	Invalid DD Index.
18051	Invalid UCB ID.
18052	No storage class for DD.

Operator Commands Reason Codes

Reason Code	Description
19000	Do not issue a diagnostic message.
19001	Required keyword not found.
19002	Required parameter not found.
19003	Invalid keyword.
19004	Command verb is not known.
19005	SMS is not active.
19006	Caller has insufficient authority.
19007	The command was abnormally terminated.
19009	Unable to add recovery routine.
19010	Caller did not provide target device number.
19011	Caller's requested range exceeds maximum.
19012	Unable to obtain storage.
19013	MVS message services failed.
19014	Storage group not defined.
19015	Volume not defined.
19016	Invalid keyword value.
19017	Invalid delimiter.
19018	Invalid syntax.
19019	Invalid blank.
19020	Null configuration.
19021	No systems.
19022	COMMDS experienced an error.
19023	Function code is not in library drive.
19024	Construction type is not in storage group, volume library, or drive for VARY command.
19025	Unsuccessful retrieval of the base configuration.
19026	The requested OAM display is unknown.
19027	OAM execution failed.
19028	OAM execution failed due to bad parameter to OAM.
19029	Stop building the display.
19030	D SMS OAM not allowed.
19031	The conversion of console name cannot be done, due to the error of CONVCON macro.
19032	UCBLOOK macro failed.
19033	Too many operands specified for keyword.
19034	Display request not recognized for RLS related information.
19035	Invocation of ?IGWSSCMD failed.

Error Recovery Reason Codes

Reason Code	Description
20000	Invalid subcomponent ID; no match was found during an IGDERRC3 SCID table search.
20001	No ERRCA pointer was passed when called by one of the error recovery modules (FRR/ESTAE—that is, PTR_RCA is 0).
20002	An SDWA pointer was not passed by the calling module when an SDWA pointer was required (that is, PTR_SDWA is 0).
20003	The RECORD request was not completed by the IGDERRC3 SDWA record module.
20004	The caller of RECORD service is attempting to record for the second or subsequent time.
20005	IGDERRCD (or other error recovery service) is already in process for this ERRCA, and an infinite loop was avoided by simply returning.
20006	SYMREC services returned a nonzero return code; all or part of the symptom record may not have been recorded in SYS1.LOGREC; see the SYMREC return and reason codes for details. (**NOTE:** the IGD306I SMS error message generated by IGDERRCD contains the SYMREC return/reason codes.)
20007	The PC Router returned a nonzero return code; the SMS address space was probably not available, or an error was detected during SMS processing.
20008	The caller of recording services requested the IGD306I SMS error message to be returned to the subpool specified, but the caller is in cross memory mode, so the request has been denied.
20009	A recording services module is trying to access the SSIVT, but cannot do so because the SSIBSUSE pointer from the SSIB is null.
20010	An active recovery environment already exists.
20011	ESTAE create failed.
20012	ESTAE cancel failed.
20013	The recovery environment is not active.
20014	The token of the recovery environment to be canceled does not match the token of the existing recovery environment.
20015	The local recovery routine stack is full.
20016	The local recovery routine stack is empty.
20017	The local recovery routine to be deleted cannot be found in the local recovery routine stack.
20018	The logic trace table header does not exist.
20019	A valid ERRCA does not exist.
20020	The abending error has not been recorded yet.
20021	The environment is not valid for a dump to be taken.
20022	Unable to obtain a lock.
20023	Unable to release a lock.
20024	SVC SDUMP failed.
20025	Branch SDUMP failed.
20026	Trace type not N or E.
20027	Trace data contains at least one zero pointer or length value.

Tape Volume Record Update Reason Codes

Reason Code	Description
21000	The deletion of the error recovery routine for the SMS tape volume record update failed.
21001	The system attempted to update the use attribute of the tape volume from scratch to private, but the use attribute for the tape volume was already private.
21002	The use attribute for the tape volume was not updated due to a bad return code and reason code from the CBRUXCUA installation exit.
21003	The tape volume record update failed due to an error from CBRXVOL.
21004	The retrieval of the tape volume record failed due to an error from CBRXVOL.
21005	The retrieval of a storage group definition failed due to an error from IGDCNS.
21006	The library name for the tape volume does not exist in any of the storage group definitions.
21007	The addition of the error recovery routine for the SMS tape volume record update failed.
21008	The USE attribute could not be changed in the tape volume record due to an error from CBRXLCS.
21009	The library name for the tape volume could not be retrieved due to an error from CBRXLCS.
21010	The retrieval of the storage group name(s) failed due to an error from SJF.
21011	Storage group names were not retrieved from SJF.
21012	The request to retrieve and update the tape volume record failed because the data required to retrieve the tape volume constructs was incomplete.
21013	The retrieval of the SMS construct names failed due to an error from SJF.
21014	The caller of SMS tape volume record update requested an invalid function.
21999	An abend occurred during the execution of SMS tape volume record update.

Device Pool SSI Codes

Reason Code	Description
22000	Bad SJF reason code
22001	Device Pools not found.

SMS Data Set Stacking Exit Codes

Reason Code	Description
23000	Unable to establish recovery environment for Data Set Stacking Service.
23001	Data Set Stacking Service internal error.
23002	Both SMS and managed mountable flags are on.
23003	Neither SJF token nor SMS construct information provided for data set being stacked on.
23004	SJF retrieval request failed.
23005	SJF update request failed.

Reason Code	Description
23006	Unexpected return code from invoked routine.
23007	Delete recovery routine failed.
23008	Construct access services failure.
23009	Message services failure.
23010	SJF TERMINATE request failed.
23011	Neither SJF token nor data set name provided for primary data set.
23012	Construct not found.
23013	The ACS routine failed the allocation.
23014	SSSA5SGN is 0 from ACS routine.
23015	Internal error from IDAX.
23016	Unexpected error from LCS.
23100	The ACS routine assigned a storage class to a data set being stacked on a non-SMS data set.
23101	No storage groups of compatible types assigned to a data set which is part of a data set collection.
23102	Storage group ACS routine did not allow the use of the primary data set's storage group for a data set being stacked on SMS tape.
23999	Data Set Stacking Service failed; error undetermined.

Chapter 17. Utility Service Diagnostic Aids

The following DFSMS Utility Programs offer diagnostic aids:
- IEBCOPY (PDSE Version)
- IEBCOPY (ESCON® Support)

IEBCOPY (PDSE Version)

The following diagnostic aids are provided by IEBCOPY:
- Maintenance level message
- Messages from DFSMSdfp subcomponents
- WTO/WTP of critical messages
- A SNAP/TRACE data set
- Debug keywords in the PARM field
- Extension of normal messages with additional information
- Additional debugging messages
- User abend after certain errors
- Abend after a disastrous error
- Consolidated data set information in dumps

Maintenance Level Message

Message **IEB1035I** contains a module identification field from the MCA data area. Each APAR or PTF incorporates the MCA so that the identification field contains the most recent maintenance applied. This field consists of:
- The assembly date
- The product FMID
- APAR/PTF number or timestamp. If no maintenance has been applied, then expect to find the word ″NONE″ possibly followed by 4 digits (which are the hour and minute of assembly).

Message IEB1066D displays the APAR or PTF level of each part or module in IEBCOPY. This is a debug message, and does not normally appear unless you supply an IBMSNAP DD statement code the DEBUG option in the invocation PARM field.

Messages from DFSMSdfp Subcomponents

IEBCOPY captures messages from other DFSMSdfp subcomponents that it may invoke, and displays any messages the subcomponents provide (that is, the IGW prefix messages) on the SYSPRINT file.

WTO/WTP of Critical Messages

Besides attempting to display messages on SYSPRINT, IEBCOPY can issue messages as WTOs. ROUTCDE=11,DESC=(7) turns messages into a WTP and associates them with the job. WTPs are included in the job's system message data set, and sent (TPUT) to the TSO/E user if IEBCOPY is running under TSO/E and PROFILE WTP is set.

The following messages are always issued as WTPs:

```
IEB103I members inaccessible
IEB133I storage unavailable
IEB139I I/O error (obsolete)
IEB150I too many directory blocks on track
IEB157I DD statement not found
```

© Copyright IBM Corp. 1979, 2002

```
IEB1021I through IEB1023I I/O error
IEB1025E DD name list error
IEB1028I key 0 super state
IEB1099S not APF authorized
IEB1120D internal error nnn
```

These messages are issued as WTPs only when SYSPRINT fails to open:

```
IEB165I directory I/O error
IEB168I members unreachable
IEB169I directory truncated
IEB172I open failed
IEB1068W JCL/label conflict
IEB1069E JCL/label conflict
IEB1070W directory blocks for new sequential data set
IEB1071E PDS on magnetic tape
IEB1072W PDS lacks directory
IEB1073W sequential data set has directory blocks
```

About half of the remaining messages are also sent to a TSO/E user (via WTP) if IEBCOPY is invoked as a TSO command processor. (This does not happen when the TSO CALL command is used to invoke IEBCOPY.) Note that invocation of IEBCOPY as a TSO command without using the TSO CALL command is not supported.

A SNAP/TRACE Data Set

The presence of an IBMSNAP DD statement causes option DEBUG to be forced and causes SNAP macros to be issued at many places in the code. Each SNAP has a unique ID=nnn assigned. Storage headings identify the CSECT issuing the SNAP and the control block or storage area being snapped.

The DCB parameters for IBMSNAP are those documented for the SNAP macro in the MVS macros book, RECFM=VBA, LRECL=125, BLKSIZE=882 or 1632.

An IBMSNAP DD statement might also be included with jobs that invoke IEBCOPY dynamically (such as SMP) to receive debug messages when debug keywords cannot be added directly to the IEBCOPY parameter list.

The presence of the IBMSNAP DD statement is the same as if any invoker of IEBCOPY coded “DEBUG” in the PARM field. Thus, an //IBMSNAP DD DUMMY statement is enough to get the debug messages on SYSPRINT. An //IBMSNAP DD SYSOUT=A statement gets both debug messages on SYSPRINT and SNAP dumps on IBMSNAP.

The first item that appears in the IBMSNAP data set is a dump of the parameters provided to IEBCOPY, including the DD name list. Use this when IEBCOPY is dynamically invoked to find the incorrect parameters being passed to IEBCOPY.

When a load or unload of a PDSE is in progress, each record read from or written to the unload data set is SNAP. A dump of the parameters provided to IEBCOPY including the DD name list. Use this when IEBCOPY is dynamically invoked to find the incorrect parameters being passed to IEBCOPY.

Debug Keywords in the PARM Field

You can activate several of the debugging aids by including additional, debug keywords in the PARM field at IEBCOPY invocation.

DEBUG-related Keywords:

DBTXT Causes additional information to be inserted into IEBCOPY normal processing messages (that is, the message numbers listed in the MVS messages manual) and also prints IEB1066D messages.

DBMSG Causes additional messages to be issued. These messages always have a suffix “D” for Debug, and are not included in the MVS message manual. They can only be interpreted by customer service or development by looking at the code where the messages originate. Customers are not expected to understand these messages.

DBDMP Causes IEBCOPY to print a diagnostic dump on SYSPRINT when an internal error, program check, or user abend occurs.

DBABE Causes IEBCOPY to abend after issuing specific messages. A user abend is issued corresponding to the message number. For example, after message IEB1021I abend U1021 is issued. DBABE implies DBDMP.

DEBUG Activates DBTXT, DBABE, DBDMP, and DBMSG.

TRACE-related Keywords. Only the first 4 characters are required:

TRIOT I/O trace without buffers or CCWs. This is TRRIOT+TRWIOT.

TRBUF I/O trace with buffers and without CCWs. This is TRRBUF+TRWBUF.

TRCCW I/O trace without buffers and with CCWs. This is TRRCCW+TRWCWW.

TRACE I/O trace with buffers and CCWs for both reads and writes.

TRRIOT, TRRBUF TRRCCW, TRRALL
Apply only to read operations.

TRWIOT,TRWBUF TRWCCW, TRWALL
Apply only to write operations.

Special Test Related Keywords:

MINW=*value* Reduces the minimum for the variable length GETMAIN request.

FREE=*value* Limits the amount of storage to contain Control Table Entries and saved PDS directory information.

Extension of Normal Messages with Additional Information

When the DBTXT option is active, additional fields are inserted in regular IEBCOPY messages. For example, message:

```
IEB120I <dd_name> VALIDATION ERROR
```

becomes:

```
IEB120I <dd_name> VALIDATION ERROR AT OFFSET X'hhhh' X'oooo'
```

where *hhhh* and *oooo* are offsets into the module (IEBDV0) detecting the error.

Additional Debugging Messages

When the DBMSG option is in effect, messages IEB1040D through IEB1049D are issued along with many other D suffix messages. These messages trace the values of various fields. In addition, these messages require that the IEBCOPY program listings be consulted to determine the fields and point in execution where the messages are issued.

User Abend after Certain Errors

When the DBABE option is in effect, IEBCOPY issues a user abend after issuing any of these messages:

```
U1021 -- IEB1021I I/O error SYNAD text
U0172 -- IEB172I  <dd_name> COULD NOT BE OPENED
U0183 -- IEB183I Unexpected EOF on unload dataset
U0184 -- IEB184I Physical block too big
U1044 -- (no message) I/O intercept posted reading
U2044 -- (no message) I/O intercept posted reading
```

Abend after a Disastrous Error

A system 0C1 abend might occur when IEBCOPY detects a catastrophic situation. If an event happens that ″absolutely should not occur,″ a short string of text might be present in the load module at the abend PSW location.

These abends are controlled by conditional assembly and are not normally included in the shipped product. However, abends might be present in ESP/ISP/EIO/pre-release code or test APARs (AYnnnnn).

Consolidated Data Set Information in Dumps

The program static data area in IEBCOPY is at the start of the load module and is page aligned. It is the CSECT IEBCOMCA and is always based on R4. Certain control blocks always appear at a fixed offset. Most useful are:

MCA+X'1000' Input file D$C
MCA+X'1200' Output file D$C
MCA+X'1400' SYSUT3 D$C
MCA+X'1600' SYSUT4 D$C
MCA+X'1800' Load file D$C
MCA+X'1A00' Unload file D$C

The D$C contains a DD name and a DCB pointer at offset +X'60', the DEVTAB at +X'A0', the JFCB at +X'C0' and the DSCB at +X'174'.

Reversion to Channel Programs That Are Not ECKD (ESCON Support)

Coding OPTCD=W in a DD statement that references a PDS causes channel programs basically equivalent to those of IEBCOPY before ESCON to be used. All performance advantages from the ESCON copy will be lost. Use of these channel programs is provided to help problem determination and to allow a fall-back position when the ESCON channel program causes a problem.

Chapter 18. VIO Diagnostic Aids

The following major diagnostic aids are provided for virtual I/O (VIO):

- A description of how to analyze diagnostic information that is recorded by recovery routines
- An explanation of messages that are issued by VIO.

Analyzing Diagnostic Information Recorded by Recovery Routines

The information that is recorded on SYS1.LOGREC and SYS1.DUMP can be retrieved by the IFCEREP0 service aid and the IPCS dump formatter, respectively.

Analyzing SYS1.DUMP Data Sets

To get a dump of a SYS1.DUMP data set, use IPCS as described in *z/OS MVS IPCS User's Guide*. The SDUMP macro allows VIO recovery routines to take limited dynamic (automatic) dumps containing volatile work areas before the areas are swapped out or modified. SDUMP macros are issued to generate dumps that record the status of the current environment whether retry or percolation is performed.

Analyzing the SYS1.LOGREC Data Set

Information written by recovery routines to the SYS1.LOGREC data set is used primarily to monitor incidents—both when retry of a failure is attempted and when percolation to the next recovery routine takes place. To get a dump of the SYS1.LOGREC data set, use the IFCEREP0 service aid as described in *z/OS MVS Diagnosis: Tools and Service Aids*.

IFCEREP0 formats the standard area—the first 404 bytes—of each SDWA into a series of titles, each followed by pertinent data found in the standard area. IFCEREP0 will put the variable area—the last 108 bytes—of each SDWA in an alphanumeric or hexadecimal format, whichever is appropriate.

The VIO recovery modules, IDDWIFRR and IDAVBPR1, put the data in the variable area in alphanumeric format so that the following messages can be constructed.

Module IDAVBPR1 constructs a diagnostic message of the following format:

```
jobname,VIO,modname,addresses
```

where:

jobname Specifies an 8-byte job name from the TIOT.

modname Specifies the VIO module in control, or last in control, when the error was detected.

addresses Specifies a list of addresses of the pertinent control blocks that existed at the time of the error. The list supplied depends on the module in control at the time of error. The format and list for each module are as follows:

Module	Format and List
IDAVBPC1	A(DSPC)=a
IDAVBPO1	A(DSPC)=a,A(VOP1)=a

© Copyright IBM Corp. 1979, 2002

IDAVBPP1	A(DSPC)=a,A(BUFC)=a
IDAVBPS1	A(DSPC)=a

where:

a is the address of the control block.

DSPC locates the DSPCT header.

VOP1 locates the VOPEN parameter list passed to module IDAVBPO1.

BUFC locates the VREADWR parameter list passed to module IDAVBPP1.

Module IDDWIFRR constructs a diagnostic message of the following format:

```
JOBNAME=jobname,VIO LMOD=IDDWI,function,addresses
```

where:

jobname Specifies an 8-byte job name from the TIOT.

function Specifies a code that indicates which VIO module was in control, or last in control, when the error was detected. If the code is WIEXCP, then module IDDWIAPP, IDDWICPI, or IDDWITRM was in control. If the code is WICLOSE, module IDDWICLS was in control. If the code is WIJOURN, module IDDWLIRN was in control.

addresses Specifies a list of addresses of the pertinent control blocks which existed at the time of the error. The list supplied depends on the module in control at the time of error. The format and list for each module are as follows:

Module	Format and List
IDDWIAPP	A(IOB)=a,A(VDSCB)=a
IDDWICLS	A(VDSCB)=a
IDDWICPI	A(IOB)=a,A(VDSCB)=a
IDDWIJRN	A(TCB)=a,A(VDSCB)=a
IDDWITRM	A(IOB)=a,A(VDSCB)=a

where:

a is the address of the control block.

IOB locates the input/output block.

TCB locates the task control block.

VDSCB
locates the virtual data set control block.

Messages Issued by VIO

IEC006I UNABLE TO ACTIVATE A VIO DATA SET DURING RESTART PROCESSING

Explanation: Auxiliary storage management (ASM) was unable to reset its control blocks for a VIO data set to the data set's status at the time it was journaled. This message should be followed by message IEF086I.

System Action: Terminate restart processing.

Programmer Response: Rerun job. Also refer to message IEF086I.

Problem Determination: Refer to Table 1, items 1, 3, 4, and 29, in:
- *z/OS MVS System Messages, Vol 1 (ABA-AOM)*
- *z/OS MVS System Messages, Vol 2 (ARC-ASA)*
- *z/OS MVS System Messages, Vol 3 (ASB-BPX)*
- *z/OS MVS System Messages, Vol 4 (CBD-DMO)*
- *z/OS MVS System Messages, Vol 5 (EDG-GFS)*
- *z/OS MVS System Messages, Vol 6 (GOS-IEA)*
- *z/OS MVS System Messages, Vol 7 (IEB-IEE)*
- *z/OS MVS System Messages, Vol 8 (IEF-IGD)*
- *z/OS MVS System Messages, Vol 9 (IGF-IWM)*
- *z/OS MVS System Messages, Vol 10 (IXC-IZP)*

Refer to message IEF086I.

ASM Module that Detects the Error: ILRGOS.

VIO Module that Issues Message: Module IDAVBPJ2 examines the return code in register 15 that is returned by ASM, and, if the return code is greater than 8, IDAVBPJ2 issues the message.

Chapter 19. VSAM Diagnostic Aids

This section contains overviews of the major diagnostic aids that are provided for VSAM subcomponents, and cites publications that contain more detailed information. Each subcomponent is discussed in a separate section:

- "Access Method Services (AMS) Diagnostic Aids" on page 473
- "Catalog Management Diagnostic Aids" on page 477
- "VSAM OPEN/CLOSE/End-of-Volume (O/C/EOV) Diagnostic Aids" on page 479
- "VSAM Record-Level Sharing (RLS) Diagnostic Aids" on page 482
- "VSAM Record Management (R/M) Diagnostic Aids" on page 492
- "VSAM Record Management Return and Reason Codes" on page 504

Access Method Services (AMS) Diagnostic Aids

This section explains the diagnostic aids provided for access method services (IDCAMS), explains how to find key areas in a dump, and offers suggestions for isolating different types of problems.

The following major diagnostic aids are provided for access method services:

- Trace tables, which provide a trace of the flow of control between modules and within modules.
- Dump points, which provide the facility to dump selected areas of virtual storage and take a full region dump.
- The TEST option, which you can set to print out the trace tables or to obtain dumps at selected dump points if access method services is invoked with a batch job.
- Termination codes and full region dumps, which are produced when the processor detects an unrecoverable condition.

Trace Tables

The processor maintains two trace tables during each execution: the inter-module trace table, which records the flow of control *between*modules, and the intra-module trace table, which records the flow of control *within* modules.

You can find the trace tables in any full region dump, you can print them using the TEST option, or you can display them on a TSO/E terminal. "TEST Option" on page 474 explains how to print the tables in a dump.

Inter-Module Trace Table

The inter-module trace table begins with the characters INTER and contains the IDs of the last 20 modules that had control. The module IDs are the last 4 characters of the module name. For example, if the trace appears as follows:

```
INTER ... SA01 EX01 RI01 RI02
```

then you know that IDCRI02 had control at the time of the dump.

The inter-module trace table is updated by the system adapter not only as each module is entered, but also upon return from a module. Thus, if RI01 calls TP01 which calls IO01 and then returns back to RI01, the trace table appears as follows:

```
INTER ... RI01 TP01 IO01 TP01 RI01
```

© Copyright IBM Corp. 1979, 2002

Intra-Module Trace Table

The intra-module trace table begins with the characters INTRA and contains the last 20 trace points encountered within modules. Each module has trace points placed at key locations, for example, at the start of procedures and around calls to other modules.

The IDs of the trace points consist of 4 characters: the first 2 characters are the mnemonic identifier of the module being traced, and the last 2 characters identify a specific point within the module. The expansion of the UTRACE macro for trace ID DLLC appears as follows:

```
OLDERID2 = NEWERID2;
NEWID2 = 'DLLC';
```

Dump Points

The IBM Support Center VSAM Customer Support will provide dump point information when a dump of a region or selected area is required for diagnosing a problem.

Each module has built-in dump points that invoke diagnostic dumping routines if the TEST option is in effect. The dump points have been placed at key locations in each module (for example, around calls to other processor and nonprocessor modules). Each dump point specifies what information will be dumped. Some dump points allow symbolic dumping of selected areas of virtual storage (for example, parameter lists or return codes); all dump points allow dumping of the full region and printing of the trace tables.

Certain access method services modules have the dumping of selected areas of virtual storage built in. Dump points can be used to dump these selected areas. The areas dumped vary with each dump point and are identified with descriptive codes.

Dumping of selected areas can occur with or without a full region dump, as described in “TEST Option”.

TEST Option

If you invoked access method services in a batch job, you can use the TEST option to activate the printing of diagnostic output at selected points within access method services. The TEST option is controlled by the TEST keyword as explained in “TEST Keyword” on page 475.

The TEST option provides you with the ability to print:
- The inter-module and intra-module trace tables.
- Selected areas of virtual storage.
- Full region dump.

Each variation of the TEST option provides an additional level of information. The possible variations are:
- Print the trace tables only
- Print the trace tables and selected areas of virtual storage
- Print the trace tables and selected areas of virtual storage and take a full region dump.

TEST Keyword

You can enter the TEST keyword either in the PARM field of the EXEC card that invokes the processor, or on a PARM command. By using the PARM command, you can turn the TEST option on and off or change the TEST option for different function commands.

The format of the PARM command follows.

PARM	**[TEST({[TRACE]** **[AREAS** (*areaid* [*areaid*...])] **[FULL** ((*dumpid* [*begin* [*count*]]) [(*dumpid*...)....])]| **OFF})]**

where:

TEST({[TRACE] [AREAS (*areaid***[*areaid*...])] [FULL ((***dumpid***[*begin* [*count*]]) [(***dumpid***...)...])]\| OFF})**
Specifies the diagnostic aids to be used. After the TEST option has been established, it remains in effect until it is reset by another PARM command. The TRACE, AREAS, and FULL parameters may be used concurrently.

TRACE
Specifies that trace tables are to be listed whenever the built-in dump points of the processor are encountered.

AREAS(*areaid* **[***areaid***...])**
Identifies modules that are to have selected variables dumped at their dump points. areaid is a 2-character area identifier defined within the implementation.

FULL ((*dumpid* **[** *begin* **[** *count***]]) [(***dumpid***...)....])**
Specifies that a region dump, as well as the trace tables and selected variables, is to be provided at the specified points. dumpid specifies the 4-character identifier of the dump point.

begin A decimal integer that specifies the number of times the program is to go through the dump point before beginning the dump listing. The default is 1.

count A decimal integer that specifies the number of times through the dump point that dumps are to be listed. The default is 1.

If the FULL keyword is used, an AMSDUMP DD statement must be provided. For example:

```
//AMSDUMP DD SYSOUT=A
```

OFF Specifies that testing is to stop.

Each time a PARM command is specified, the TEST parameters override the TEST parameters in effect from the previous PARM command.

Figure 27 on page 476 shows a portion of the output from the command:

```
PARM TEST ( FULL (LCTP,2,1) )
```

and a portion of the dump produced.

The trace tables and the selected area, DARGLIST, are printed each time the dump point LCTP is encountered. A full region dump is produced the second time that dump point LCTP is encountered.

How to Use the TEST Option

If a problem occurs and you have no idea which modules are involved, run the job again with the TRACE keyword. From the inter-module trace table you can identify the modules involved. The TRACE keyword, however, produces a large amount of output.

If you suspect that specific modules are involved, you can rerun the job with the AREAS keyword and specify the identifiers of several suspected modules. You will obtain trace output only for the specified modules.

When VSAM Customer Support identifies the dump points at which a full dump should be taken, rerun the job with the FULL keyword. The AREAS and FULL keywords can be used in combination to obtain trace tables and selected areas throughout several modules, but a full region dump will be taken only at selected points.

```
IDCAMS  SYSTEM SERVICES                        TIME: 23:23:00          06/21/86          PAGE    1

   PARM    TEST ( FULL (LCTP,2,1) )

IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

   LISTCAT  ENTRY MN01.CL001041/CLMR ) ALL

 IDC0924I  DUMP ROUTINE INVOKED AT 'LCTP'

INTERMODULE TRACE: EX01 SA02 LC01 SA02 LC01 SA02 LC01 SA02 LC01 SA02 LC01 SA02 LC01 SA02 LC01 SA02 LC01 SA02 LC01 DP01
INTRAMODULE TRACE: R137 RINN SACL RITM SAFP R199 SADP EX1F EXFS SACL LCIN SAGP SAGP SAGP SAGP SAGP SAGP SAGP SAGP LCTP

   DARGLIST = 00000000 0023803C D3C3F0C1 00000000 01300000 <- Selected fields; Text Processor Argument List

                                                                                          Module that called for dump
IDC0924I  DUMP ROUTINE INVOKED AT 'LCTP'

INTER-MODULE TRACE: DB01 TP01 D801 TP01 DB01 TP01 LC01 DB01 LC01 SAC2 DB01 SA02 DP01 SA02 LC01 DB01 LC01 DB01 LC01 DB01
INTRA-MODULE TRACE: SALD SAGP TP5N TPCC TP2I TP2I TPDB TPDB TPDB TP2N TB2N TB1N LCEN LCBL SACA ZZCA ZZCA LCAL LCWA LCTP

   DARGLIST = 00C00000 002405C5 D3C3F002 00000001 00790000 000A0C2C 0024063E <- Selected fields; Text Processor Argument List

IDC0925I  DUMP 001 PRODUCED AT DUMP POINT 'LCTP'
                 ^
                 |- ID of snap-dump
```

DA6R1033

Figure 27. Example of TEST Option Output (Part 1 of 2)

```
                        .
                        .                                        ID of snap-dump
                        .
JOB AMSLIST         STEP                  TIME 232310      DATE 06/21/86      ID = 001             PAGE 0001

PSW AT ENTRY TO SNAP   071D2000 002033A8

SEGMENT TABLE ORIGIN REGISTER 02018A00

TCB     012C68  RB   0023F270  PIF    00000000 DEB  0024BD88  TIOT  0024F730  CMP   00000000  TRN  00000000
                MSS  00012198  PK/FLG 10010008 FLG  000001F5  LLS   00000000  JLB   00000000  JST  00012068
                FSA  0A24F6F0  TCP    000122F8 TME  000121D8  PIB   80000053  NTC   00000000  OTC  00000000
                LTC  00000000  IQE    000000C0 ECB  00000000  XTCB  00000000  LP/FL 55000000  RESV 00000000
                STAE 0024F7A8  TCT    0024FC18 USER 00000000  NDSP  00000000  MDIDS 00000000  JSCB 0001ED74
                RESV 00000000  RESV   00000000 RESV 00000000  EXT1  00000000  BITS  0000080C  DAR  00000000
                EXT2 00012148  PCB    00012158 GQE  0023F73C  ABP   00235310
        EXT2    GTF  00000000  ST/RCM 00000000


ACTIVE RBS

PRB    24FDF8  NM IDCAMS      SZ/STAB 2C0500D0   USE/EP 002C240C   PSW 071D2000 002033A8   C 000000   WT/LNK 00012068
               CDP   NOTE/LOAD 00200000   MODLNTH EE000A9C

SVRB   23F27C  NM SVC-A05A    SZ/STAB 0C14D062   USE/EP 002FBC00   PSW 07000000 002900E8   C 000273   WT/LNK 0024FDF8
               RG 0-7   00000030    0024EE00    0024B598    0023D7C0    000006BC    00012248    0024F2C0    0024C070
               RG 8-15  0024B598    0024F2C0    60202B38    0024EBC0    00203B37    0024EBC0    00202B54    00000000

JOB PACK AREA
                        .
                        .                                                                                  DA6R1034
                        .
```

Figure 27. Example of TEST Option Output (Part 2 of 2)

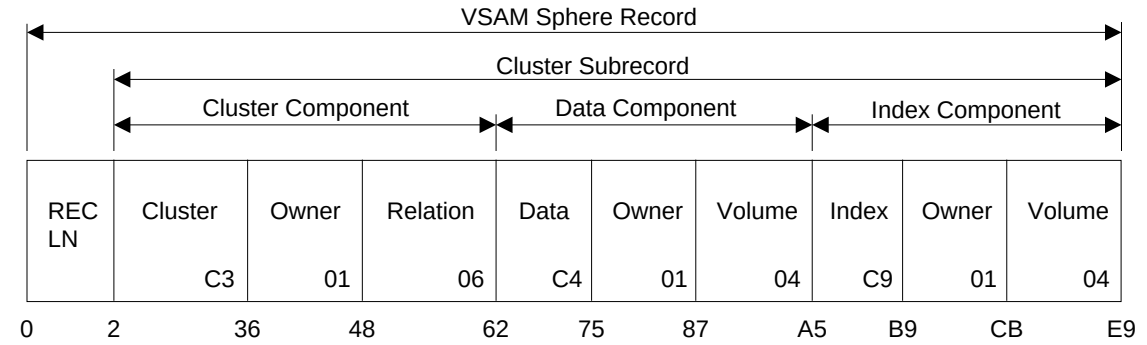
Catalog Management Diagnostic Aids

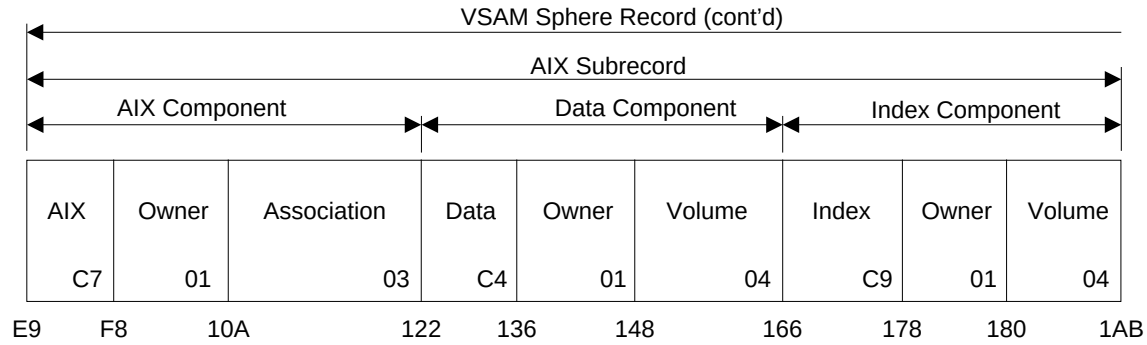
The DIAGNOSE command output is designed to help you diagnose catalog management problems.

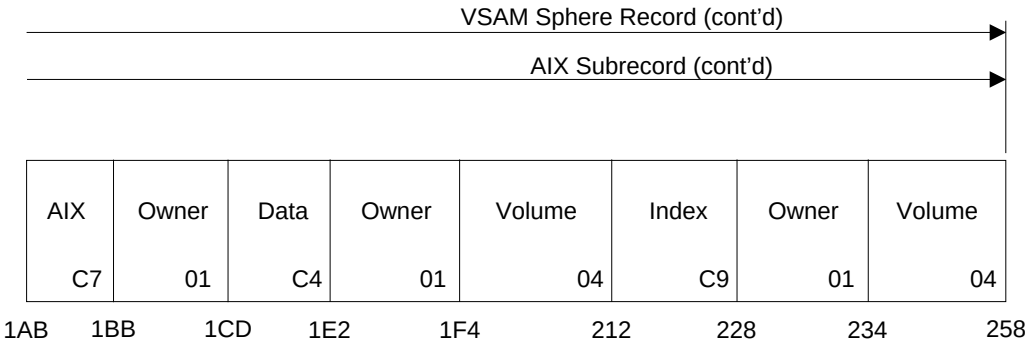
Sample DIAGNOSE Output

Illustrations of catalog records follow, in the format in which they would be displayed after execution of the DIAGNOSE command.

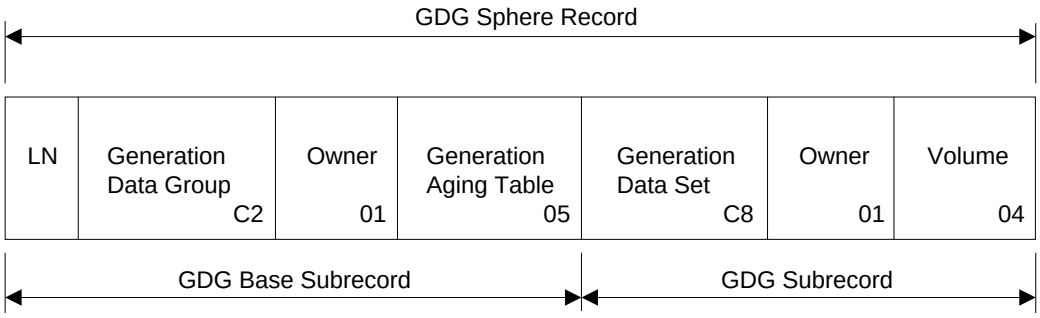
The following illustration shows a VSAM sphere record for a KSDS data set:

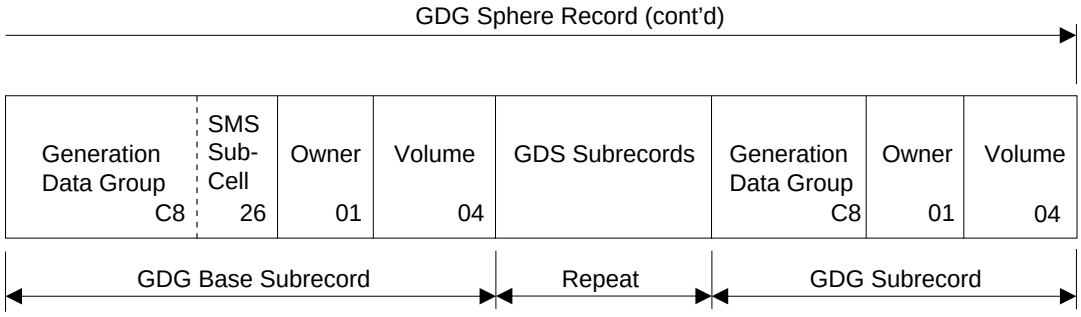




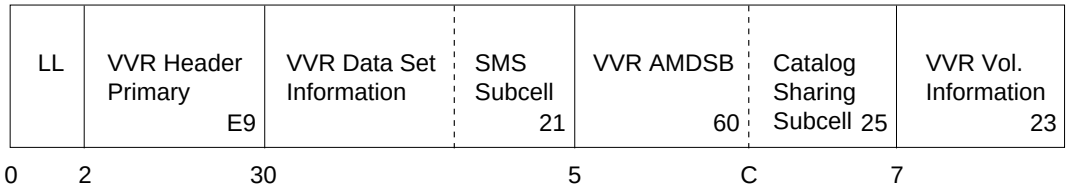


The following illustration shows a GDG sphere record consisting of a base record and 10 subrecords (as indicated in the generation aging table cell):

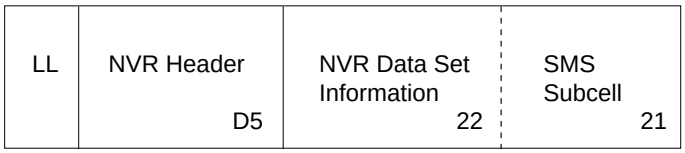




The following illustration shows a VVDS record (primary VVR):



The following illustration shows an NVR record:



VSAM OPEN/CLOSE/End-of-Volume (O/C/EOV) Diagnostic Aids

The following major diagnostic aids are provided for VSAM O/C/EOV:

- A description of the GTF trace facility, which is used to trace VSAM O/C/EOV activity
- A description of failure-related messages that are issued by VSAM O/C/EOV and the function codes that they contain

The diagnostic aids and return codes that are described in this section are also applicable when RLS is active.

Generalized Trace Facility

Use the generalized trace facility (GTF) to obtain diagnostic information during the processing of VSAM Record Management (R/M). If GTF is active and **TRACE=**

(record management tracing) is specified in your DD statement, VSAM GTF trace records, X'FF5', are written to the GTF data set.

VSAM GTF X'F61' Record

The VSAM GTF X'F61' was designed to provide a method for tracing I/O activity for data in a shared resource pool. This GTF record is not supported for VSAM RLS.

Considerations When Requesting X'F61' Records: The following items should be considered when you request X'F61' records:

- GTF must be started with the USRP (or USR) parameter specified with an event identifier of X'F61'.
- GTF X'F61' records are generated only for jobs processed with VSAM LSR or GSR.
- If the ATTACH macro is used, the SZERO=YES parameter must be coded.

Mapping of the X'F61' Record

The X'F61' record is an information record to be used for specialized purposes. Because of its dependence on detailed design and implementation, IBM might change the mapping of the record with new product releases or versions or as a result of service. Table 112 gives detailed information related to the X'F61' record.

Table 112. X'F61' Record Information

Offset	Name	Length	Description
0(X'00')	ASCB	4	ASCB address
4(X'04')	JOBNAME	8	Jobname
12(X'0C')	ASID	2	Current ASID
14(X'0E')	REQ_TYPE	1	Type of request
15(X'0F')	SHR_POOL	1	LSR resource pool ID
16(X'10')	CI	4	Requested CI
20(X'14')		4	Reserved
24(X'18')	FLAGS	1	Processing flags **Bit** — **Meaning When Set** **0** — GSR (LSR if OFF) **1** — Index (data if OFF) **2** — CI found in Hiperspace™ or read from DASD (CI found in address space buffer pool if OFF) **3** — CI found in Hiperspace **4 – 6** — Reserved **7** — Cross-System Share Option 4 (share option 3 if OFF)
25(X'19')	XREG_SHROPT	1	Cross-Region Share Options SHR(1)=X'01'X SHR(2)=X'02'X SHR(3)=X'03'X SHR(4)=X'04'X
26(X'1A')	CISIZE	2	CI size for the component
28(X'1C')	DSNAME	44	VSAM component name

Table 112. X'F61' Record Information (continued)

Offset	Name	Length	Description
72(X'48')	VOLSER	6	Volume serial for the component
78(X'4E')	BUFRSIZE	2	Buffer size
80(X'50')	TRANSID	1	RPL transid
81(X'51')	RECLEN	3	Record length

Interactive Problem Control System (IPCS)

The interactive problem control system (IPCS) provides installations with an interactive, online facility for diagnosing software failures. Using unformatted dumps, IPCS formats and analyzes them to produce reports that you either view at the terminal or print. As you examine the dump of a software failure, IPCS accumulates information about the dump. IPCS uses the information it gathers each time you process the dump.

See *z/OS MVS IPCS User's Guide* for information on how to use IPCS and *z/OS MVS IPCS Commands* for information on the IPCS commands and examples.

See *z/OS MVS IPCS Customization* for product information and programming requirements needed to use IPCS and customize access to IPCS. It gives examples on writing various kinds of exit routines.

IPCS allows you to analyze these types of dumps:
- System dumps, which include:
  - ABEND dumps
  - SVC dumps
  - Stand alone dumps.

  Tip: System dumps might include GTF trace data.
- Live system storage
- External GTF trace information written to direct access or tape storage devices.

Printing GTF Records

You can use the GTFTRACE subcommand of IPCS to format and print GTF records contained in a dump or a trace data set. GTF trace records can be formatted interactively in an IPCS session or in batch mode. You can select the kinds of trace data you want to process by using the appropriate trace data selection keywords.

This example creates a formatted report of the GTF trace records for SVC events:

```
gtftrace svc
```

The following two examples create a formatted report of the GTF trace record for VSAM R/M control blocks.

Interactively in an IPCS session:
1. The IPCS dump source is the trace data set (via DDNAME or DSNAME).
2. Enter the IPCS subcommand:

```
GTFTRACE USR(AM01)
```

When using batch mode, you need to specify the name of the input trace data set and the names of the three output data sets in the JCL as shown:

Input: Trace data set 'SYS1.TRACE'(TRACE DD)

Output:
- IPCS dump directory data set (IPCSDDIR DD)
- Formatted output (IPCSPRNT DD)
- TSO/E messages (SYSTSPRT DD)

IPCSPRNT, the IPCS print file, contains the dump or trace output. The SETDEF PRINT subcommand tells IPCS to direct output to the IPCS print file. The SETDEF TERMINAL subcommand tells IPCS to direct output to the SYSTSPRT file. TSO/E messages and some IPCS messages are directed to SYSTSPRT even if SETDEF PRINT NOTERMINAL was entered.

The IPCSDDIR dump directory data set is a VSAM indexed cluster that IPCS uses to store information. Usually, the dump directory is used internally by IPCS and IPCS formatting routines but there are subcommand interfaces for you to store and delete information. See *z/OS MVS IPCS User's Guide* for more information about the IPCSPRNT, SYSTSPRT, and IPCSDDIR files.

```
//JOBNAME   JOB  ,accounting
//STEP1     EXEC PGM=IKJEFT01,REGION=4096K,DYNAMNBR=50
//IPCSDDIR  DD DSN=IPCSU1.DUMP.DIR.DISP=SHR
//TRACE     DD DSN=SYS1.TRACE,UNIT=293,VOL=SER=338003,DISP=SHR
//IPCSTOC   DD SYSOUT=*
//IPCSPRNT  DD SYSOUT=*
//SYSTSPRT  DD SYSOUT=*
//SYSTSIN   DD *
 IPCS
 SETDEF NOCONFIRM PRINT NOTERMINAL DDNAME(TRACE)
 DROPDUMP
 GTFTRACE USR(AM01)
 DROPDUMP
 END
 /*
```

VSAM OPEN/CLOSE/End-of-Volume Return and Reason Codes

VSAM O/C/E macros return to their caller return codes set in register 15 and reason codes set in the access method control block (ACB) ACBERFLG field (1 byte at ACB + X'31'). VSAM macro return codes and reason codes are in *z/OS DFSMS Macro Instructions for Data Sets*.

VSAM Record-Level Sharing (RLS) Diagnostic Aids

This section explains the diagnostic aids provided for VSAM record level sharing and offers suggestions for isolating different types of problems.

The following diagnostic aids and types of problems are discussed:
- VSAM RLS component trace
- VSAM RLS IPCS processing
- VSAM RLS SMSVSAM abends
- VSAM RLS server recycle and reactivation
- Console dumps
- SMSVSAM initialization errors
- Problems sharing control
- VSAM RLS hang conditions
- VSAM RLS deadlock and timeout problems

When RLS is active, the return codes for VSAM O/C/EOV and record management macros are still applicable. See VSAM OPEN/CLOSE/End-of-Volume Return and

Reason Codes and VSAM Record Management Return and Reason Codes for a list of these return codes and their explanations.

VSAM RLS Component Trace

DFSMSdfp provides a trace service for use with all functions provided by VSAM RLS. When the tracing is activated, a trace table is created in the VSAM RLS address space.

This trace facility is also used by other functions. Specific trace options for other functions are described in their sections.

You can use SYS1.PARMLIB member CTncccxx to specify trace options for the TRACE CT command and to turn off tracing. CTncccxx must exist at IPL for tracing to be available on the system. The default is CTISMS00. For more information on using CTncccxx to specify tracing options, see ″CTncccxx″ in *z/OS MVS Initialization and Tuning Reference*.

Trace also allows external writers (WTR) to capture trace buffers. Capturing trace buffers increases diagnostic capability and decreases the possibility of losing trace data.

Complete syntax and usage information for TRACE can be found in *z/OS MVS System Commands*.

TRACE Command for VSAM RLS Component Trace

The syntax of the TRACE command for tracing the VSAM RLS component is:
TRACE CT,WTRSTART=membername or TRACE CT,WTRSTOP=membername or TRACE CT,ON|nnnK|nnnM|OFF,COMP=SYSSMS,PARM=CTISMSxx-->

PARM=CTISMS*xx* identifies the SYS1.PARMLIB member that contains the tracing options.

The default SYSSMS trace table size is 72KB.

The *nnn*K or *nnn*M keywords can be used to specify trace table sizes of 16–999 KB or 1–2047 MB, respectively.

You are then prompted to specify the trace options to be in effect. You must respond with the REPLY command, using the following syntax: REPLYid ,JOBNAME=SMSVSAM ,ASID=(smsvsam-asidlist.) ,OPTIONS=(namename...),WTR=procname|DISCONNECT ,END-->

Notes:

1. The *id* value is the identification number, 0-99, specified on the prompting message.
2. The REPLY may be continued on multiple lines; the END must be given as the final parameter to identify the end of the REPLY.
3. OPTIONAL (only needed when using CTRACE WRITER) - Create a member in SYS1.PROCLIB to allocate a CTRACE WRITER data set(s). Refer to the *Using an External Writer to Capture Trace Data* section in the *z/OS MVS Initialization and Tuning Reference* for an example procedure. Note that the name of the procedure is the name specified when starting the trace writer. For example, to start the trace writer with a procedure named CTWDASD, issue the following command:

   ```
   TRACE CT,WTRSTART=CTWDASD
   ```

The TRACE and REPLY commands are fully described in *z/OS MVS System Commands*.

Trace Options

The valid options for use with this trace facility are shown in Table 113.

Table 113. Valid Trace Facility Options

Trace Event	Description
ENTRY	All module entries.
EXIT	Module exits with nonzero return codes only.
EXITA	All module exits.
CALL	Equivalent to specifying both ENTRY and EXIT.
UEXIT	Entries and exits of user exits.
RRTN	Entries and exits of recovery routines.
CB	Control block changes.
POST	Usage of certain MVS services.
SPECIAL	Entries and exits of commonly shared functions.
SUSP	Suspension or resumption of a work unit.
CONFIG	Configuration changes.
RLS1	Special purpose trace event for use under specific IBM programming support direction.
RLS2	Special purpose trace event for use under specific IBM programming support direction.
RLS3	Special purpose trace event for use under specific IBM programming support direction.
RLS4	Special purpose trace event for use under specific IBM programming support direction.
RLS5	Special purpose trace event for use under specific IBM programming support direction.
DLKPD	Deadlock/timeout/retained-lock problem determination trace event.
COMP=(SMSVSAM)	Specifies all the functional areas of VSAM RLS component.
SUBCOMP=(*component- list*)	Specifies specific subcomponents within the VSAM RLS component.

The following terms are used to designate the subcomponents of VSAM RLS:

BLC	SHM	SQM	VSS
CERS	SLCH	VOC	
SCM	SMLS	VQUI	
SHC	SMPM	VRM	

Notes:

1. You may also specify **ALL** rather than list the **COMP** or **SUBCOMP** keywords.

   COMP=(SMSVSAM) is equivalent to specifying **ALL** subcomponents. However, in most cases you would only trace the specific subcomponents.
2. The **SUBCOMP** keyword can be specified without the **COMP** keyword.
3. Any of the options listed above except for **ALL**, **COMP**, and **SUBCOMP** can be turned off by prefixing them with **NO** (for example, **NOVOC**, **NOSLCH**).

The following command will turn off the trace facility:

TRACE CT,OFF,COMP=SYSSMS

Here are some examples of typical replies to activating SMS trace:

- To trace record management entry/exit

  ```
  R XX,OPTIONS=(ENTRY,EXITA,SUBCOMP=(VRM)),WTR=CTWDASD,END
  ```

- To trace deadlock/timeout problem determination information

  ```
  R XX,OPTIONS=(DLKPD,SUBCOMP=(VRM)),WTR=CTWDASD,END
  ```

- If special trace code were provided for RLS1, the following would be recommended by IBM programming support to utilize that special trace code:

  ```
  R XX,OPTIONS=(RLS1,SUBCOMP=(VRM)),WTR=CTWDASD,END
  ```

Formatting the VSAM RLS Component Trace Table

There is no mechanism to flush trace buffers in storage to the trace writer data sets. So, in addition to having trace writer data sets available, a storage dump of the SMSVSAM address space and data space is required. If the problem being traced does not result in a dump the MVS DUMP command can be used to produce an SVCDUMP-type dump containing the trace buffer data. See *z/OS MVS IPCS Commands* for a complete description of this service.

It is possible with a sufficiently large trace table (8M recommended, use 16M if possible) that there is no data in the trace writer data sets.

The IPCS CTRACE command is used to format and view the information in the component trace writer data sets and the trace buffers in storage:

```
CTRACE COMP(SYSSMS) FULL LOCAL
```

SMSVSAM Abends

Abends in the SMSVSAM address space can be identified by COMPID=DF122. The dump title (also available in syslog) contains:

- COMPID=DF122
- CSECT and offset of the instruction in error
- Compile date and maintenance level of the CSECT
- The abend code
- Return and reason codes

When the failing CSECT cannot be identified, **????** will appear in the dump title for the CSECT name.

In some cases you can identify the failing CSECT using IPCS option 6, ST command to find::

- The failing PSW
- The general purpose registers and ARs
- The calling sequence

  Use the panel provided by the IPCS IGWVIPCS to run option Q to find the calling sequence.

  Note: This procedure might not provide information for all abends.

Activating the SMSVSAM Address Space

The SMSVSAM address space automatically starts at IPL if the RLSINIT (YES) keyword is specified in the IGDSMS*xx* member of SYS1.PARMLIB. You can also start the SMSVSAM address space after IPL by issuing the following command from the MVS console:

```
V SMS,SMSVSAM,ACTIVE
```

Terminating the SMSVSAM Address Space

To terminate the SMSVSAM address space, issue the following command from the MVS console:

```
V SMS,SMSVSAM,TERMINATESERVER
```

Note: Use the VSMS,SMSVSAM,TERMINATESERVER command before you partition a system out of the XCF Sysplex. Failure to do so can result in unexpected abends in the SMSVSAM address space.

To terminate and automatically restart the SMSVSAM address space, issue the following command from the MVS console:

```
FORCE SMSVSAM,ARM
```

Note: You should use the FORCE SMSVSAM,ARM command if the SMSVSAM address space did not teminate successfully with the V SMS,SMSVSAM,TERMINATESERVER command. In this case, make a console dump of the hung SMSVSAM address before you issue the FORCE SMSVSAM,ARM command, and report the problem to your IBM representative.

The SMSVSAM address space automatically terminates and restarts after detecting fatal internal errors. If this situation occurs, report the failure, along with any dumps that were produced, to your IBM representative.

After any six consecutive restarts of the SMSVSAM address space, the system issues message IGW418D, which prompts the MVS operator for a reply to cancel or to continue with SMSVSAM initialization.

Console Dumps

Console dumps that are generated when VSAM RLS is running will contain the SMSVSAM address space, its data spaces, and the catalog address space.

For example:

```
DUMP COMM=(EXAMPLE OF SMSVSAM DUMP)
R nn,JOBNAME=(SMSVSAM,CATALOG),CONT
R nn,DSPNAME=('SMSVSAM'.SMSVSAM,'SMSVSAM'.MMFSTUFF),END
```

SMSVSAM Initialization Errors

Message IGW414I is issued when the SMSVSAM server initializes successfully.

When initialization is not successful, use the D SMS,SMSVSAM command to gather data that can be used to determine the cause of failure.

Table 114 on page 487 can be used as a guide to what actions to take for specific initialization problems.

Table 114. Initialization Errors

Symptom	Action
Server cannot successfully connect to the lock structure, IGWLOCK00.	Save the output from the following command: `D XCF,STR,STRNAME=IGWLOCK00`
Connectivity is not available.	If you have another Coupling Facility available, initiate a rebuild to cause the lock structure to become resident in the other lock structure. If a rebuild cannot be done, SMSVSAM will not initialize until the connectivity problem is resolved.
Waiting for replies from other systems.	1. Dump the SMSVSAM address spaces from all the systems in the Parallel Sysplex. 2. Recycle SMSVSAM address spaces on the systems that have not replied.
Structure does not exist.	If possible, modify your policy to allow the lock structure to be defined. The SMSVSAM address spaces should attempt to reconnect when they are notified that a new policy is in place. If they do not reconnect, you might want to recycle all the servers.
Other problems that occur during initialization.	Take a console dump of the SMSVSAM address space on all systems and contact IBM's Customer Support for problem diagnosis.

Sharing Control Problems

Problems in sharing control are usually indicated by abends in the IGWXSxxx modules. Or by incorrect output of the SHCDS command.

Contact IBM's Customer Support when you have the following documentation:
- DFSMSdss dump of the active sharing control data sets
- Abend dumps
- Syslog
- SHCDS command output
- Console dump (only when the SHCDS command output is incorrect)

The following is an example job for obtaining a DFSMSdss dump of the sharing control active data set. Change the OUT and DUMP statements as appropriate.

```
//DMPSHCDS JOB ...
//* JOB TO DSS DUMP SHARING CONTROL ACTIVE DATA SET. CHANGE JOB
//* AND OUT DD STATEMENTS AS APPROPRIATE.
//DSSDMP  EXEC  PGM=ADRDSSU,REGION=4M
//SYSPRINT DD SYSOUT=H
//OUT DD DSN=DUMP.SHCDS,DISP=(,KEEP),UNIT=CART,
// VOL=SER=CART01,LABEL=(1,SL)
//SYSIN    DD  *
  DUMP   DS(INCLUDE(SYS1.DFPSHCDS.ACTIVE.SPLXPK))  SHARE -
         TOLERATE(ENQFAILURE) -
         OUTDD(OUT)
```

VSAM RLS Hang Conditions

Should a hang condition occur, you can generate dumps of the following spaces from all systems:

- SMSVSAM address space
- SMSVSAM dataspace
- Catalog address space

The following is an example of how to generate a dump SYS01 for systems SYS01, SYS02, and SYS03:

```
SYS01   DUMP COMM=(MULTI SYSTEM DUMP)
SYS01   R nn,JOBNAME=(SMSVSAM,CATALOG),CONT
SYS01   R nn,REMOTE=(SYSLIST=(SYS02,SYS03)),CONT
SYS01   R nn,DSPNAME=('SMSVSAM'.SMSVSAM,'SMSVSAM'.MMFSTUFF),END
SYS01   R nn,DSPNAME=('SMSVSAM'.*),END
```

To get all data spaces, use .* as the last line of the example shows.

You might need to dump one or more of the users' address spaces if it appears that they are involved in the hang condition.

IBM's Customer Support might request a report of the active threads. Invoke IPCS IGWVIPCS and run options 1 and 2 to obtain a summary of TCBs running in the SMSVSAM address space and the users' requests that were active at the time of the failure.

VSAM RLS Deadlock and Timeout Problems

Deadlock or timeout errors in SMSVSAM will cause CICS to issue messages DFHFC0164, DFHFC0165, DFHFC0166, and DFHFC0167. They should provide enough information to determine the cause of the problem.

If additional information is needed, the CICS FC level 2 trace will provide the VPDI (mapped by IFGVPDI). The VPDI is the source of the information provided in the CICS messages listed above. The SMSVSAM trace OPTIONS=DLKPD can be used to trace the VPDI.

Note: The information returned in the VPDI can be out of date. There is a lag between the time the problems are detected and the request fails. For a timeout, information on the blocking holder might not be returned because that request completed before the information could be extracted.

VSAM Record-Level Sharing Return and Reason Codes

This section describes the return codes from SMSVSAM, SMPM_CFPurge, and SMPM_CFQuery.

Return Codes from SMSVSAM

The SMSVSAM address space is the server for VSAM RLS.

The following return and reason codes from SMSVSAM might appear in messages issued by CICS, CICSVR, and DFSMSdss. They might also include ERRDATA, that can be used to explain the return and reason code information.

For return codes from VSAM O/C/EOV and record management macros, see "VSAM OPEN/CLOSE/End-of-Volume Return and Reason Codes" on page 482 and "VSAM Record Management Return and Reason Codes" on page 504.

Contact IBM Customer Support for errors with return code DEC 36 (X'24'). A system dump might have been taken.

Return code 4 is given for information errors.

Return code 8 can indicate an error using the SMSVSAM interfaces (application logic error) or setup problems (such as missing RACF authority).

Server-not-available type errors indicate that the SMSVSAM server was not active at the time the request was made, and are not an error condition.

Table 115. SMSVSAM Return and Reason Codes

Return Code	Reason Code	Explanation
4	X'60EF0008'	There are no locks to bind or unbind.
4	X'61FF0000'	The request is successful, but non-RLS mode is established.
4	X'61FF0011'	The request is successful, RLS mode established. Other connectors constant.
4	X'61FF0012'	The request is successful, RLS mode established. Lost locks constant.
4	X'61FF0013'	The request is successful, RLS mode established. Retained locks constant.
4	X'61FF0014'	The request is successful, No quiesce operation is performed. System is in local mode (as opposed to sysplex mode). Local mode constant.
8	X'60FF0001'	Request to connect a subsystem that is already active in the Parallel Sysplex.
8	X'60FF0002'	Request to connect an online application, which is already connected as a batch application.
8	X'60FF0003'	Request to connect/disconnect an online application, subsystem name was not passed.
8	X'60FF0004'	Request to connect or disconnect an online application, and the subsystem name has an invalid format.
8	X'60FF0005'	Request to connect an online application, and no quiesce exit was provided.
8	X'60FF0006'	Application is not authorized to the RACF SUBSYSNM class.
8	X'60FF0007'	Unable to obtain virtual storage.
8	X'60FF0008'	The subsystem name specified is not registered.
8	X'60FF0009'	Attempt to disconnect an online application, but the application has OPEN RLS ACBs.
8	X'60FF000A'	Attempt to connect or disconnect an online application, and a control ACB was not passed.
8	X'60FF000B'	SMSVSAM server is not available.
8	X'60FF000C'	Connect or Disconnect request issued in invalid mode (e.g. SRB mode instead of TCB mode).
8	X'60FF000D'	Application has been canceled.
8	X'61FF0001'	Catalog Locate Failure for the specified sphere. ERRDATA contains additional information.

Table 115. SMSVSAM Return and Reason Codes (continued)

Return Code	Reason Code	Explanation
8	X'61FF0002'	The specification is not an SMSVSAM data set.
8	X'61FF0003'	A Quiesce operation is already in progress for this sphere. Would occur if two DFSMSdss jobs concurrently tried to dump the same sphere.
8	X'61FF0004'	A batch application has the sphere open for output and this is an attempt to take a non BWO copy.
8	X'61FF0005'	Request was not issued in supervisor state.
8	X'61FF0006'	LOG(ALL) was not specified for the data set.
8	X'61FF0007'	The request to copy/dump has been canceled.
8	X'61FF0008'	The request type is not supported.
8	X'61FF0009'	Virtual storage was not available.
8	X'61FF000A'	SMSVSAM server was not available.
8	X'61FF000B'	The copy is not valid, since the SMSVSAM server failed during the copy.
8	X'61FF000C'	The task which issued the request already has an active non-RLS open for the sphere.
8	X'61FF000D'	Request to terminate the operation, but there is no operation in effect.
8	X'61FF000E'	Request mismatch.
8	X'61FF000F'	Request to terminate the operation, but the operation is being canceled.
8	X'61FF0010'	There is a non-RLS quiesce active within this address space. The situation across the sysplex is unknown.
8	X'60EF0001'	Error accessing sharing control. ERRDATA has qualifying information.
8	X'60EF0002'	Catalog locate failed. ERRDATA contains qualifying information.
8	X'60EF0003'	The requestor does not have update authority to the sphere.
8	X'60EF0004'	The request passed an invalid ACB.
8	X'60EF0005'	The request did not specify an SMS VSAM data set.
8	X'60EF0006'	The request was issued in an invalid mode.
8	X'60EF0009'	No virtual storage.
8	X'60EF000A'	Invalid request parameter list.
8	X'60EF000B'	Open for output failed. ERRDATA contains qualifying information.
8	X'60EF000C'	Dynamic allocation failure. ERRDATA contains qualifying information.
8	X'60EF000D'	Dynamic unallocation failure. ERRDATA contains qualifying information.
8	X'60EF000E'	Open failure. ERRDATA contains qualifying information.

Table 115. SMSVSAM Return and Reason Codes (continued)

Return Code	Reason Code	Explanation
8	X'60EF000F'	Close failure. ERRDATA contains qualifying information.
8	X'60EF0010'	Incorrect parameter list passed.
8	X'60EF0011'	Number of clusters in original sphere and restored sphere do not match.
8	X'60EF0012'	Recovery required expected, but was not set. ERRDATA contains qualifying information.
8	X'60EF0013'	Request issued in invalid mode.
8	X'60EF0014'	Data set not found.
8	X'60EF0015'	An RLS cell does not exist. The operation cannot be performed.
36	Varies	Severe error. ERRDATA contains qualifying information.

Return Codes from SMPM_CFPurge

The return and reason codes shown in 116 can be returned from the SMPM_CFPurge interface.

Table 116. SMPM_CFPurge Return and Reason Codes

Return Code	Reason Code	Explanation
4	X'64xxFA10'	Request completed successfully. Retained locks were not purged as part of the request, and recovery is required.
4	X'64xxFA12'	Request completed successfully. Retained locks were purged as part of the request, and recovery is required.
4	X'64xxFA1E'	Request completed successfully. No purge was necessary because the system is not using VSAM RLS.
4	X'64xxFA1F'	Request completed successfully. No purge was necessary because the system is not running in Parallel Sysplex mode.
4	X'64xxFA20'	Request completed successfully. Retained locks were purged as part of the request.
8	X'64xxFA0E'	The base cluster name passed was not found in the catalog. No CF Cache information is retained for this sphere.
8	X'64xxFA0F'	An open sphere prevented the operation from completing.
8	X'64xxFA11'	Locks were held and the request specified SMPM_Fail.
36	Varies.	Severe error. ERRDATA contains qualifying information.

Return Codes from SMPM_CFQuery

The return and reason codes shown in 117 can be returned from the SMPM_CFQuery interface.

Table 117. SMPM_CFQuery Return and Reason Codes

Return Code	Reason Code	Explanation
4	X'64xxFA1E'	Request completed successfully. No query was necessary because the system is not using VSAM RLS.
4	X'64xxFA1F'	Request completed successfully. No query was necessary because the system is not running in Parallel Sysplex mode.
8	X'64xxFA0E'	The base cluster name passed was not found in the catalog. No CF Cache information is retained for this sphere.
36	Varies.	Severe error. ERRDATA contains qualifying information.

VSAM Record Management (R/M) Diagnostic Aids

The following major diagnostic aids are provided for record management:

- Explanations of how to analyze exclusive control errors and physical I/O errors
- Explanations of how to analyze ABEND0CX errors
- An explanation of how to diagnose and recover from damaged indexes and damaged data control intervals
- A description of the record management trace facility
- A description of record management return codes

Control Block Information

VSAM R/M uses two types of control blocks:

- User control blocks (RPL, ACB, and EXLST)
- Control blocks built for record management by VSAM OPEN (all other control blocks)

The RPL's RPLFDBWD field (4 bytes at offset X'0C') provide valuable failure-related codes. A description of the return codes and reason codes can be found in *z/OS DFSMS Macro Instructions for Data Sets*.

The VSAM SNAP dump facility provides hexadecimal listings of some control blocks and data areas. To obtain a SNAP dump, perform one of the following tasks:

- Code the SNAP macro with the SDATA=CB option (described in *z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN*, *z/OS MVS Programming: Authorized Assembler Services Reference ENF-IXG*, *z/OS MVS Programming: Authorized Assembler Services Reference LLA-SDU*, *z/OS MVS Programming: Authorized Assembler Services Reference SET-WTO*).
- Specify this option to ABDUMP via the CHNGDUMP operator command or the IEAABD*nn* or IEADMP*nn* members of SYS1.PARMLIB.

The SNAP dump facility is only available at SYSABEND time.

UPAD Exit or WAITX

If a UPAD exit is provided (RPLWAITX flag - bit 5 at RPL +X'29') for a deferred request, VSAM takes the UPAD exit for event completion. When the request can be restarted, VSAM takes the UPAD exit for a POST if in cross-memory or SRB mode. If not in cross-memory or SRB mode, VSAM issues a POST SVC. The user-provided UPAD exit must return control to the posted request in VSAM at the

point just following where control was given to the UPAD WAIT exit. For more information about UPAD and WAITX, see *z/OS DFSMS: Using Data Sets*.

VSAM RLS does not support UPAD exits.

RLSWAIT Exit

For synchronous VSAM RLS requests only, by specifying RPLWAITX the application can provide an RLSWAIT exit that issues WAIT for the completion of the request. When the request is completed, VSAM POSTs the request. The RLSWAIT exit ensures the ECB is marked POSTed before returning to VSAM. Note that RLSWAIT is entered only for a request completion wait, never for a resource or I/O completion wait. See *z/OS DFSMS: Using Data Sets* for more details.

Error Conditions

VSAM R/M returns to its caller return codes in register 15 and feedback words set in the request parameter list (RPL + X'0C'). See the chapter on macro instruction return codes and reason codes in *z/OS DFSMS Macro Instructions for Data Sets*, for a description of all such codes.

When diagnosing a problem, record the 4-byte RPL feedback word on the keyword worksheet in *z/OS DFSMSdfp Diagnosis Guide*. Append the 4-byte field to the keyword prefix **RC**.

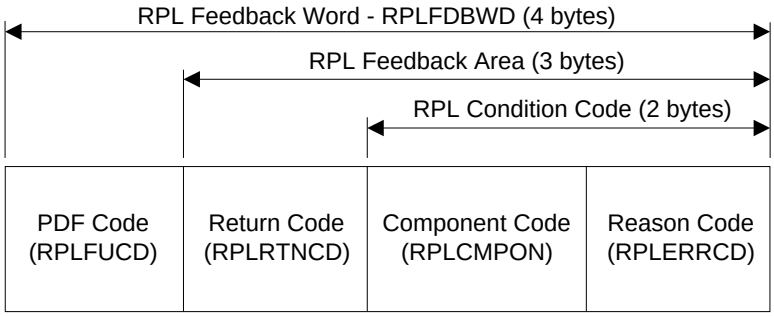
Example: The RPL feedback word is **2D08009C**; specify this modifier keyword as shown:

```
RC2D08009C
```

The IBM Support Center can use these codes to determine which module is in control and the reason the error occurred. You can use the codes in the RPL feedback word with other portions of the VSAM record management diagnostic aids to obtain additional failure-related information. The IBM Support Center may request an additional dump using the SLIP service aid to trap the error code when it is being set by a module.

The RPL feedback word is located at offset 12 (X'0C') in the RPL and contains the following information.

RPL	Length	Description
12(X'0C')	1	Problem determination function (PDF) code. This helps identify the module that detected the error.
13(X'0D')	1	RPL return code. This code is returned in register 15.
14(X'0E')	1	Component code. This code identifies the component being processed when the error occurred.
15(X'0F')	1	Reason code. This code, when paired with the return code in the second byte above, identifies the reason for an error.



When analyzing a problem involving record management, consider the following items:

- What kind of application program was running when the error was encountered?
  - User application
  - IMS application
  - CICS application
  - CICS running under IMS.
  - System function (SMF/catalog)
  - Other application
- What was the application program attempting to do when the error occurred?
- What were the request options? (Get/Put, Dir/Seq, Syn/Asy, and so forth).
- Where, within record management, was the error encountered?
  - Abend type-of-failure: use failing PSW instruction address.
  - Wait type-of-failure: use information from the RB that issued the Wait SVC.
  - RPL feedback error (RPL +X'0C'): See the chapter describing VSAM macro instruction return codes and reason codes in *z/OS DFSMS Macro Instructions for Data Sets*.
- What was record management attempting to do when the error occurred?
  - If the failure-related documentation is produced while record management is processing, you can use the PLH R14 push-down stack to identify the function being performed when the failure occurred.
- Is the data set damaged?
  - You may use the EXAMINE command to analyze a data set's integrity (whether it is damaged or not). If a data set is damaged, EXAMINE provides details about the nature of the damage. See the chapter describing functional command format in *z/OS DFSMS Access Method Services* for details about the EXAMINE command.

Exclusive Control Error Analysis

If a second request requires a record that is in a buffer in which a previous request is positioned, the second request may fail with an exclusive control error (RPLERRCD field at RPL +X'0F' = X'14'). If a message area is provided in the second RPL, VSAM record management returns the address of the RPL that has the position of the resource in the message area of the second RPL. You are responsible for relinquishing position in the initial RPL before redriving the second RPL.

An exclusive control conflict may also arise if an initial request causes a CI or CA split. A second request may receive RPLERRCD = X'14' as just mentioned, if it needs a record in the same CI or CA. You should either wait for completion of the

request whose RPL is pointed to by the second RPL's message area, or issue "ENDREQ" against the RPL pointed to by the second RPL's message area before redriving the second request.

ABEND0CX Error Analysis

The dump analysis procedure used for a record management ABEND0CX is essentially the same as that used for other DFSMS components. First, locate the registers at the time of the abend and the address at which the abend occurred. Using a listing of the failing module, determine the reason for the abend. You can use the PLH R14 push-down stack to determine the flow of the request through record management.

Damaged Data Sets (Non-RLS Access)

If multiple systems share a data set and you fail to run an explicit **VERIFY** before opening the data set for output from either system (distinct from the implicit **VERIFY** initiated by VSAM Open), you may cause data set damage.

Determining Which Data Set Might Be Damaged

If you have a dump resulting from an ABEND0C4 in load module IDA019L1, or a console dump of a wait or loop with the PSW pointing into load module IDA019L1, then you can find the data set name (DSN) in the area pointed to by register 3. In most cases, Register 3 points to the AMB, and the data component name appears at offset X'88' into the AMB.

If you suspect a wait or loop in record management and the registers do not point into load module IDA019L1 (indeed, the symptoms of the dump seem to indicate a problem external to record management, for example, CICS waits), you can locate the AMB by using the RPL. Figure 28 shows how to do this:

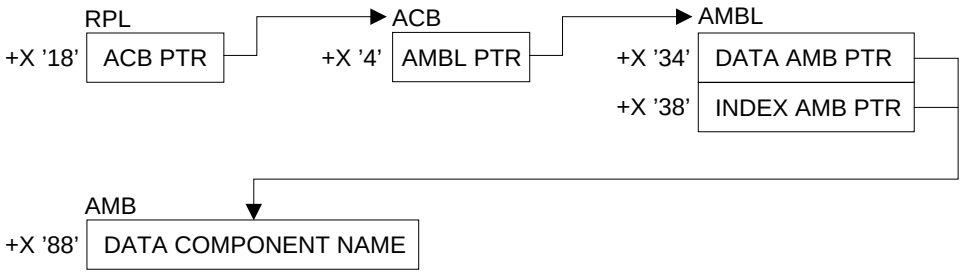


Figure 28. VSAM Record Management—How to Find a Damaged Data Set

Note: Because of its dependency on design and implementation, the control block offsets may change in new product releases or versions as a result of service.

RLS has a different structure. The user's RPL at RPL+4 points to another RPL in the VSAM server address space. The ACB, AMBL, and AMB are all in the VSAM server address space.

Damaged Indexes

If the index component of a key-sequenced data set (KSDS) is damaged, the system may indicate the problem in any of several different ways. These include:

- Abends
- Loops
- Waits
- Missing records
- Duplicate records
- Out-of-sequence records

118 lists typical symptoms of a damaged data set when IDCAMS detects the damage.

- Messages

Table 118. IDCAMS-Detected Messages

Message ID	Condition
MSGIDC3302I	Action error.
MSGIDC3308I	Duplicate record.
MSGIDC3314I	Out of sequence record, missing records, duplicate records, no record found.
MSGIDC3351I	VSAM I/O error RC156 or RC24 or RC32.
MSGIDC3350I	No record found (NORECF) or incorrect length.
MSGIEC070I	RC32 RC202 or RC104 RC203.
MSGIEA000I	Command Reject (CMD REJ).

- ABEND0C4 in any of the following VSAM modules:

```
IDA019RC      IDA019RE     IDA019RG      IDA019RH
IDA019RI      IDA019RJ     IDA019RN      IDA019RW
```

- Loops:
  - Loops issuing SVC121 (SVC X'79') or Start I/O (SIO)
  - Loops in any of the following VSAM modules:

```
IDA019RA      IDA019RB     IDA019RC      IDA019RE
IDA019RH      IDA019RI     IDA019RJ      IDA019RN
IDA019RW      IDA019R2     IDA019R3 (IDAM19R3)
```

- Waits issued from IDA019RZ
- Waits with the TCB structure indicating a wait in IDAM19R3 (IDA019R3)
- RPL feedback word (RPLFDBWD) as shown:

```
2D08009C      9208009C     A608009C      A708009C
9108009C      D808009C     E008009C      D708009C
2A080020      2B080020     2C080020      DB080020
DF080020
```

- RPL error code, (RPLERRCD) as shown:

```
RC32   (Return Code x'20')
RC156  (Return Code x'9C')
```

Recovering from Index Damage

1. Use the access method services REPRO command to copy the data component of the KSDS. Specify the data component name (not the cluster name) in the REPRO *INFILE* parameter. See the chapter describing functional command format in *z/OS DFSMS Access Method Services* for details about the REPRO command.
2. Use a sort utility to sort the copied file. This eliminates any duplicate keys that might exist in the file.
3. DELETE and re-DEFINE the damaged cluster.
4. REPRO from the sorted file to the newly defined cluster. Record management rebuilds the index component.

   Notes:

   a. This method does not work for a catalog or for a keyed cluster containing spanned records.

   b. Processing performed against the data set after initial index damage occurs will corrupt the data set content. This method is an attempt to recover as much as possible of the data set. If a backup copy of the data set exists, it may contain more desirable data set content than available by this method.

Damaged Data Control Intervals

Another form of data set damage can occur to KSDS, ESDS, and RRDS data sets. A control interval (CI) may become damaged. This kind of damage may cause several symptoms, including RC156 (X'9C'), and I/O failures such as incorrect length or no record found.

Recovering from Data CI Damage

1. REPRO. the data component of the cluster. Specify the data component name (not the cluster name) in the REPRO *INFILE* parameter.
2. This repro will fail; however, you will have copied all the available data that precedes the damaged portion of the data set. The high-used RBA value in the catalog entry for the data set that you specify in the REPRO *OUTFILE* parameter indicates how many bytes were successfully recovered before the damage was encountered. To recover valid data following the damaged CI, add the CISIZE (from the LISTCAT of the damaged data set) to the high-used RBA of the OUTFILE. Copy from the damaged data set data component using the REPRO *FROM* parameter. Copy *FROM* the high-used RBA + CISIZE, effectively skipping the damaged CI.
3. Continue this process, skipping damaged CIs until you have retrieved all valid data.

   Note: The extent of the damage affects how much data you can successfully recover. If a backup copy of the data set exists, it may contain more valid data than the data set that results from this recovery method.

Physical I/O Errors

The IBM Support Center may request a CCWTRACE of I/O activity to examine the I/O to the DASD device. Do this by invoking GTF with the CCW, IO, IOSB, and SSCH options. Use the trace output to examine the data sent to and received from the DASD device by the VSAM block processor. See *z/OS MVS Diagnosis: Tools and Service Aids* for details about using CCWTRACE.

VSAM Record Management Trace Facility (Non-RLS Access)

Use the VSAM record management trace facility (R/M trace) to record VSAM record management control blocks while VSAM is processing. GTF must also be active for the R/M trace function to write records to the trace data set. See “Generalized Trace Facility” on page 479 for additional information on using GTF for tracing in VSAM.

Use **IPCS** to print the GTF trace records. Because a trace work area for the data set being traced is obtained in private low storage at the time the data set is opened, activating R/M trace may cause some storage overload.

Note: The VSAM Record Management Trace is a tool. Overlaying or releasing VSAM’s control blocks while the trace is active may cause unpredictable results, such as, invalid trace information and various abends.

When to Use the Record Management Trace Facility

Use the R/M trace to perform these tasks:

- Capture data when a problem occurs.

  Problems include incorrect data in a data set, missing records, incorrect control block information, and program checks because of incorrect data and/or fields in VSAM R/M control blocks.
- Capture VSAM R/M control blocks **before an error code is passed back to the caller**.

  VSAM R/M control blocks are captured before the calling program can:
  - Erase or overwrite the VSAM data
  - Abnormally terminate
  - Close the data set, freeing VSAM R/M control blocks

Starting the Record Management Trace Function

You must take the following steps to use trace:

1. Activate GTF on your system. When you start GTF, specify **USR** or **USRP** with an **AM01** event identifier, which is X'FF5'. If neither USR or USRP is specified, GTF ignores the data passed to it by trace.
2. Place an **AMP=**('TRACE=(subparameters)') on the DD statement of the data set that you want to trace.

The **AMP=('TRACE=(**subparameters**)')** specifies to VSAM that you want to trace the control blocks associated with the data set identified by the DD statement. For complete syntax information on the AMP parameter, see the section on coding the DD statement in *z/OS MVS JCL Reference*.

The following are the valid subparameters for trace, including their brief descriptions.

- **HOOK=(**n,n,...**)** Specifies where in the VSAM R/M code tracing is to occur. The default is HOOK=(1). HOOK is an optional subparameter.

  119 lists the predefined trace point IDs, their associated modules, and their functions.

Table 119. Predefined Trace IDs, Modules, and Functions

TRACE POINT ID	Module	Description
0000	IDA019R1	Entry to VSAM.
0001	IDA019R1	Exit from VSAM.
0002	IDAM19R3	Prior to SVC 121 for writes of CIs (no reads).
0003	IDA019RZ	After I/O, wait for CI reads and writes.
0004	IDA019SE	Prior to call to EOV (SVC 55).
0005	IDA019SE	After return from EOV.
0006	IDA019RE	Start of a CI split.
0007	IDA019RE	After completion of a CI split.
0008	IDA019R3	All I/O occurring during a CI split.
0009	IDA019RF	Start of a CA split.
0010	IDA019RF	After completion of a CA split.
0011	IDA019R3	All I/O occurring during a CA split.
0012	IDA019RJ	Prior to index CI split (IDA019RJ entry).
0013	IDA019RI	After call to IDA019RJ (index split).
0014	IDA019RU	After completion of an upgrade request.
0015	IDA019RW, IDA019SY	Shared resources—after I/O, no errors.
	IDA019RZ	Non-shared resources—after I/O, no errors.
0016	IDA019RW, IDA019SY	Shared resources—after I/O, error occurred.
	IDA019RZ	Non-shared resources—after I/O, error occurred.
0017	IDA019RP	After return from JRNAD exit.
0018	IDA019S7	Before SVC 109 call to update the VSI block.
0019	IDA019S7	Before control blocks are updated from VSI.
0020	IDA019R4	Before data record is compressed.
0021	IDA019R4	After data record is compressed.
0022	IDA019R4	Before data record is decompressed.
0023	IDA019R4	After data record is decompressed.
0024	IDAM19R3	Prior to SVC 121 for reads of CIS.
0025 through 0099		Reserved.
0100 through 0255		User trace points.

Notes:

1. There is no limit to the number of trace points you can specify to trace.
2. The HOOK subparameter must be enclosed in parentheses, even if only one trace point ID is specified (for example, HOOK=(1)).

- **ECODE=ANY|**codenumber Limits tracing. When used, tracing occurs only if an error code is being returned to the caller. If ANY is specified, tracing is performed for any nonzero return code. If codenumber (a specific error code) is provided, tracing occurs only if the RPLFDBK code matches that error code.

ECODE is an optional subparameter. If ECODE is not used, the RPLFDBK code will not determine if tracing is to occur.

When an error code (codenumber) is given, it must be a positive decimal number. For example, ECODE=12 causes tracing when one of the following situations occurs:
– A buffer needs to be written
– An attempt was made to store a record out of ascending order sequence
– A physical read error occurred for an index component sequence set.

The three error situations are indicated by the RPLFDBK code of X'0C' and the content of register 15 which is 0, 8, or 12, respectively.

Note: If the user's LERAD or SYNAD exit routine resets the return code before VSAM returns to the caller, this exit routine may fail. Its failure depends on which trace point is active, and when the call to the user's exit routine is made. See *z/OS DFSMS Installation Exits* for more information on user exit routines.

- **KEY=**keydata|lowRBA-highRBA Limits tracing. When used, tracing only occurs if the record key matches keydata, or if the record's RBA value is within the range of the lowRBA and highRBA values. Keydata is the EBCDIC representation of the whole key or the first few characters of a range of keys. KEY is an optional subparameter.

  If **KEY=**keydata is specified, the keydata may be any length up to 44 bytes. The keydata value does not need to be the same length as the record's key length. The shorter key length is used to determine the amount of bytes to be compared; this allows you to use generic key values and specify a range of keys.

  If **KEY=**lowRBA-highRBA is specified, tracing occurs only if the RBA of the record being processed is within the range of lowRBA and highRBA values.

  PARM1, byte 5 bit 5 determines the value of the KEY. If this bit is 0, the KEY field contains a key value; if this bit is 1, the KEY field contains an 8-byte lowRBA value, a dash, and an 8-byte highRBA value.

  Note: The KEY subparameter must not contain quotes, commas, or parentheses.

- **PARM1=**trace options

  PARM1 controls the tracing of any user-opened data sets.

  The PARM1 subparameter specifies which VSAM Record Management data areas are to be traced. It controls the tracing of the user-opened data sets.

  PARM1 is required. If TRACE is specified without any subparameters, only VSAM Open/Close/EOV GTF records are built.

  For an extended format data set, the CPA is not a valid control block to trace. If specified, the CPA will be ignored.

  For a compressed data set, the buffers might contain data in a compressed format.

  The PARM1 value must be entered in hexadecimal. Each bit in the subparameter represents a trace option. If the bit is active (1), the corresponding control block is traced or, in the case of bits in byte 5, the corresponding condition is taken. The bits are as follows.

Table 120. PARM1 Subparameter Bits

Byte 0	Notes	Description
1...	1	ABP–actual block processor.
.1..	1	ACB–access method control block.

Table 120. PARM1 Subparameter Bits (continued)

Byte 0	Notes	Description
..1.		AMB–access method block.
...1	1	AMBL–access method block list.
.... 1...		AMBXN–access method block extension.
.... .1..		AMDSB–access method data set statistics block.
.... ..1.	1	ARDB–address range definition block.
....1	1	BIB–base information block.

Table 121. PARM1 Subparameter Bits

Byte 1	Notes	Description
1...	2	BSPH–buffer subpool header.
.1..	2	BUFC–buffer control block.
..1.	1	CMB–cluster management block.
...1	7	CPA–channel program area.
.... 1...	1	CSL–core save list.
.... .1..		DIWA–data insert work area.
.... ..1.	1	EDB–extent definition block.
....1	1	HEB–header element block.

Table 122. PARM1 Subparameter Bits

Byte 2	Notes	Description
1...		ICWA–index create work area.
.1..		IICB–ISAM interface control block.
..1.		IMWA–index modification work area.
...1		IOMB–I/O management block.
.... 1...	7	IOSB–I/O supervisor block.
.... .1..		IXSPL–index search parameter list.
.... ..1.	1	LPMB–logical-to-physical mapping block.
....1	2	PLH–placeholder.

Table 123. PARM1 Subparameter Bits

Byte 3	Notes	Description
1...		RPL–request parameter list.
.1..		SRA–sphere record area.
..1.		UPT–upgrade table.
...1	1	VAT–valid AMBL table.
.... 1...	1	VMT–volume mount table.
.... .1..		VSI–VSAM shared information block.
.... ..1.	1	VVT–VSRT vector table.
....1		VSRT–VSAM shared resources table.

Table 124. PARM1 Subparameter Bits

Byte 4	Notes	Description
1...		WAX–work area for path processing.
.1..		WSHD–working storage header.
..1.	2	Buffers.
...1		User's search argument or key
.... 1...		User's record.
... .1..		Caller's registers.
... ..1.	6	Compression blocks
... ...x		Reserved.

Table 125. PARM1 Subparameter Bits

Byte 5	Notes	Description
1...		Do not trace data control blocks.
.1..		Do not trace index control blocks.
..1.	2	Trace all control blocks.
...1	1	Limit: one trace of control blocks.
.... 0...		KEY=keydata (the KEY contains a key value).
.... 1...		KEY=lowRBA-highRBA (the KEY contains RBA values).
.... .1..		Trace AIX®, PATH, or UPGRADE processing (PARM2 required).
.... ..1.	3	Validity-check control blocks; trace if bad.
....x		Reserved.

- **PARM2=**trace options

  PARM2 controls the tracing of any VSAM-opened data sets.

  The PARM2 subparameter is used to control the tracing of:
  – An alternate index's base cluster when opened as a path
  – A base cluster's UPG (upgrade) data set.

  PARM2 is used only if PARM1, byte 5 bit 5 (X'04') is specified. It has the same options as PARM1's except for the last byte (byte 5) which is shown in 126.

Table 126. PARM2 Subparameter Bits

Byte 5	Notes	Description
xxxx		Same usage as in PARM1.
.... xx..		Reserved.
.... ..1.	4	Trace all associated data sets.
....1	5	Trace UPGRADE control blocks.

Notes:

1. When **limit: one trace of control blocks** (byte 5 bit 3) is specified, the control blocks indicated with note 1 are traced only on the first call to R/M trace. These control blocks, generally, do not change after the data set is opened.
2. When **trace all control blocks** (byte 5 bit 2) is specified, the control blocks indicated with note 2 are traced even when the current request does not use them. If this bit is off, only those control blocks directly associated with the active request are traced.

Attention: Turning this bit on can cause a large amount of GTF data, depending on the number of strings and buffers and the size of buffers.

3. This option causes R/M trace to validity-check the pointers in the VSAM R/M control blocks, and if a chaining error is detected, the trace is taken.
4. When byte 5 bit 6 of PARM2 is off, only the data set being processed when R/M trace was called is traced. When this bit is on, R/M trace locates and traces data sets associated with the calling data set.
   – If the calling or associated data set was user-opened, PARM1 is used.
   – If the calling or associated data set was VSAM-opened, PARM2 is used.
   – When byte 5 bit 7 of PARM2 is off, UPGRADE data sets are not traced unless the UPGRADE was the calling data set. When byte 5 bit 7 of PARM2 is on, R/M trace treats UPGRADES as associated data sets; they are traced when the calling data set is a path, alternate index, or base.
   – When byte 4 bit 6 of PARM1 is on, tracing of compression related control blocks will occur. This includes control blocks CMSP, CMWA, and PCB. The data buffers for a compressed data set will not be traced unless both the buffers and compression block bits are set on.
   – Tracing of the CPA and IOSB is ignored for data sets defined as extended format or compressed.

Example of a DD statement requesting trace:

```
//KSDS01 DD DSN=VSAM.DATA.SET,DISP=SHR,
//     AMP=('TRACE=(PARM1=F00203000010',
//     'HOOK=(1,5),KEY=C1C2C3C4F7F8,PARM2=F123456789AB)')
```

The **TRACE=** on this DD statement activates the R/M trace for **VSAM.DATA.SET**. The R/M trace records are written out when the following conditions are met:

1. The record being processed has a key which begins with the character string ABCD78 as specified by **KEY=C1C2C3C4F7F8**.
2. Record management returns to the caller (HOOK 1) or returns from a call to EOV (HOOK 5) as specified by **HOOK=(1,5)**.

When these conditions are met, R/M trace writes to GTF the following data as requested by **PARM1**: ABP, ACB, AMB, AMBL, EDB, LPMB, and PLH The 10 in byte 5 of **PARM1** requests the ABP, ACB, AMBL, EDB, LPMB be traced only on the first call to R/M trace. **PARM2** has no effect on tracing because PARM1, byte 5 bit 5 (trace AIX, PATH, or UPGRADE processing) was not specified.

Adding Trace Points

Several trace points are predefined into VSAM; each has a unique trace point ID.

If one of these predefined trace points (specifiable by HOOK) is insufficient, you can add your own trace point by providing the code below:

VSAM Diagnostic Aids

```
VER nnnn  XXXX,XXXX                            ORIGINAL INSTRUCTION
VER nnn4  ????,???? MAIN                       NEXT SEQUENTIAL INSTRUCTION
VER pppp  ZZZZ,ZZZZ PATCH                      LOCATION OF PATCH
REP nnnn  47F0,pppp       BC    15,PATCH       BRANCH TO THE PATCH
REP pppp  BFFF,3078       ICM   15,15,AMBTRACE IS TRACE ACTIVE?
REP ppp4  4780,pp10       BZ    EXIT           NO, SKIP HOOK CODE
REP ppp8  58F0,F000       L     15,0(15)       LOAD TRACE ADDRESS
REP pppC  05EF            BALR  14,15          GO TO TRACE
REP pppE  0064            DC    X'0064'        ANY USER'S TRACE POINT ID
REP pp10  XXXX,XXXX EXIT                       ORIGINAL INSTRUCTION
REP pp14  47F0,nnn4       B     MAIN           RETURN TO NEXT SEQUENTIAL
                                               INSTRUCTION
```

You must ensure that the following registers contain the indicated data:
Register 2 Address of the PLH
Register 3 Address of the AMB (data or index)
Register 15 Address of IDA019ST

Failure to have these registers set could cause program checks and other unpredictable results. Register 0 is altered during the execution of trace routine. Care must be used in the selection of trace points, and in the selection of control blocks to be traced, because not all control blocks may be valid at a given time.

Ending the Record Management Trace Function

The tracing of a data set terminates when that data set is closed. If you want to terminate the tracing before closing the data set, you must stop GTF. You can then restart GTF, but if you specify USR or USRP with FF5, R/M tracing will resume.

Printing the Record Management Trace Output

VSAM trace records are formatted and printed with IPCS. “Printing GTF Records” on page 481 gives some examples of this service. For information on how to use IPCS, see *z/OS MVS IPCS User’s Guide*, for information on the command syntax, see *z/OS MVS IPCS Commands*.

VSAM Record Management Return and Reason Codes

VSAM provides return codes and reason codes to indicate the results of macro calls.

The return codes listed in this section are also applicable when RLS is active.

Return codes from Record Management (Request) Macros are set in register 15; reason codes are set in the RPL RPLFDBWD.

VSAM control block manipulation macros set return codes in registers 15 and 0.

Return codes and reason codes from VSAM macros can also be found in *z/OS DFSMS Macro Instructions for Data Sets*.

Return Codes from the Record-Management (Request) Macros

After a request macro or a CHECK or ENDREQ macro is issued, register 15 contains a return code.

After an asynchronous request to access a data set, VSAM indicates in register 15 whether the request was accepted, as shown in Table 127 on page 505.

Table 127. Return Codes from the Record-Management(Request) Macros

Return Code	Description
0(X'00')	Request was accepted.
4(X'04')	Request was not accepted because the request parameter list indicated by the request (RPL=address) was active for another request.
16(X'10')	• SMSVSAM server is not active • invalid OPEN connection

After a synchronous request, or a CHECK or ENDREQ macro, register 15 indicates whether the request was completed successfully, as shown inTable 128 .

Table 128. Return Codes from the Record-Management(Request) Macros

Return Code	Description
0(X'00')	Request completed successfully.
4(X'04')	Request was not accepted because the request parameter list indicated by the request (RPL=address) was active for another request.
8(X'08')	Logical error; specific error is indicated in the feedback field in the RPL.
12(X'0C')	Physical error; specific error is indicated in the feedback field in the RPL.
16(X'10')	• SMSVSAM server is not active • invalid OPEN connection

The feedback area of the request parameter list (RPL) contains additional diagnostic information that is used with the return codes in register 15 to determine the cause of an error.

The feedback area in the RPL is a fullword field:
Byte0 Problem determination code
Byte1 RPL return code (same as register 15)
Byte2 Component code
Byte3 Reason code

VSAM does not branch to an exit routine when register 15 is 0 on return from a request. The list in Table 129 describes the reason codes that may be found in the RPL feedback area when register 15 is 0.

Table 129. Return Codes from the Record-Management(Request) Macros - R15=0

RPLRTNCD Code	Condition
0(X'00')	Request completed successfully.
4(X'04')	Request completed successfully. For retrieval, VSAM mounted another volume to locate the record; for storage, VSAM allocated additional space or mounted another VSAM EOV was called.
8(X'08')	For GET requests, indicates that a duplicate key follows; for PUT requests, indicates that a duplicate key was created in an alternate index with the nonunique attribute.
12(X'0C')	(Shared resources only.) A buffer needs to be written.
16(X'10')	Control area split was required because a sequence set control interval had free space insufficient to contain the key to be inserted.
24(X'18')	Buffer found but not modified: no buffer writes performed.

Table 129. Return Codes from the Record-Management(Request) Macros - R15=0 (continued)

RPLRTNCD Code	Condition
28(X'1C')	A CI split for the CI with the RBA acquired from RPLDDDD which was interrupted. The CI was read as nonupdate with address access. This warning condition indicates that duplicate data records may exist.
32(X'20')	Possible causes: • Request deferred for a resource held by the terminated RPL is asynchronous and cannot be restarted by TERMRPL. • A MRKBFR request is invalid because no candidate buffer could be found. • For IDARETLK, there were no locks to retain since no update locks exist for this SUBSYSNM/LUWID/SPHERE.
36(X'24')	Possible data set error condition was detected by TERMRPL: 1. The request was abnormally terminated in the middle of its I/O operation. 2. One of the data/index BUFCs of the string contains data that needs to be written (BUFCMW=ON) but it was invalidated by TERMRPL.
40(X'28')	Error in PLH data BUFC pointer was detected by TERMRPL.
43(X'2B')	EOV called to retrieve or update the dictionary token in the catalog for a compressed data set.
44(X'2C')	EOV called to update catalog statistics.

See the following discussions for the logical-error and physical-error return codes.

Function Codes for Logical and Physical Errors

When a logical or physical error occurs during processing that involves alternate indexes, VSAM provides a code in the RPLCMPON field that indicates whether the base cluster, its alternate index, or its upgrade set was being processed and whether upgrading was satisfactory or may have been incorrect because of the error (see Table 130).

Table 130. Function Codes For Logical and Physical Errors

Code	What Was Being Processed	Status of Upgrading
0(X'00')	Base cluster	Satisfactory.
1(X'01')	Base cluster	May be incorrect.
2(X'02')	Alternate index	Satisfactory.
3(X'03')	Alternate index	May be incorrect.
4(X'04')	Upgrade set	Satisfactory.
5(X'05')	Upgrade set	May be incorrect.

Logical-Error Return Codes

When a logical-error-analysis exit routine (LERAD) is provided, it gets control for logical errors, and register 15 does not contain 8, but contains the entry address of the LERAD routine.

Table 131 on page 507 gives the contents of the registers when VSAM exits to the LERAD routine.

Table 131. Contents of Registers When a LERAD Routine Gets Control

Register	Contents
0	Unpredictable.
1	Address of the request parameter list that contains the feedback field the routine should examine. The register must contain this address if the exit routine returns to VSAM.
2-13	Same as when the request macro was issued. Register 13, by convention, contains the address of the processing program's 72-byte save area, which must not be used as a save area by the LERAD routine if the routine returns control to VSAM.
14	Return address to VSAM.
15	Entry address to the LERAD routine. The register does not contain the logical-error indicator.

If a logical error occurs and a LERAD exit routine is not provided (or the LERAD exit is inactive), VSAM returns control to the processing program following the last executed instruction. Register 15 indicates a logical error (8), and the feedback field in the request parameter list contains a code identifying the error. Register 1 points to the request parameter list.

See *z/OS DFSMS Installation Exits* for additional information on the LERAD exit routine.

Table 132 gives the logical-error return codes in the feedback field and explains what each means.

Table 132. Logical-Error Reason Codes found in the RPL Feedback Field when R15=8 and LERAD is not active.

RPLERRCD Code	Symbol	Condition
4(X'04')	RPLEODER	End of data set encountered (during sequential retrieval). Either no EODAD routine is provided, or one is provided and it returned to VSAM and the processing program issued another GET.
8(X'08')	RPLDUP	Attempt was made to store a record with a duplicate key.
12(X'0C')	RPLSEQCK	Attempt was made to store a record out of ascending key sequence; record may also have a duplicate key.
16(X'10')	RPLNOREC	Record not found.
20(X'14')	RPLEXCL	• For MACRF=RLS, this code means that there was an intra-LUDWID exclusive control conflict. • For non RLS, record already held in exclusive control by another requester.
21(X'15')		For MACRF=RLS, the request was rejected due to deadlock; the deadlock was resolved by rejecting this request.
22(X'16')		For MACRF=RLS, the request was rejected due to timeout. The request was presumed to be an inter resource manager deadlock.
24(X'18')	RPLNOMNT	• For MACRF=RLS, lock request rejected because the lock is held by a failed sybsystem. • For non RLS, the record resides on a volume that cannot be mounted.

Table 132. Logical-Error Reason Codes found in the RPL Feedback Field when R15=8 and LERAD is not active. (continued)

RPLERRCD Code	Symbol	Condition
28(X'1C')	RPLNOEXT	Data set cannot be extended because VSAM cannot allocate additional direct-access storage space. Either there is not enough space left in the data space for the secondary-allocation request or an attempt was made to increase the size of a data set by splitting the control area (high used RBA change) during processing with SHROPT=4 and DISP=SHR.
32(X'20')	RPLINRBA	An RBA was specified that does not give the address of any data record in the data set. This error will be returned by register 10 and register 8 if the RBA provided addresses a relative CI greater than 4GB.
36(X'24')	RPLNOKR	Key ranges were specified for the data set when it was defined, but no range was specified that includes the record to be inserted.
40(X'28')	RPLNOVRT	Insufficient virtual storage in the address space to complete the request. Or there is insufficient storage available to dynamically add another string.
44(X'2C')	RPLINBUF	Work area not large enough for the data record (GET with OPTCD=MVE).
48(X'30')	RPLINTRM	Invalid options, data set attributes, or processing conditions specified for TERMRPL request: • CNV processing • The specified RPL is asynchronous • Chained RPLs • PATH processing • Shared resources (LSR/GSR) • Create mode • RRDS • Data set contains spanned records • User not in Key 0 and supervisor state • EOV in process (secondary allocation).
52(X'34')	RPLPTERM	The previous request was TERMRPL.
56(X'38')	RPLCTERR	• For MACRF=RLS, RPL reuse violation. The RPL request contained positioning information from a previous request and the ACB or LUDWID specified in the RPL did not match the previous ACB or LUDWID. • For non RLS, error from catalog update at the beginning of a CI/CA split for a backup while opening a data set.
60(X'3C')	RPLCTERR	Available for RLS use.
64(X'40')	RPLNOPLH	• For RLS, the limit of 1024 outstanding requests for this ACB has been exceeded. • As many requests are active as the number specified in the STRNO parameter of the ACB macro; therefore, another request cannot be activated. • Or there is insufficient storage available to dynamically add another string.
68(X'44')	RPLINACC	Attempt was made to use a type of processing (output or control-interval processing) that was not specified when the data set was opened.
72(X'48')	RPLINKEY	• For RLS, add GETIX or PUTIX to any data set organization. • For non RLS, one of the following may apply: – Keyed GET request for an ESDS – GETIX or PUTIX to ESDS – Fixed length RRDS
76(X'4C')	RPLINADR	• An ADR or CI PUT was issued to add a record to a KSDS or VRRDS • Or a CI PUT was issued to a fixed length RRDS.
80(X'50')	RPLERSER	An ERASE request was issued for access to an ESDS or CI.

Table 132. Logical-Error Reason Codes found in the RPL Feedback Field when R15=8 and LERAD is not active. (continued)

RPLERRCD Code	Symbol	Condition
84(X'54')	RPLINLOC	• OPTCD=LOC was specified for a PUT request or in the *previous* request parameter list, in a chain of parameter lists. • LOC not supported (RLS only)
88(X'58')	RPLNOPTR	A sequential GET or PUT request was issued without VSAM having been positioned for it, or a change was made from addressed access to keyed access without VSAM having been positioned for keyed sequential retrieval, or an illegal switch between forward and backward processing was attempted.
92(X'5C')	RPLINUPD	A PUT, ERASE, or IDALKCD was issued without a previous GET for UPDATE. Or a PUTIX was issued without a previous GETIX.
96(X'60')	RPLKEYCH	• For RLS, PUT NUP attempt to change key specified by previous IDALKADD request. • For non-RLS, an attempt was made to change the prime key or the reference key during an update.
100(X'64')	RPLDLCER	Attempt was made to change the length of a record during an addressed update.

Table 132. Logical-Error Reason Codes found in the RPL Feedback Field when R15=8 and LERAD is not active. (continued)

RPLERRCD Code	Symbol	Condition
104(X'68')	RPLINVP	The RPL options are either invalid or conflicting in one of the following ways: • SKP was specified and either KEY was not specified or BWD was specified. XRBA was not specified in the RPL OPTCD when a GET DIR or a POINT REQUEST was issued in ADR|CNV mode with LRD=OFF and PLARG points to a non-zero argument (RBA), while processing an EA data set. • BWD was specified for CNV processing. • FWD and LRD were specified. • Neither ADR, CNV, nor KEY was specified in the RPL. • BFRNO is invalid (less than 1 or greater than the number of buffers in the pool). • WRTBFR, MRKBFR, or SCHBFR was issued, but either TRANSID was greater than 31 or a shared-resources option was not specified. • ICI processing was specified, but a request other than a GET or a PUT was issued. CNV processing for a compressed data set was specified. Only VERIFY and VERIFY REFRESH are allowed. For RLS only: • IDARECOV, IDALKREL, IDAINQRC, IDARETLK, or IDAQUIES issued and ACB is not an RLS Control ACB. • A Record Management GET or PUT was issued against an RLS Control ACB. • No LUWID specified for IDALKREL, IDARETLK, or IDALKCD. • SUBSYSNM was specified in ACB at OPEN, but GET, IDALKADD, PUT, POINT, or ERASE specified LUWID=0. • RLS options (LUWID, read integrity options, cold start) are not supported for NSR/GSR/LSR access. • CNV access specified for RLS access or an extended function data set. • ADR access to KSDS for RLS. • Invalid record management request issued against a control ACB. • Non-commit protocol application specified: – CRE on POINT or GET NUP – KL on GET UPD to a recoverable sphere • Invalid RLS request: LSR or MSS request macros, GETIX, or PUTIX. • IDAEADD is invalid for KSDS, RRDS, or VRRDS.
108(X'6C')	RPLINLEN	RECLEN specified was larger than the maximum allowed, equal to 0, smaller than the sum of the length and the displacement of the key field, or not equal to record (slot) length specified for a relative record data set.
112(X'70')	RPLKEYLC	KEYLEN specified was too large or equal to 0.
116(X'74')	RPLINLRQ	• Available for RLS. • For non RLS, invalid request during load mode. A GET, POINT, ERASE, direct PUT, skip sequential PUT, or PUT with OPTCD=UPD not permitted during initial data set loading (that is, for storing records in the data set the first time it is opened).
120(X'78')	RPLINTCB	Current job step TCB is not correct one. For RLS, request issued in cross-memory mode.
124(X'7C')	RPLUEXCL	• Available for RLS. • For non RLS, a request was canceled from a user JRNAD exit.

Table 132. Logical-Error Reason Codes found in the RPL Feedback Field when R15=8 and LERAD is not active. (continued)

RPLERRCD Code	Symbol	Condition
128(X'80')	RPLIXHHP	Index is invalid, request cannot be completed.
132(X'84')	RPLSRLOC	An attempt was made in locate mode to retrieve a spanned record.
136(X'88')	RPLARSRK	An addressed GET was issued for a spanned record in a key-sequenced data set.
140(X'8C')	RPLSRISG	Inconsistent spanned-record segments.
144(X'90')	RPLNBRCD	Invalid pointer in an alternate index (no associated base record).
148(X'94')	RPLNXPTR	The maximum number of pointers in the alternate index has been exceeded.
152(X'98')	RPLNOBFR	(Shared resources only.) Not enough buffers are available to process the request.
156(X'9C')	RPLINCNV	An invalid control interval was detected during keyed processing. The possible invalid conditions are: 1. A key is not greater than the previous key. 2. A key is not in the current control interval. 3. A spanned record RDF is encountered. 4. A freespace pointer is invalid. 5. The number of records does not match a group RDF record count.
160(X'A0')	RPLBMWER	A request was issued to invalidate a modified buffer. Required quiesce exit does not exist.
161(X'A1')	RPLQCLRJ	QUICLOSE request is rejected because the sphere is already marked quiesced.
162(X'A2')	RPLQUIRJ	The quiesce status of the sphere means this IDAQUIES request cannot be accepted from this application. Issued for the following: • This is a QUICMP request and field QUIERTOK in the IFGQUIES parameter area does not contain a request token corresponding to a QUIESCE exit invocation which VSAM is waiting for a QUICMP response from this application. • This QUICEND or QUIBEND request is rejected because the application signalled completion of its processing for this QUICOPY or QUIBWO event. • This QUICEND or QUIBEND request is rejected because the applications' QUIESCE exit was not driven for this QUICOPY or QUIBWO event.
163(X'A3')	RPLQACBO	IDAQUIES type QUICMP request rejected. ACBs for the sphere remain open for the application and this is an IDAQUIES type QUICLOSE.
164(X'A4')	RPLQCNCL	IDAQUIES request did not complete successfully. The request is canceled.
167(X'A7')	RPLQNSUP	The field QUIESTYP in the IFGQUIES parameter area specifies an invalid request type or the eye-catcher in IGFQUIES is invalid.
169(X'A9')	RPLQCTGF	IDAQUIESC request failed because the Catalog Locate issued for the specified sphere or component name failed.
170(X'AA')	RPLQUNFL	The QUIOPEN, QUICEND, or QUIBEND request is rejected because the requested unquiesce operation is already started for this sphere.
172(X'AC')	RPLACQER	IDAQUIES request failed because the specified sphere is not an SMS VSAM data set.
176(X'B0')	RPLSTGER	The ACB specified in IFGQUIES is not a valid ACB open for RLS processing.

Table 132. Logical-Error Reason Codes found in the RPL Feedback Field when R15=8 and LERAD is not active. (continued)

RPLERRCD Code	Symbol	Condition
181(X'B5')	RPLQRACF	This IDAQUIES request is rejected because the requestor does not have update authority to the sphere. Issued for the following: • This is a type QUICLOSE request. Successful completion of the request results in a catalog update to mark the sphere quiesced. Since the requestor does not have update authority, the request is rejected. • The catalog shows this sphere is quiesced. Successful completion of the QUIOPEN request would result in an ICF update to reset the quiesced state of the sphere. Since the requestor does not have update authority, the request is rejected.

182(X'B6')	RPLQINPR	IDAQUIES request rejected because an IDAQUIES is already in progress for this sphere. If an RPL message area (address in RPLERMSA) of sufficient length (specified in RPLEMLEN) is specified, the following information is returned:
		`Offset  Length  Description`
		`------  ------  -----------`
		`  0       1     Type of quiesce event already in progress for this sphere.`
		`                Quiesce type constants are defined in IFGQUIES macro.`

183(X'B7')	RPLMIGRA	IDAQUIES request rejected because data set is migrated. .
184(X'B8')	RPLABEND	An ABEND condition occurred while processing this VSAM request. The VSAM RLS FRR (Functional Recovery Routine) intercepted the failure and failed the VSAM request with this reason code.
186(X'BA')	RPLEOVER	End of volume initialization failed when DATASET tried to extend.
188(X'BC')	RPLNO241	Storage in subpool 241 is not available.
192(X'C0')	RPLIRRNO	Invalid relative record number.
196(X'C4')	RPLRRADR	An addressed request was issued to a relative record data set.
200(X'C8')	RPLPAACI	Addressed or control-interval access was attempted through a path.
204(X'CC')	RPLPUTBK	PUT-insert requests (or for RLS, IDALKADD requests) are not allowed in backward mode.
208(X'D0')	RPLINVEQ	Invalid ENDREQ request.
212(X'D4')	RPLNOSPL	Unable to split index.
224(X'E0')	RPLMOIB	A MRKBFR request was issued for an invalid buffer.
228(X'E4')	RPLINVMD	A cross-memory caller is not in supervisor state, in SRB, in cross-memory mode, or callers of RPL does not specify SYN processing.
229(X'E5')	RPLDELCH	The record length changed during decompression processing.
232(X'E8')	RPLUPERR	Cross-memory mode caller did not post the ECB in the UPAD exit routine.
235(X'EB')	RPLBMWER	If the SMSVSAM server is not available, the IDAQUIES request fails with R15=16.
236(X'EC')	RPLINVSI	Validity check error from SVC 109 for share options 3 or 4.
240(X'F0')	RPLUSTAT	Buffer pool status is unknown. The buffer use chain may be changing or a buffer is being modified or invalidated. Reissue the request.
244(X'F4')	RPLSVR14	Register 14 stack size is not large enough.
245(X'F5')	RPLCMSCE	Severe error returned from CMS for a compress call.
246(X'F6')	RPLCMSDE	Severe error returned from CMS for a decompress call.
248(X'F8')	RPLRST14	Register 14 return offset is negative.

Table 132. Logical-Error Reason Codes found in the RPL Feedback Field when R15=8 and LERAD is not active. (continued)

RPLERRCD Code	Symbol	Condition
250(X'FA')	RPLINVDT	No valid directory token exists. The data set cannot be decompressed.
251(X'FB')	RPLBMWER	Internal VSAM error.
252(X'FC')	RPLER252	Record mode access is not valid for a linear data set.
253(X'FD')	RPLER253	Verify function is not valid for a linear data set.
254(X'FE')	RPLERQUS	I/O activity on the data set not quiesced before issuing WTBFR TYPE=DS.

Physical-Error Return Codes

When a physical-error-analysis exit routine (SYNAD) is provided, it gets control for physical errors, and register 15 does not contain 12, but contains the entry address of the SYNAD routine.

For additional information on the SYNAD exit routine, see *z/OS DFSMS Installation Exits*.

Table 133 gives the contents of the registers when VSAM exits to the SYNAD routine.

Table 133. Contents of Registers When a SYNAD Routine Gets Control

Register	Contents
0	Unpredictable.
1	Address of the request parameter list that contains a feedback return code and the address of a message area, if any. If a request macro was issued, the RPL is the one pointed to by the request macro; if a CLOSE macro was issued, the RPL was built by VSAM to process the close request. Register 1 must contain this address if the exit routine returns to VSAM.
2-13	Same as when the request macro or CLOSE macro was issued. Register 13, by convention, contains the address of the processing program's 72-byte save area, which may not be used by the SYNAD routine if it returns control to VSAM.
14	Return address to VSAM.
15	Entry address to the SYNAD routine. The register does not contain the physical-error indicator. **Note:** The SYNAD exit, like other user exits, might not return control to VSAM.

If a physical error occurs and a SYNAD exit routine is not provided (or the SYNAD exit is inactive), VSAM returns control to the processing program following the last executable instruction. Register 15 indicates a physical error (12), and the feedback field in the request parameter list contains a code identifying the error. Register 1 points to the request parameter list.

Table 134 on page 514 gives the physical-error return codes in the feedback field and explains what each indicates. If the user provided a message area, it contains a physical-error message with more details about the error.

Table 134. Physical-Error Reason Codes in the RPL Feedback Field from a Request Macro

RPLERRCD Code	Symbol	Condition
4(X'04')	RPLRDERD	Read error occurred for a data component.
8(X'08')	RPLRDERI	Read error occurred for the index set of an index component.
12(X'0C')	RPLRDERS	Read error occurred for the sequence set of an index component.
16(X'10')	RPLWTERD	Write error occurred for a data component.
20(X'14')	RPLWTERI	Write error occurred for the index set of an index component.
24(X'18')	RPLWTERS	Write error occurred for the sequence set of an index component.
		All physical errors are detected by IDA019R5 from I/O Management abnormal-end appendage, IDA121A4.

Table 135 gives the format of a physical-error message. The format and some of the contents of the message are purposely similar to the format and contents of the SYNADAF message, which is described in *z/OS DFSMS Macro Instructions for Data Sets*.

Table 135. Format of Physical-Error Messages

Field	Bytes	Length	Discussion
Message Length	0-1	2	Binary value of 128
	2-3	2	Unused (0)
Message Length-4	4-5	2	Binary value of 124 (provided for compatibility with SYNADAF message)
	6-7	2	Unused (0)
Address of I/O Buffer	8-11	4	The I/O buffer associated with the data in relation to which the error occurred
The rest of the message is in printable format:			
Date	12-16	5	YYDDD (year and day)
	17	1	Comma (,)
Time	18-25	8	HHMMSSTH (hour, minute, second, and tenths and hundredths of a second).
	26	1	Comma (,)
RBA	27-38	12	Relative byte address of the record in relation to which the error occurred.
	39	1	Comma (,)
Data-Set Type	40	1	'D' for data or 'I' index
	41	1	Comma (,)
Volume Serial Number	42-47	6	Volume serial number of the volume in relation to which the error occurred.
	48	1	Comma (,)
Job Name	49-56	8	Name of the job in which error occurred.

Table 135. Format of Physical-Error Messages (continued)

Field	Bytes	Length	Discussion
	57	1	Comma (,)
Step Name	58-65	8	Name of the job step in which error occurred.
	66	1	Comma (,)
Unit	67-70	4	Device number on which the error occurred.
	71	1	Comma (,)
Device Type	72-73	2	The type of device in relation to which the error occurred (always DA for direct access).
	74	1	Comma (,)
ddname	75-82	8	The ddname of the DD statement defining the data set in relation to which the error occurred.
	83	1	Comma (,)
Channel Command	84-89	6	The channel command that occasioned the error in the first two bytes, followed by '-OP'
	90	1	Comma (,)
Message condition codes:	91-105	15	Messages are divided according to ECB X'41'—'INCORR LENGTH' 'UNIT EXCEPTION' 'PROGRAM CHECK' 'PROTECTION CHK' 'CHAN DATA CHK' 'CHAN CTRL CHK' 'INTFCE CTRL CHK' 'CHAINING CHK' 'UNIT CHECK' 'SEEK CHECK'
If the type of the unit check can be determined, this message is replaced by one of the following:			

Table 135. Format of Physical-Error Messages (continued)

Field	Bytes	Length	Discussion
			'CMD REJECT' 'INT REQ' 'BUS OUT CK' 'EQP CHECK' 'DATA CHECK' 'OVER RUN' 'TRACK COND CK' 'SEEK CHECK' 'COUNT DATA CHK' 'TRACK FORMAT' 'CYLINDER END' 'INVALID SEQ' 'INVALID SUFFFIX' 'NO RECORD FOUND' 'FILE PROTECT' 'MISSING A.M.' 'OVERFL INCP' X'48'—'PURGED REQUEST' X'4A'—'I/O PREVENTED' X'4F'—'R.HA.RO. ERROR' For any other ECB completion code—'UNKNOWN COND'
	106	1	Comma (,)
Physical Direct-Access Address	107-120	14	BBCCHHR (bin, cylinder, head and record)
	121	1	Comma (,)
Access Method	122-127	6	'VSAM'

Control Block Manipulation Return Codes

When the control block manipulation routine returns to the caller after successful completion, register 15 contains 0. If the request is GENCB, register 0 contains the total length of the area that contains the control block(s). Register 1 contains the address of the area.

When the control block manipulation routine returns to the caller with a nonzero value in register 15, an error occurred. If the request is TESTCB and the caller supplied an ERET keyword, return is to the location specified by the ERET keyword. Otherwise, the control block manipulation routine returns control to the point of invocation, via the return address in register 14.

Register 15 contains a return code, which is explained in Table 136 on page 517.

Table 136. Control Block Manipulation Return Codes

Reg 15	Condition
0(X'00')	Successful completion.
4(X'04')	An error has been detected. The error code in register 0 indicates the type of error.
8(X'08')	Invalid use of the execute form of this macro. Since the return code is set by the macro expansion and not by the control block manipulation routine, the register 0 contents do not indicate an error code.

Register 0 contains an error code (see Table 137).

Table 137. Control Block Manipulation Error Codes

Code	Applicable Macros[1]	Condition
1(X'01')	G,M,S,T	The function type is invalid.
2(X'02')	G,M,S,T	The control-block type is invalid.
3(X'03')	G,M,S,T	The keyword type is invalid.
4(X'04')	M,S,T	The control block to be processed is not of the type specified.
5(X'05')	S,T	The ACB to be processed is closed—it must be open.
6(X'06')	S,T	The cluster whose index component was to be processed is not key-sequenced (does not include an index).
7(X'07')	M,S	The EXLST entry to be processed is not present.
8(X'08')	G	Not enough virtual storage is available, or (with AM=VTAM specified) list and execute forms are inconsistent.
9(X'09')	G,S	User area is too small.
10(X'0A')	G,M	Exit address is not specified in the input.
11(X'0B')	M	The RPL to be processed is active, or it is already being processed.
12(X'0C')	M	The ACB to be processed is open—it must be closed.
13(X'0D')	M	No exit address is specified in the input for the exit to be activated.
14(X'0E')	G,M,T	An invalid combination of option codes (for example, for MACRF or OPTCD) is specified.
15(X'0F')	G,S	The user area is not on a fullword boundary.
16(X'10')	G,M,S,T	A VTAM keyword is specified with AM=VTAM not specified.
19(X'13')	M,S,T	A specified keyword refers to a field beyond the end of the control block to be processed.
20(X'14')	S	A specified keyword requires processing with shared resources to be specified, but it is not.
21(X'15')	S,T	The block to be displayed or tested does not exist, because the data set is a dummy data set.
22(X'16')	S	AM=VTAM is specified with SHOWCB for RPL fields=NIB, but the RPLNIB is off, or SHOWCB RPL fields=Arg (CID) but the RPLNIB bit is on.

Table 137. Control Block Manipulation Error Codes (continued)

Code	Applicable Macros[1]	Condition
23(X'17')	G	The value specified in the length parameter exceeds the 65535 byte limit. All errors in control block manipulation are detected by IDA019C1.
26(X'1A')	S	A request was made using a field name which allows the returned data to have a one word length, but the value being returned requires two words. A value of X'FFFFFFFF' is returned in place of the true value. Register 15 will contain a value of zero.

Note:

1. G=GENCB, M=MODCB, S=SHOWCB, T=TESTCB

Appendix. Accessibility

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

- Use assistive technologies such as screen-readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen-readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Volume I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

© Copyright IBM Corp. 1979, 2002

Notices

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

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

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

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

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

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

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

© Copyright IBM Corp. 1979, 2002

IBM Corporation
Information Enabling Requests
Dept. DZWA
5600 Cottle Road
San Jose, CA 95193 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

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

Programming Interface Information

This publication is intended to help you to do diagnosis of DFSMSdfp failures and primarily documents diagnosis, modification, and tuning information.

Attention: Do not use this diagnosis, modification, and tuning information as a programming interface.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States, or other countries, or both:

3890
AIX
CICS
CICS/ESA
CICS/MVS
CT
DB2
DFSMSdfp
DFSMSdss
DFSMShsm
ECKD
ESCON
Footprint
Hiperspace
IBM
IBMLink
MVS/ESA
OS/2
Parallel Sysplex
RACF
RMF
TotalStorage
VTAM
z/OS

UNIX is a registered trademark of X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

Glossary

This glossary defines technical terms and abbreviations used in DFSMS documentation. If you do not find the term you are looking for, refer to the index of the appropriate DFSMS manual or view the *IBM Dictionary of Computing Terms* located at:

http://www.ibm.com/networking/nsg/nsgmain.htm

This glossary includes terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published part of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.
- The *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

The following cross-reference is used in this glossary:

See: This refers the reader to (a) a related term, (b) a term that is the expanded form of an abbreviation or acronym, or (c) a synonym or more preferred term.

A

access method services. A multifunction service program that manages VSAM and non-VSAM data sets, as well as catalogs. Access method services provides the following functions:

- Defines and allocates space for VSAM data sets and catalogs
- Converts indexed-sequential data sets to key-sequenced data sets
- Modifies data set attributes in the catalog
- Reorganizes data sets
- Facilitates data portability among operating systems
- Creates backup copies of data sets
- Assists in making inaccessible data sets accessible
- Lists the records of data sets and catalogs
- Defines and builds alternate indexes

active control data set (ACDS). A VSAM linear data set that contains an SCDS that has been activated to control the storage management policy for the installation. When activating an SCDS, you determine which ACDS will hold the active configuration (if you have defined more than one ACDS). The ACDS is shared by each system that is using the same SMS configuration to manage storage. See also *source control data set, communications data set*.

aggregate backup. The process of copying an aggregate group and recovery instructions so that a collection of data sets can be recovered later as a group.

aggregate group. A collection of related data sets and control information that have been pooled to meet a defined backup or recovery strategy.

automated tape library data server. A device consisting of robotic components, cartridge storage areas, tape subsystems, and controlling hardware and software, together with the set of tape volumes that reside in the library and can be mounted on the library tape drives. Contrast with *manual tape library*. See also *tape library*.

B

backup. The process of creating a copy of a data set or object to be used in case of accidental loss.

base configuration. The part of an SMS configuration that contains general storage management attributes, such as the default management class, default unit, and default device geometry. It also identifies the systems or system groups that an SMS configuration manages.

basic catalog structure (BCS). The name of the catalog structure in the integrated catalog facility environment. See also *integrated catalog facility catalog*.

binder. The DFSMS program that processes the output of language translators and compilers into an executable program (load module or program object). It replaces the linkage editor and batch loader in z/OS.

© Copyright IBM Corp. 1979, 2002

C

communications data set (COMMDS). The primary means of communication among systems governed by a single SMS configuration. The COMMDS is a VSAM linear data set that contains the name of the ACDS and current utilization statistics for each system-managed volume, which helps balance space among systems running SMS. See also *active control data set* and *source control data set*.</

compress. (1) (1) To reduce the amount of storage required for a given data set by having the system replace identical words or phrases with a shorter token associated with the word or phrase. (2) (2) To reclaim the unused and unavailable space in a partitioned data set that results from deleting or modifying members by moving all unused space to the end of the data set.

connectivity. (1) (1) The considerations regarding how storage controls are joined to DASD and processors to achieve adequate data paths (and alternative data paths) to meet data availability needs. (2) (2) In a system-managed storage environment, the system status of volumes and storage groups.

construct. One of the following: data class, storage class, management class, storage group, aggregate group, base configuration.

D

DASD volume. A DASD space identified by a common label and accessed by a set of related addresses. See also *volume.*

data class. A collection of allocation and space attributes, defined by the storage administrator, that are used to create a data set.

data set. In DFSMS, the major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access. In z/OS non-UNIX environments, the terms *data set* and *file* are generally equivalent and sometimes are used interchangeably. See also *file*. In z/OS UNIX environments, the terms *data set* and *file* have quite distinct meanings.

data set collection. A group of data sets which are intended to be allocated on the same tape volume or set of tape volumes as a result of data set stacking.

data set stacking. The function used to place several data sets on the same tape volume or set of tape volumes. It increases the efficiency of tape media usage and reduces the overall number of tape volumes needed by allocation. It also allows an installation to group related data sets together on a minimum number of tape volumes, which is useful when sending data offsite.

default unit. Part of the SMS base configuration, it identifies an esoteric (such as SYSDA) or generic (such as 3390) device name. If a user omits the UNIT parameter on the JCL or the dynamic allocation equivalent, SMS applies the default unit if the data set has a disposition of MOD or NEW and is *not* system-managed.

DFSMSdfp. A DFSMS functional component or base element of z/OS, that provides functions for storage management, data management, program management, device management, and distributed data access.

DFSMSdss. A DFSMS functional component or base element of z/OS, used to copy, move, dump, and restore data sets and volumes.

DFSMShsm. A DFSMS functional component or base element of z/OS, used for backing up and recovering data, and managing space on volumes in the storage hierarchy.

dictionary. A table that associates words, phrases, or data patterns to shorter tokens. The tokens replace the associated words, phrases, or data patterns when a data set is compressed.

dual copy. A high availability function made possible by nonvolatile storage in some models of the IBM 3990 Storage Control. Dual copy maintains two functionally identical copies of designated DASD volumes in the logical 3990 subsystem, and automatically updates both copies every time a write operation is issued to the dual copy logical volume.

dummy storage group. A type of storage group that contains the serial numbers of volumes no longer connected to a system. Dummy storage groups allow existing JCL to function without having to be changed. See also *storage group*.

E

expiration. The process by which data sets or objects are identified for deletion because their expiration date or retention period has passed. On DASD, data sets and objects are deleted. On tape, when all data sets have reached their expiration date, the tape volume is available for reuse.

extended format. The format of a data set that has a data set name type (DSNTYPE) of EXTENDED. The data set is structured logically the same as a data set that is not in extended format but the physical format is different. Data sets in extended format can be striped or

compressed. Data in an extended format VSAM KSDS can be compressed. See also *striped data set, compressed format*.

F

file. A collection of information treated as a unit. In z/OS non-UNIX environments, the terms *data set* and *file* are generally equivalent and are sometimes used interchangeably. See also *data set*.

file system. In the z/OS UNIX HFS environment, the collection of files and file management structures on a physical or logical mass storage device, such as a diskette or minidisk. See also *HFS data set*.

filtering. The process of selecting data sets based on specified criteria. These criteria consist of fully or partially-qualified data set names or of certain data set characteristics.

G

group. (1) With respect to partitioned data sets, a member and the member's aliases that exist in a PDS or PDSE, or in an unloaded PDSE. (2) A collection of users who can share access authorities for protected resources.

H

hierarchical file system (HFS) data set. A data set that contains a POSIX-compliant file system, which is a collection of files and directories organized in a hierarchical structure, that can be accessed using z/OS UNIX System Services. See also *file system.*

I

indexed VTOC. A volume table of contents with an index that contains a list of data set names and free space information, which allows data sets to be located more efficiently.

integrated catalog facility catalog. A catalog that is composed of a basic catalog structure (BCS) and its related volume tables of contents (VTOCs) and VSAM volume data sets (VVDSs). See also *basic catalog structure, VSAM volume data set*.

Interactive Storage Management Facility (ISMF). The interactive interface of DFSMS that allows users and storage administrators access to the storage management functions.

M

manual tape library. A set of tape drives defined as a logical unit by the installation together with the set of system-managed volumes which can be mounted on those drives. See also *tape library*. Contrast with *automated tape library*.

MVS/ESA. Multiple Virtual Storage/Enterprise Systems Architecture. A z/OS operating system environment that supports ESA/390.

O

OAM managed volume. Optical or tape volumes controlled by the object access method (OAM).

object. A named byte stream having no specific format or record orientation.

object access method (OAM). An access method that provides storage, retrieval, and storage hierarchy management for objects and provides storage and retrieval management for tape volumes contained in system-managed libraries.

object backup storage group. A type of storage group that contains optical or tape volumes used for backup copies of objects. See also *storage group*.

optical disk drive. The mechanism used to seek, read, and write data on an optical disk. An optical disk drive can be operator-accessible or library-resident.

optical library. A storage device that houses optical drives and optical cartridges, and contains a mechanism for moving optical disks between a cartridge storage area and optical disk drives.

optical volume. Storage space on an optical disk, identified by a volume label. See also *volume*.

P

partitioned data set (PDS). A data set on direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

partitioned data set extended (PDSE). A system-managed data set that contains an indexed directory and members that are similar to the directory and members of partitioned data sets. A PDSE can be used instead of a partitioned data set.

performance. (1) (1) A measurement of the amount of work a product can produce with a given amount of resources. (2) (2) In a system-managed storage environment, a measurement of effective data processing speed with respect to objectives set by the storage administrator. Performance is largely determined by throughput, response time, and system availability.

permanent data set. A user-named data set that is normally retained for longer than the duration of a job or interactive session. Contrast with *temporary data set*.

pool storage group. A type of storage group that contains system-managed DASD volumes. Pool storage groups allow groups of volumes to be managed as a single entity. See also *storage group*.

primary data set. When referring to an entire data set collection, the primary data set is the first data set allocated. For individual data sets being stacked, the primary data set is the one in the data set collection that precedes the data set being stacked and is allocated closest to it.

program management. The task of preparing programs for execution, storing the programs, load modules, or program objects in program libraries, and executing them on the operating system.

program object. All or part of a computer program in a form suitable for loading into virtual storage for execution. Program objects are stored in PDSE program libraries and have fewer restrictions than load modules. Program objects are produced by the binder.

R

recovery. The process of rebuilding data after it has been damaged or destroyed, often by using a backup copy of the data or by reapplying transactions recorded in a log.

S

shelf. A place for storing removable media, such as tape and optical volumes, when they are not being written to or read.

shelf location. (1) A single space on a shelf for storage of removable media. (2) In DFSMSrmm, a shelf location is defined in the removable media library by a rack number, and in a storage location, it is defined by a bin number. See also *rack number, bin number*.

SMS configuration. A configuration base, Storage Management Subsystem class, group, library, and drive definitions, and ACS routines that the Storage Management Subsystem uses to manage storage. See also *configuration, base configuration, source control data set* .

SMS control data set. A VSAM linear data set containing configurational, operational, or communications information that guides the execution of the Storage Management Subsystem. See also *source control data set, active control data set, communications data set*.

SCDS. See *Source control data set.*

storage administrator. A person in the data processing center who is responsible for defining, implementing, and maintaining storage management policies.

storage class. A collection of storage attributes that identify performance goals and availability requirements, defined by the storage administrator, used to select a device that can meet those goals and requirements.

storage control. The component in a storage subsystem that handles interaction between processor channel and storage devices, runs channel commands, and controls storage devices.

storage group. A collection of storage volumes and attributes, defined by the storage administrator. The collections can be a group of DASD volumes or tape volumes, or a group of DASD, optical, or tape volumes treated as a single object storage hierarchy. See also *VIO storage group, pool storage group, tape storage group, object storage group, object backup storage group, dummy storage group*.

storage management. The activities of data set allocation, placement, monitoring, migration, backup, recall, recovery, and deletion. These can be done either manually or by using automated processes. The Storage Management Subsystem automates these processes for you, while optimizing storage resources. See also *Storage Management Subsystem*.

Storage Management Subsystem (SMS). A DFSMS facility used to automate and centralize the management of storage. Using SMS, a storage administrator describes data allocation characteristics, performance and availability goals, backup and retention requirements, and storage requirements to the system through data class, storage class, management class, storage group, and ACS routine definitions.

storage subsystem. A storage control and its attached storage devices. See also *tape subsystem*.

stripe. In DFSMS, the portion of a striped data set, such as an extended format data set, that resides on one volume. The records in that portion are not always logically consecutive. The system distributes records among the stripes such that the volumes can be read from or written to simultaneously to gain better performance. Whether it is striped is not apparent to the application program.

striping. A software implementation of a disk array that distributes a data set across multiple volumes to improve performance.

system data. The data sets required by z/OS or its subsystems for initialization and control.

system group. All systems that are part of the same Parallel Sysplex and are running the Storage Management Subsystem with the same configuration,

minus any systems in the Parallel Sysplex that are explicitly defined in the SMS configuration.

system-managed storage. Storage managed by the Storage Management Subsystem. SMS attempts to deliver required services for availability, performance, and space to applications. See also *system-managed storage environment*.

system-managed tape library. A collection of tape volumes and tape devices, defined in the tape configuration database. A system-managed tape library can be automated or manual. See also *tape library.*

system management facilities (SMF). A component of z/OS that collects input/output (I/O) statistics, provided at the data set and storage class levels, which helps you monitor the performance of the direct access storage subsystem.

system programmer. A programmer who plans, generates, maintains, extends, and controls the use of an operating system and applications with the aim of improving overall productivity of an installation.

T

tape configuration database. One or more volume catalogs used to maintain records of system-managed tape libraries and tape volumes.

tape library. A set of equipment and facilities that support an installation's tape environment. This can include tape storage racks, a set of tape drives, and a set of related tape volumes mounted on those drives. See also *system-managed tape library, automated tape library data server*.

tape storage group. A type of storage group that contains system-managed private tape volumes. The tape storage group definition specifies the system-managed tape libraries that can contain tape volumes. See also *storage group*.

tape subsystem. A magnetic tape subsystem consisting of a controller and devices, which allows for the storage of user data on tape cartridges. Examples of tape subsystems include the IBM 3490 and 3490E Magnetic Tape Subsystems.

tape volume. A tape volume is the recording space on a single tape cartridge or reel. See also *volume*.

temporary data set. An uncataloged data set whose name begins with & or &&, that is normally used only for the duration of a job or interactive session. Contrast with *permanent data set*.

threshold. A storage group attribute that controls the space usage on DASD volumes, as a percentage of occupied tracks versus total tracks. The *low migration threshold* is used during primary space management and interval migration to determine when to stop processing data. The *high allocation threshold* is used to determine candidate volumes for new data set allocations. Volumes with occupancy lower than the high threshold are selected over volumes that meet or exceed the high threshold value.

U

use attribute. (1) The attribute assigned to a DAD volume that controls when the volume can be used to allocate new data sets; use attributes are *public*, *private*, and *storage*. (2) For system-managed tape volumes, use attributes are *scratch* and *private*.

V

validate. To check the completeness and consistency of an individual ACS routine or an entire SMS configuration.

volume. The storage space on DASD, tape, or optical devices, which is identified by a volume label. See also *DASD volume, optical volume, tape volume*.

VSAM record-level sharing (VSAM RLS). An extension to VSAM that provides direct record-level sharing of VSAM data sets from multiple address spaces across multiple systems. Record-level sharing uses the z/OS Coupling Facility to provide cross-system locking, local buffer invalidation, and cross-system data caching.

VSAM sphere. The base cluster of a VSAM data set and its associated alternate indexes.

VSAM volume data set (VVDS). A data set that describes the characteristics of VSAM and system-managed data sets residing on a given DASD volume; part of an integrated catalog facility catalog. See also *basic catalog structure, integrated catalog facility catalog*.

Index

Numerics

3995 library long OBR format mapping 237
3995 library miscellaneous data record mapping 239
3995 optical disk drive long OBR mapping 238

A

ABDUMP routine 248, 250
accessibility 519
ACS executor return codes 438
ACS services, SMS 443
ACS translator return codes 437
AMASPZAP utility 248
AMS diagnostic aids
  access method services 473
  record level sharing 482
ANALYSIS command 391
AOM diagnostic aids 3

B

BDAM Diagnostic Aids 13

C

CAF trace
  activation 245
  GTF (Generalized Trace Facility) 479
  optional work area trace (O/C/EOV) 255
  record management trace 504
  SMS trace facility 431
  VSAM record management trace facility 498
catalog
  management, diagnostic aids 477
  return codes 69
CBROAM load module 241
CBSA 252, 253
common filter services
  AOM 108
  reason codes 108
  return codes 108
completion codes, BDAM 13
component ownership
  DADSM (DF107), DSCB7 45
  Mapping Macros (SC101), DEB 351
  OPEN/CLOSE/EOV (28413), DCB (BTAM, QTAM) 309
  OPEN/CLOSE/EOV (Data Management)
    DCB (BDAM) 298
    DCB (EXCP, SAM, BPAM) 259
    DCB (GAM) 331
    DCB (ISAM) 283
    DCB (TCAM) 320
compression management services (CMP), compression services 15
control block
  chains 420
control block *(continued)*
  ID 251
  record management information 492
CTRACE subcommand, AOM 4
CVAF (common VTOC access facility)
  abend dump 39
  CVAFGTF 27
  function and subfunction keywords 45
  return codes 47
  status codes 47
  tracing options 27
  VTOC index error messages 38
CVAF diagnostic aids
  CVSTAT field codes 40, 44
  debug examples 36
  debug trap options 33
  description 27
  GTF record format 29
  module-to-function cross-reference 27
  restriction 29
  trace events 29
CVAF-GTF trace of events
  CVAF events 27, 31, 32
  error records 248
  O/C/EOV 248
  output examples 249
CVSTAT
  CVSTAT 40
  field codes 40

D

DADSM (direct access device space management)
  OBTAIN function return codes 73
  RENAME function
    diagnostic information 83
    return codes 82
    status codes 90
  SCRATCH function
    diagnostic information 92
    return codes 91
    secondary status codes 100
    status codes 99
  UPDATE function diagnostic information 101
DADSM/CVAF Diagnostic Aids
  abend dumps 105
  COMMRC 103
  CREATE information 49, 82
  CREATE return codes 47, 65
  DASD calculation services 109
  description 27, 105, 113
  EXTEND return codes 65, 70
  ISJF function return codes 70
  LSPACE return codes 71
  module-to-function cross-reference 45
  PARTREL (partial release) return codes 74
  return and diagnostic codes 47

© Copyright IBM Corp. 1979, 2002

DASD Common Services Diagnostic Aids
  description 107, 113
  error recovery and detection 107
  problem determination area 107
  reason codes 108
  return codes 108
Data Control Block (BDAM) 298
Data Control Block (BTAM and QTAM) 309
Data Control Block (BTAM, QTAM) 309
Data Control Block (EXCP, SAM, BPAM) 258
Data Control Block (GAM) 331
Data Control Block (ISAM) 283
Data Extent Block (DEB) 350
DB2 trace
  description 245
  log data 245
DEB (Data Extent Block) 350
DESERV macro
  DESRC_SEVE 113
  reason codes 113
device
  device information services 110
  information services 110
DIAGNOSE output 477
diagnostic aids
  DASD calculation services 109
  dump types 2
  general information 1
  system dump types 2
diagnostic records, GTF data set 250
disability 519
Distributed FileManager/MVS diagnostic aids 117
Distributed FileManager/MVS errors
  access method error codes 141
  agent errors 140
  basic error processing 117
  communication errors 119
  communications error code 139
  DDM data stream errors 119
  DDM reply messages returned by command processors 126
  DDM reply messages returned by Distributed FileManager/MVS 125
  DFM errors 117
  DFM messages 120
  Distributed FileManager/MVS codes 136
  file/directory error codes 141
  formatting LOGREC records 132
  general error codes 137
  GTF trace of input/ output buffers 134
  locking error codes 141
  problem determination tools 127
  resource problems 118
  slip trap to SVC dump 132
  SRVDGN data items 135
  system logic errors 118
  system messages 120
  trace facility for Distributed FileManager/MVS 127
  unexpected source termination 119
dump
  points 474
dump *(continued)*
  types 2
dump points 474

E

error codes, DESERV macro 113
error conditions, feedback information 493
error records 248, 250
ERTB display panel 145
exception
  PDSE messages 391
  PDSE trace 398

F

field codes, CVSTAT 40
Format-7 DSCB 45
FREELATCH command 391

G

GTF (Generalized Trace Facility), printing GTF records 481

H

HFS component trace
  HFS trace table formatting 381
  tracing HFS manipulation 375

I

ICB (interrupt control block) 416
IFGOCETR started task, open/close 255
IMC (intermodule communications) 428
Input/Output Block 335
inter-module trace table 473
intra-module trace table 474
IPCS (interactive problem control system)
  AOM 3
  HFS trace table formatting 381
  invoking SMSDATA 422
  OAM IPCS processing 242
  PDSE trace table formatting 403
  printing GTF records 481
  printing R/M trace output 504
  SMSDATA 420
  trace table formatting 3, 129
  trace table formatting for VSAM RLS 485
  VSAM 481
ISJF function
  diagnostic information 70
  return codes 70
ISMF diagnostic aids 143
  controlling ISMF dump processing 147
  description 143, 187
  dump processing 143
  ERTB display panel 145
  execution options 143

ISMF diagnostic aids *(continued)*
inter-module traces 150
ISPF invoked in normal mode 148
ISPF invoked in test/trace mode 148
ISPF log 148
logging detailed error data 149
logging/abend control entry panel 144
passing control to ISPF or TSO/E 148
reason codes 151
recovering from abends 147
return codes 151
standard log information 149
trace-point traces 150

K

keyboard 519

L

LACS (Library Automation Communication Services) return and reason codes 218
LCS External Services (CBRXLCS) 212
logrec information
compression services messages 15
reason codes 17
return codes 17
symptom record 16
LSPACE macro
diagnostic information 71
parameter list diagnostic fields 73
return codes 71

M

Macro Names
DCBD 283, 298, 309, 320, 331
IEZDEB 350
IEZIOB 335
IEZIOB, IOB 335
maintenance patch area
adding a temporary change 248
description 248
locating 248
MDR (miscellaneous data record) 238
media manager diagnostic aids
description 187, 195
return codes 190
sample summary dump 188
messages
BDAM 13
CVAF VTOC error messages 38
SMS failure source identification 419
VIO 471

O

O/C/EOV (open/close/end-of-volume) diagnostic aids
abend error recording 248
description 247, 258, 479
O/C/EOV (open/close/end-of-volume) diagnostic aids *(continued)*
GTF 479
locating an object module in a storage dump 247
messages, trace 256
problem determination 255
trace 255
OAM (Object Access Method) diagnostic aids
CAF trace 245
CBROAM load module 241
DB2 problems 244
DB2 trace 245
description 195, 245
dumps 241
LACS return and reason codes 218
OAM DB2 databases 196
OAM IPCS processing 242
OAM macro return and reason codes 206
OSREQ return and reason codes 196
resource contention problems 244
SYS1.LOGREC records 234
SYSABEND DD statement 241
VERBEX OAMDATA 242
OAMDATA, OAM IPCS processing 242
OBR (outboard recorder) record 234
OBTAIN macro
return codes 73
using PARTREL 74
OPEN/CLOSE/End-of-Volume (O/C/EOV) diagnostic aids
abend error recording 248
description 247, 258, 479
GTF 479
locating an object module in a storage dump 247
messages, trace 256
problem determination 255
trace 255
optical device driver 235
optical device error recovery 235

P

PARM, TEST option 474
PARTREL macro
diagnostic information 75
return codes 74
using PARTREL 82
PDSE (partitioned data set extended), DADSM return codes 69
PDSE component trace
description 398
PDSE trace table formatting 403
trace table formatting 129
PDSE CTRACE Writer Options 400
PDSE diagnostic aids
ANALYSIS command 391
FREELATCH command 391
PDSE Trace Options 399
problem determination
area 107
error records 249

R

reason codes
- ACS executor 440
- ACS services 440
- ACS translator 441
- AOM 7
- catalog services 443
- common filter services 108
- configuration services 444
- construct access services 443
- DASD calculation services 109
- DASD common services 108
- DESERV macro 113
- device control facility 459
- device information services 110
- device pool 462
- disposition processing 449
- error recovery 461
- fast VTOC/VVDS access 459
- interpreter/dynamic allocation 450
- intersystem communications 450
- ISMF 151
- LACS 218
- message services 451
- operational services 451
- operator commands 460
- OSREQ macro 196
- scheduling services 452
- sms interface 439
- SMS tape volume record update 462
- subsystem initialization 452
- VDSS services 455
- VOLREF services 453

record management
- ABEND0CX error analysis 495
- control block information 492
- damaged data CIs 497
- damaged data sets 495
- damaged indexes 496
- diagnostic aids 492
- error conditions 493
- exclusive control error analysis 494
- physical I/O errors 497
- printing trace output 504
- return codes 504
- RPL feedback 493
- trace facility 498
- UPAD 492
- WAITX 492

RENAME macro
- DASD calculation services 109
- diagnostic information 83
- RENAME function 90
- return codes 82
- SCRATCH function 99
- status codes 90

return codes
- AOM 7
- control block manipulation 516
- DADSM functions 47, 82, 100
- DASD calculation services 109

return codes *(continued)*
- device information services 110
- ISMF 151
- LACS 218
- logical error 506
- media manager 190
- open and close 482
- OSREQ macro 196
- physical error 513
- record management macros 504
- VSAM 504
- VSAM record management 504

RLS component trace
- trace facility 483
- trace table formatting for VSAM RLS 485

RRCBSA
- contents 250
- storage dump 251

S

SAM diagnostic aid
- description 413, 417
- ICB 416
- SYNAD routine 417

SCRATCH macro
- diagnostic information 92, 99
- return codes 91
- secondary status codes 100
- status codes 99

SDWA (system diagnostic work area) 253

shortcut keys 519

SMSDATA
- output examples 425
- verb exit functions 420

SSI vector table 426

Storage Management Subsystem diagnostic aids
- ACS executor reason codes 440
- ACS executor return codes 438
- ACS services reason codes 440
- ACS translator reason codes 441
- ACS translator return codes 437
- automatic data area format 429
- catalog services reason codes 443
- configuration services reason codes 444
- construct access services reason codes 443
- control block chaining validation 420
- control block format 420
- description 419, 462, 465
- device control facility reason codes 459
- device pool SSI reason codes 462
- disposition processing reason codes 449
- dump contents 435
- error recovery reason codes 461
- fast VTOC/VVDS access reason codes 459
- IMC format 428
- IMC location 435
- interpreter/dynamic allocation reason codes 450
- intersystem communications reason codes 450
- invoking SMSDATA 422
- message services reason codes 451

Storage Management Subsystem diagnostic aids *(continued)*
message-indicated failure source identification 419
operational services reason codes 451
operator commands reason codes 460
procedure flow trace table format 430
return codes 437
scheduling services reason codes 452
SMS address space trace table 431
SMSDATA dump formatter 420
SMSDATA output examples 425
SMSDATA trace record format 427
SSI vector table format 426
subsystem initialization reason codes 452
subsystem initialization return codes 438
subsystem interface reason code 439
subsystem interface return codes 439
SUMDUMP sample 435
SYS1.LOGREC 432
tape volume record update reason codes 462
trace event options 431
trace table format 427
tracing events 431
VDSS services reason codes 455
verb exit functions 420
VOLREF services reason codes 453
SYNAD routine 417
SYS1.LOGREC records
driver records 235
error recording 253
hardware error records 234
SMS records 432
software error records 234
VIO 469
system dump types 2

T

TEST keyword 475
TEST option 474
Trace command for PDSE component trace 398
trace tables 473
description 473
dump points 474
formatting 129, 381
formatting for VSAM RLS 485
formatting with IPCS 3, 403
inter-module 473
inter-module trace table 473
intra-module 474
intra-module trace table 474
SMS address space trace table 431
SMS procedure flow trace table format 430
SMS trace table format 427
TEST keyword 475
TEST option 474
trace tables 473, 474
VSAM 473
transfer control table 247

U

UPDATE function diagnostic information 101, 103
utility service diagnostic aids
description 465, 468
IEBCOPY (PDSE version) 465

V

V-type address constant, unresolved 248
VERBEXIT subcommand
OAM IPCS processing 242
valid parameters 242
VIO (virtual I/O) diagnostic aids 469, 473
VSAM diagnostic aids
ABEND0CX error analysis 495
catalog management 477
control block information 492
control block manipulation 516
damaged data CIs 497
damaged data sets 495
damaged indexes 496
description 473, 504
dump points 474
error conditions 493
exclusive control error analysis 494
function codes 506
GTF 479
inter-module trace table 473
intra-module trace table 474
logical error return codes 506
open and close return codes 482
physical error return codes 513
physical I/O errors 497
printing GTF records 481
record management diagnostic aids 492
record management trace facility 498
return codes 504
RPL feedback 493
SNAP dump facility 492
TEST option 474
trace tables 473
UPAD 492
WAITX 492
VSAM RLS
trace table formatting for VSAM RLS 485
tracing VSAM RLS 483
VTOC (volume table of contents) index, error messages 38

X

XCTL table 247

Readers' Comments — We'd Like to Hear from You

z/OS
DFSMSdfp Diagnosis Reference

Publication No. GY27-7618-02

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

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

Name _____ Address _____

Company or Organization _____

Phone No. _____

Readers' Comments — We'd Like to Hear from You
GY27-7618-02

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
RCF Processing Department
M86/050
5600 Cottle Road
San Jose, CA
U.S.A. 95193-0001

Fold and Tape | **Please do not staple** | Fold and Tape

GY27-7618-02

Cut or Fold
Along Line

Program Number: 5694-A01

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GY27-7618-02