z/OS
Version 2 Release 3

DFSMSdfp Diagnosis

SC23-6863-30

Note

Before using this information and the product it supports, read the information in "Notices" on page 631.

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

Last updated: 2019-07-02

Contents

Figures

Tables

About this document

This document describes DFSMSdfp problem diagnosis.

The first chapter, Chapter 1, “Diagnosis guide,” on page 1, helps you initially to identify, describe, report, and resolve DFSMSdfp failures.

The remaining sections help 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. You should use the reference information in the rest of this book to gather and interpret failure-related data that supplements any data obtained from the initial failure when directed to do so by the diagnosis procedures in Chapter 1, “Diagnosis guide,” on page 1 or by the IBM Support Center.

For information about the accessibility features of z/OS®, for users who have a physical disability, see Appendix A, “Accessibility,” on page 627.

References to CICS

In this document, when the term “CICS,” is used, it refers to CICS® for MVS/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 documents in those libraries.

z/OS information

This information explains how z/OS references information in other documents and on the web.

When possible, this information uses cross document links that go directly to the topic in reference using shortened versions of the document title. For complete titles and order numbers of the documents for all products that are part of z/OS, see *z/OS Information Roadmap*.

To find the complete z/OS library, go to IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSLTBW/welcome).

How to send your comments to IBM

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

Important: If your comment regards a technical question or problem, see instead "If you have a technical problem" on page xxv.

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

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

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

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

To help us better process your submission, include the following information:

- Your name, company/university/institution name, and email address
- The following deliverable title and order number: z/OS DFSMSdfp Diagnosis, SC23-6863-30
- The section title of the specific information to which your comment relates
- The text of your comment.

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

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

If you have a technical problem

If you have a technical problem or question, do not use the feedback methods that are provided for sending documentation comments. Instead, take one or more of the following actions:

- Go to the IBM Support Portal (support.ibm.com).
- Contact your IBM service representative.
- Call IBM technical support.

Summary of changes

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line to the left of the change.

Summary of changes for z/OS Version 2 Release 3 (V2R3)

Changed

- Chapter 7, "DADSM/CVAF diagnostic aids," on page 109 has been updated with return code information for read-only volumes.
- The following sections have also been updated with diagnosis information for read-only volumes:
  - "DADSM CREATE (allocate) return and diagnostic codes" on page 127
  - "DADSM extend to same volume diagnostic codes" on page 144
  - "DADSM RENAME return and diagnostic codes" on page 154
  - "DADSM partial release return and diagnostic codes" on page 161
  - "DADSM SCRATCH return and diagnostic codes" on page 170
  - "DADSM UPDATE return and diagnostic codes" on page 179
  - "VTOC/Data set services (VDSS) reason codes" on page 552
- "Configuration services reason codes" on page 532 and "VTOC/Data set services (VDSS) reason codes" on page 552 have been updated with new reason codes for cloud support.
- "CVSTAT field codes" on page 118 has been updated with a new DEB2X field at offset 218, to indicate that the user was accessing a partitioned concatenation using BPAM.
- "DADSM LSPACE return and diagnostic codes" on page 149 has been updated with new values for the TIMEOUT parameter.
- "Logical-Error return codes" on page 613 has been updated.
- "IGDVSUIB (volume selection user information block)" on page 520 has been updated.
- "Subsystem interface reason codes" on page 526 has been updated.
- "Configuration services reason codes" on page 532 has been updated.
- "Scheduling services reason codes" on page 549 has been updated.
- "VTOC/Data set services (VDSS) reason codes" on page 552 has been updated with new reason code 17246 (X'435E').
- "OAM macro return and reason codes" on page 299 has been updated in support of a multiple OAM configuration.
- "OSREQ return and reason codes" on page 282 has been updated to include changes to OSREQ return and reason codes.
- "OAM IPCS processing" on page 337 has been updated to include changes to OAM IPCS parameters.
- "Configuration services reason codes" on page 532 has been updated to include a new reason code.
- "Configuration services reason codes" on page 532 has been updated to include new reason codes in the macro IGDCSR.

Summary of changes for z/OS Version 2 Release 2 (V2R2)

Changes made for z/OS V2R2

New

New CVSTAT field code. For details, refer to "CVSTAT field codes" on page 118.

New field name in the IOB. For details, refer to "IOB" on page 430.

New ACS executor reason code. For details, refer to "ACS executor reason codes" on page 528.

New configuration services reason code. For details, refer to "Configuration services reason codes" on page 532.

z/OS Version 2 Release 1 summary of changes

See the Version 2 Release 1 (V2R1) versions of the following publications for all enhancements related to z/OS V2R1:

- *z/OS Migration*
- *z/OS Planning for Installation*
- *z/OS Summary of Message and Interface Changes*
- *z/OS Introduction and Release Guide*

Chapter 1. Diagnosis guide

This section enables users of the DFSMSdfp functional component of DFSMS to perform the following tasks:

- Develop a string of standard symptom keywords describing a DFSMSdfp program failure as precisely as possible. Record the keywords on the "Keyword worksheet" on page 4 for permanent reference. You can make extra copies of the blank worksheet for future use.
- Use the procedure "Searching with the keyword string" on page 2 to search an IBM software support database (or contact the IBM Support Center to execute the search using your keyword string) to determine whether:
  - An APAR already documents the problem.
  - A resolution for the problem is already available.

  You may also contact the IBM Support Center for assistance with:

  - Developing a more effective keyword string
  - Gathering additional failure-related documentation
  - Submitting adequate documentation with an APAR, if required.

  Also see *z/OS Problem Management* for information about general z/OS problem determination techniques.

The keywords that this guide describes are intended for use in freeform searches. Freeform searches enable you to retrieve only the RETAIN records that contain all the search keywords you specified. You can use the same keywords as a base from which to conduct a search of the structured database (SDB) facility. Structured searches require that you specify the exact type of problem to retain. When you enter the keyword string, the SDB facility prompts you for the required information and then builds the structured keyword string for you in the appropriate format. In other words, the structured database performs the following tasks:

- Prompts you for failure symptoms
- Generates keywords according to a strict protocol
- Initiates the search operation.

If you request assistance from the IBM Support Center, and they recommend submitting an APAR, you will need the keyword strings for which you have executed searches.

What are keywords?

A keyword describes one aspect of a program failure. When doing your own software database search or contacting the IBM Support Center for assistance, identify your program failure with as many of the keywords as apply to the problem.

The full keyword string describes the following areas:

- Type-of-failure (always required)
- Module (if applicable)
- Modifier(s)
  - Function (when possible)
  - Subfunction (if applicable)
  - Other significant failure-related modifiers
- Component identification

- Release level
- Service level

Each keyword you add makes the search argument or keyword string more specific. The more precise the keyword string, the more selective the search, thereby yielding fewer matches in the software support database. If you do not find a similar problem in the database, you can broaden the scope of the search by deleting keywords, beginning at the end of the preceding list.

Searching with the keyword string

When you have developed a keyword string describing the DFSMSdfp software failure, you are ready to search the IBM software database, using the keyword string as a search argument.

Each keyword describes one aspect of a program failure. Specifying the component identifier keyword, together with the type-of-failure keyword, as a search argument, detects all APARs for that component with that type-of-failure. The more precisely you describe the failure with additional keywords, the more selective the resulting search is, yielding fewer problem descriptions for you to review.

Recommendation: For the catalog-related failures, you should construct two keyword strings because a problem could exist in a module that services catalogs. The first string should specify the ICFCAT keyword for the catalog, and the second should specify the VSAMCAT keyword for the catalog.

Software support databases

Several different software support database facilities exist for researching software problems.

Info/System

Info/System is an interactive retrieval program product designed for use with the companion database feature Info/MVS. The database comprises several logical files of related or similar information.

ServiceLink

This IBMLink function lets you search up-to-date databases regarding problem resolution. Contents include open and resolved Authorized Program Analysis Reports (APARs) and Program Temporary Fix (PTF) information.

Software Support Facility (SSF)

SSF is an IBM online database containing information about all current APARs and PTFs.

IBM Support Center personnel have direct access to SSF and are responsible for using the set of keywords you provide as a search argument. These representatives can help you improve the effectiveness of your search argument. If the problem has been previously reported, they can retrieve the records describing both the problem and the correction.

TalkLink

TalkLink enables you to review information electronically, submit questions and requests, and receive answers and information in the following ways:

- Conferences and forums

  Question-and-answer bulletin boards enable customers to ask questions and exchange experiences on a wide variety of products and services. Bulletin board system (BBS) users, IBM personnel, and affiliates worldwide share items posted on such a bulletin board.

- Customer-to-customer messaging

  Customers can send private messages to each other.

- Support

Customers can submit or view problem reports. The conference or forum owner responds to customer entries. Users can also place orders for product-related materials.

- Software libraries

  Customers can download application programs and upload contributions.

- News and announcements

  Recent information regarding products and services are available.

Procedure

You will be most successful in searching a software support database by observing the following rules:

- Use the keywords you have developed as your freeform search argument.
- Spell keywords the same way every time, exactly as this publication specifies them.
- Specify the keywords in the order suggested in the following text.
- Include all the following keywords and those of the keywords you were able to determine in any discussion with IBM. If it becomes necessary to submit an APAR, include the keyword strings used with the APAR documentation you send to IBM.
- Search the software support database, using as many of the following keywords as you have obtained and entering them in the order shown:

  1. Type-of-failure
  2. Load module, or CSECT, or both
  3. Modifiers
  4. Component identifier
  5. Release level
  6. Service level

- Scan the resulting list of known similar problems, eliminating APAR fixes that are already installed on your system.

  – If the list of matching APARs is too long to be practical, contact the IBM Support Center for assistance.

  – If you find a matching APAR problem description, contact the IBM Support Center to obtain the fix.

  – If you do not find a matching APAR problem description, broaden the search by eliminating keywords from the search argument as directed in "Techniques for varying the search argument" on page 3.

  – If you still do not find a matching APAR problem description, contact the Support Center as directed in " Contacting the IBM Support Center for assistance" on page 66. They might be able to refine your search argument and find a match in the database; otherwise, they can direct you to gather additional failure-related information.

- "Preliminary failure source isolation" on page 5 documents the process of building a keyword string.

Techniques for varying the search argument

When using the initial search argument, if you do not find a similar APAR problem description in one of the software databases, try the following suggestions for varying the argument to see if you can obtain a match:

- If you have described the failure as a wait, loop, or performance type-of-failure, replace the type-of-failure keyword with one of the other two keywords. For example, what appears to be a wait state might actually be a loop or a performance problem.
- If more than one type-of-failure keyword applies (for example, an abend and a message both occur), try all combinations of those keywords.
- Delete one keyword at a time in the following order:

1. Load module name
2. Service level
3. Release level
4. Modifiers
5. CSECT name

If you now identify a similar problem in the database, contact the IBM Support Center to obtain the fix.

If you still do not find a matching APAR problem description, contact the IBM Support Center as directed in “ Contacting the IBM Support Center for assistance” on page 66. They might be able to refine your search argument or keyword string and find a match in the database; otherwise, they can direct you to gather additional failure-related information.

Use the following keyword worksheet to develop the keyword strings and other information to give to the IBM Support Center.

Keyword worksheet

Date:	**Time:**	**IBM Problem Number:**

Problem description (including symptoms not described by keywords)

IBM licensed program information

Product Name:	**FMID:**		
Modification:	Feature:	Order Number:	PUT Tape Level:

Base z/OS system information

Name:	**FMID:**	**PUT Tape Level:**

Keywords

Type-of-Failure:	**Component ID:**
Module Name:	Modifiers:
Release Level:	Service Level:

Keyword strings used

Information provided by IBM Support Center

Preliminary failure source isolation

Use this section to isolate the probable source of a failure to an area within DFSMS or to the interfaces with ICKDSF or DFSORT. Because you are reading this information, the assumption is that you have performed problem source identification (PSI) and some measure of problem determination (PD), and you suspect that DFSMSdfp is related to the failure. Refer to *z/OS Problem Management*, *z/OS MVS Diagnosis: Reference*, and *z/OS MVS Diagnosis: Tools and Service Aids* for PSI and PD information.

Procedure

1. If the failure involves a DFSMSdfp component and ISMF does not seem to be involved, go to "Type-of-failure keyword" on page 7.
2. If the failure seems related to ISMF, use Figure 1 on page 6 to identify the failing function, and continue.
   a. Use this table together with the associated entry in the ISPF log to isolate the failure source. "ISMF's use of the ISPF log" on page 226. Locate the SERVICE item in the failure-related log entry. Each SERVICE has its own description for failure-related return codes and reason codes.
   b. If the job control language (JCL) that ISMF created for a DFSMSdss or ICKDSF function is correct, or if DFSMShsm is in control, and the failure occurs in either DFSMSdss or DFSMShsm, use *z/OS DFSMSdss Storage Administration* or *z/OS DFSMShsm Diagnosis*, as appropriate. If the failure occurs in DFSMSrmm, use the *z/OS DFSMSrmm Diagnosis Guide*.
   c. If the failure involves a reason code associated with one of the DASD common services, use the DASD common services-specific sections of the keyword build procedures and see Chapter 8, "DASD common services diagnostic aids," on page 187 to gather additional failure-related information.
   d. To begin developing the keyword string, go to "Type-of-failure keyword" on page 7.

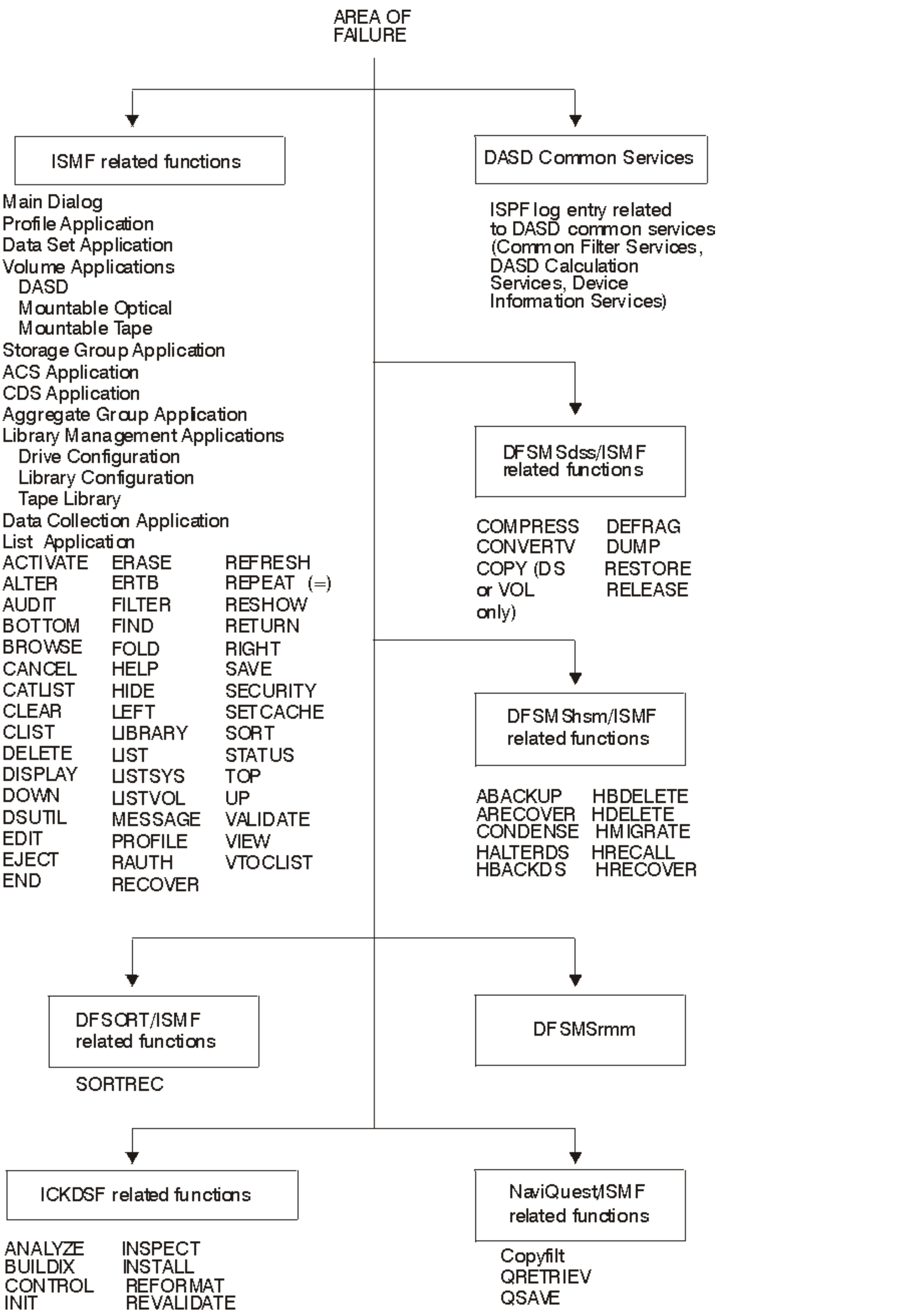


Figure 1. ISMF Failure Isolation Aid

Type-of-failure keyword

The type-of-failure keyword is used to identify an external symptom of a program failure. This keyword is required. Use Table 1 on page 7 to determine the type-of-failure keyword that best describes your problem, and then proceed as directed to construct that keyword.

Table 1. Types of Failures

Type of Failure	Description	Procedure
Abend	Abnormal termination indicated by: • The printed system output of a program • A system message's text • An ISMF abend panel • An ISPF abend panel • A TSO/E message identifying an abend • A SYS1.LOGREC record • An SVC dump in SYS1.DUMP	See "Abend keyword" on page 7.
Wait/Loop	Program unexpectedly suspended; indicated by: • No program response • Repeating messages • Repeating sequence of ISMF panels • System abends 122, 222, 322, 522, 722, A22, C22	See "Wait and loop keywords" on page 13. For the system abends, see *z/OS MVS System Codes* for a description of these abends.
Message	Error indicated by a system message	See "Message keyword" on page 17.
Incorrect Output	• Incorrect or missing output from a program • Incorrect ISMF panel flow or information	See " Incorrect output keyword" on page 27.
Performance	Performance less than what is expected	See "Performance keyword" on page 32.
Documentation	Incorrect or incomplete documentation	See "Documentation keyword" on page 33.

Abend keyword

Use this section when your program (or ISMF session) abnormally ends (abends).

Symptoms of the failure

You can identify an abend by one or more of the following indicators:

- The printed system output of a program
- A system message's text
- An ISMF abend panel
- An ISPF abend panel
- A TSO/E message identifying an abend condition
- A SYS1.LOGREC record
- An SVC DUMP

The means by which the system indicates an abend condition provides sufficient evidence (message prefix or text, operation performed, module that detected the failure, ISMF abend panel, and so forth) to determine which DFSMSdfp component received the abend.

A damaged Virtual Storage Access Method (VSAM) data set can cause an ABEND0C4 in any of these modules (repairing the data set resolves the problem):

- IDA019RC
- IDA019RE
- IDA019RF
- IDA019RG
- IDA019RH
- IDA019RI
- IDA019RJ
- IDA019RN
- IDA019RW
- IDA019R4

To determine whether you have a damaged data set, use the IDCAMS EXAMINE command as described in the topic on functional command format in *z/OS DFSMS Access Method Services Commands* and the topic on checking a VSAM key-sequenced data set cluster for structural errors in *z/OS DFSMS Using Data Sets*. The EXAMINE command provides details about the nature of data set damage.

For more information on diagnosing problems with damaged VSAM RLS data sets, refer to “VSAM record management (R/M) diagnostic aids” on page 593.

Procedure

When an abend is encountered, the system produces one or more of the following kinds of documentation: an SVC dump, a SYSABEND, SYSMDUMP, or a SYSUDUMP. To determine the ABEND CODE, go to the procedure indicated in the following table.

Subcomponent	Procedure
ISMF/ISPF abend panel	See “ISMF/ISPF—Abend panels” on page 11.
TSO/E message	See “TSO/E messages for ISMF abends” on page 12.
All other DFSMSdfp subcomponents	Continue with the next section, “Procedure for SVC dump” on page 8.

Procedure for SVC dump

SVC dumps invoked by the SDUMP macro are usually written as a result of an entry into a *functional recovery routine* (FRR) or ESTAE routine. The *component recovery routine* specifies the addresses that are dumped and directs the dump to one of the SYS1.DUMP*xx* data sets. The SVC dump contains enough information for you to build the keyword string.

You can find the structured search keywords in the Summary Diagnostic Worksheet Figure 2 on page 10 under the section RETAIN SEARCH ARGUMENT. Use these keywords in freeform searches.

If the abend code is X'08B', then SMS has experienced a "data in virtual" (DIV) abend. Do the following:

- Obtain the registers from the time of abend, using either the `IGD300I` message or the system diagnostic work area (SDWA).
- Examine the contents of register 15. The two low-order bytes contain the DIV reason code related to the abend. Append the reason code to the keyword prefix RC. and record it on the “Keyword worksheet” on page 4.

See the description of the applicable DIV reason code listed under abend code 08B in *z/OS MVS System Codes*. It might help you define more closely the source of the failure. If it indicates that the problem is external to DFSMSdfp, continue the diagnosis process within the component involved.

If the abend code is X'0F4', then an error occurred during program management binder, DCME, HFS, PDSE, or VSAM RLS processing. Do the following:

- Review SYS1.LOGREC for X'0F4' software records, PDSE symptom records, and any other records produced at the time of the error. Program management binder, DCME, HFS, PDSE, and VSAM RLS symptom records are primarily used to identify incidents identified with program management binder, DCME, HFS, PDSE, and VSAM RLS X'0F4' abends. When program management binder, DCME, HFS, PDSE, or VSAM RLS symptom records occur without an X'0F4', use the symptom strings to search for matching problems in the IBM software support database, and if no errors exist, contact the IBM Support Center.
- Prior to the ABEND error, a return code is placed in general register 15, and a unique reason code is placed in general register 0 describing the exceptional condition. Append the reason code from general register 0 to the keyword prefix RSN and record it as a modifier keyword on the “Keyword worksheet” on page 4. Remove the leading zeroes from the return code from general register 15 and append it to the keyword prefix RC. Record it as a modifier keyword on the “Keyword worksheet” on page 4. Using the information from Figure 2 on page 10 as an example, the modifier keywords would be as follows:

  ```
  abend0f4   rsn21042716  rc24
  ```

  PDSE reason codes are used primarily for searching the IBM software support database; therefore, reason code descriptions are not documented in this information.

To determine keywords for SVC dumps, do the following:

1. Use IPCS to print the summary dump (SUMDUMP). See *z/OS MVS IPCS User's Guide*.
2. The title page of the Summary Diagnostic Worksheet contains the dump header and title page, which provide failure-related symptoms extracted from the dump. (See the example in Figure 2 on page 10.) One or more of the following symptoms should be present:
   - ABEND*nnn*
   - Module or CSECT name or both
   - Component Identifier
   - Release Level
   - Service Level
   - FMID

   Refer to *z/OS Problem Management*, *z/OS MVS Diagnosis: Reference*, and *z/OS MVS Diagnosis: Tools and Service Aids* for a detailed explanation of each symptom.
3. If you can identify the ABEND CODE by using the dump header and title page, see “Building the Abend keyword” on page 11.
4. If the dump does not have a header title or does not otherwise enable you to identify the ABEND CODE, use the SUMDUMP printed from the SYS1.DUMP*xx* data set and continue with “Procedure for SYSABEND, SYSMDUMP, or SYSUDUMP” on page 10.

```
TITLE
LIST 00000000 HEADER POSITION(:hex.0058:ehex.) LENGTH(97) CHARACTER
HEADER ADDRESS(00000058)
00000058 |                          COMPID=DF115,CSECT=IGWBVLP1+0F0C,DATE=12 |
00000080 | /20/04,MAINTID=NONE    ,ABND=0F4,RC=00000024,RSN=21042716        |
                        * * * *  S Y M P T O M  * * * *
  Primary Symptom String:
    RIDS/IGWBBMF1#L RIDS/IGWBVLP1 PIDS/5695DF115 AB/S00F4 RIDS/IGWBSGLR#R VALU/HD1641854 REGS/0E010 REGS/0CEE4
    PRCS/00000024
    Symptom            Symptom data       Explanation
    ---------------    -------------      -----------
    RIDS/IGWBBMF1#L    IGWBBMF1#L         Routine identifier
    PIDS/5695DF115     5695DF115          Component identifier
    AB/S00F4           00F4               ABEND code - system
    RIDS/IGWBSGLR#R    IGWBSGLR#R         Routine identifier
    VALU/HD1641854     D1641854           Error related hexadecimal value
    REGS/0E010         0E010              Program register
    PRCS/00000024      00000024           Return code
The dump does not contain a secondary symptom string.
         ****** DUMP ANALYSIS AND ELIMINATION (DAE) ******
THIS DUMP WAS NOT SUPPRESSED BECAUSE
DAE WAS NOT CHECKING FOR PREVIOUS OCCURRENCES.
CRITERIA FOR USE AS A UNIQUE DUMP IDENTIFIER BY DAE:
     MINIMUM NUMBER OF SYMPTOMS:  07  FOUND: 09
     MINIMUM TOTAL STRING LENGTH: 025 FOUND: 130
     SYMPTOMS REQUIRED TO BE PRESENT: MOD/ CSECT/
     SYMPTOMS THAT ARE TO BE USED IF AVAILABLE, BUT ARE NOT REQUIRED:
       PIDS/ AB/S AB/U REXN/ FI/ REGS/ HRC1/ SUB1/
MVS SYMPTOM STRING:
MOD/IGWBBMF1 CSECT/IGWBVLP1 PIDS/5695DF115 AB/S00F4 REXN/IGWBSGLR
FI/9C7918F60A0D4140D1641854 REGS/0E010 REGS/0CEE4 HRC1/00000024
RETAIN SEARCH ARGUMENT:
RIDS/IGWBBMF1#L RIDS/IGWBVLP1 PIDS/5695DF115 AB/S00F4 RIDS/IGWBSGLR#R
VALU/HD1641854 REGS/0E010 REGS/0CEE4 PRCS/00000024
SYMPTOMS PRESENT FOR USE AS A UNIQUE DUMP IDENTIFIER BY DAE: RETAIN
MVS KEY   KEY     SYMPTOM DATA               EXPLANATION
-------  ------   ------------               -----------
MOD/     RIDS/    IGWBBMF1                   LOAD MODULE NAME
CSECT/   RIDS/    IGWBVLP1                   ASSEMBLY MODULE CSECT NAME
PIDS/    PIDS/    5695DF115                  PRODUCT/COMPONENT IDENTIFIER
AB/S     AB/S     S00F4                      ABEND CODE-SYSTEM
REXN/    RIDS/    IGWBSGLR                   RECOVERY ROUTINE CSECT NAME
FI/      VALU/H   9C7918F60A0D4140D1641854   FAILING INSTRUCTION AREA
REGS/    REGS/    0E010                      REG/PSW DIFFERENCE
REGS/    REGS/    0CEE4                      REG/PSW DIFFERENCE
HRC1/    PRCS/    00000024                   REASON CODE
ADDITIONAL SYMPTOM DATA NOT USED BY DAE TO IDENTIFY THIS DUMP: RETAIN
MVS KEY   KEY     SYMPTOM DATA               EXPLANATION
-------  ------   ------------               -----------
CID1/    VALU/C   DF115                      COMPONENT IDENTIFIER
AMD1/    VALU/C   12#20#91                   MODULE ASSEMBLY DATE
VRS1/    VALU/C   NONE                       VERSION-PRODUCT/PTF IDENTIFIER
RRL1/    FLDS/    IGWFRCSD                   RECOVERY ROUTINE LABEL
CDB1/    VALU/C   5695                       BASE COMPONENT IDENTIFIER
FRR1/    VALU/H   C9C7E6C6C5C6D7407F6810287F681250
                                             FRR PARAMETER AREA
ASID1/   VALU/H   004D                       TASK RELATED ASID
ORCC1/   PRCS/    0F4000                     ORIGINAL COMPLETION CODE
ORRC1/   PRCS/    00000024                   ORIGINAL REASON CODE
@201/    VALU/C   W                          DEVELOPER ASSIGNED SYMPTOM KEY
@201/    VALU/C   H                          DEVELOPER ASSIGNED SYMPTOM KEY
@202/    VALU/C   ##                         DEVELOPER ASSIGNED SYMPTOM KEY
@202/    VALU/C   #COMPID#DF115#CSECT#IGWBVLP1#0F0C#DATE#12#20#
                                             DEVELOPER ASSIGNED SYMPTOM KEY
```

Figure 2. Sample Summary Diagnostic Worksheet

Procedure for SYSABEND, SYSMDUMP, or SYSUDUMP

Depending on the JCL used, the system directs a dump to the SYSUDUMP, SYSABEND, or SYSMDUMP data set. If the system did not produce a dump, you might need to recreate the failure and obtain one. For information about obtaining a dump, see *z/OS MVS Diagnosis: Tools and Service Aids*.

Note: SYSUDUMP data sets usually do not contain enough information to be useful in diagnosing a failure.

1. Obtain a system storage dump that contains the user's program.
2. Determine the system abend code by using either:
   - The job-related information about the abend in the job log.

     This information includes the abend code, PSW contents, and general-purpose register contents. The abend code is 3 characters long. To obtain the job log, you must specify the JCL parameter `MSGLEVEL=(1,1)` on your JCL JOB card.
   - The system storage dump.

- – Locate the formatted section at the beginning of the dump. Determine the abending job by locating the job whose abend code field (TCBCMP at TCB + X'11') contains a nonzero value.
- – The field is only 3 characters long. Ignore the first (left-most) byte. If the abend code appears in the first 12 bits following the first byte, it is a system abend code. If it appears in the next 12 bits, it is a user abend code, and the value must be converted to decimal.

  For more information on dump documentation and analysis, see *z/OS MVS Diagnosis: Tools and Service Aids*.

3. When you determine the ABEND CODE, continue with "Building the Abend keyword" on page 11.

Building the Abend keyword

To build the abend keyword, do the following:

1. Use the ABEND CODE that you have extracted from the system-produced documentation.

   System abends are expressed in hexadecimal; user abends are expressed in decimal.

   - For a system abend, append the 3-character code to the keyword prefix ABEND.

     Example: If the abend code is 0C4, specify the abend type-of-failure keyword as follows:

     ```
     ABEND0C4
     ```

   - For a user abend, append the 4-digit code to the keyword prefix ABENDU.

     Example: If the abend code is 0222, specify the abend type-of-failure keyword as follows:

     ```
     ABENDU0222
     ```

2. If a message containing a return code accompanies the abend, include the return code in your keyword string as a modifier keyword. Append the code (specified exactly as it appears in the message) to the keyword prefix RC.

   Example: If the return code is 04, specify the keyword string as follows:

   ```
   ABEND0C4 RC04
   ```

   Tip: If the *z/OS MVS System Codes* description of the abend code indicates a return code and the reason code is associated with the abend, including both the return code and the reason code in your keyword string could restrict the scope of a software database search that results in no matches.

3. If the *z/OS MVS System Codes* description of the abend code indicates a reason code is associated with the abend, include the reason code in your keyword string as a modifier keyword. Append the code (specified exactly as it appears in the register) to the keyword prefix RSN.

   Example: If the abend code is X'0F4', then the reason code is found in general register 0. If the reason code is 0409F023, specify the keyword string as follows:

   ```
   ABEND0F4 RSN0409F023
   ```

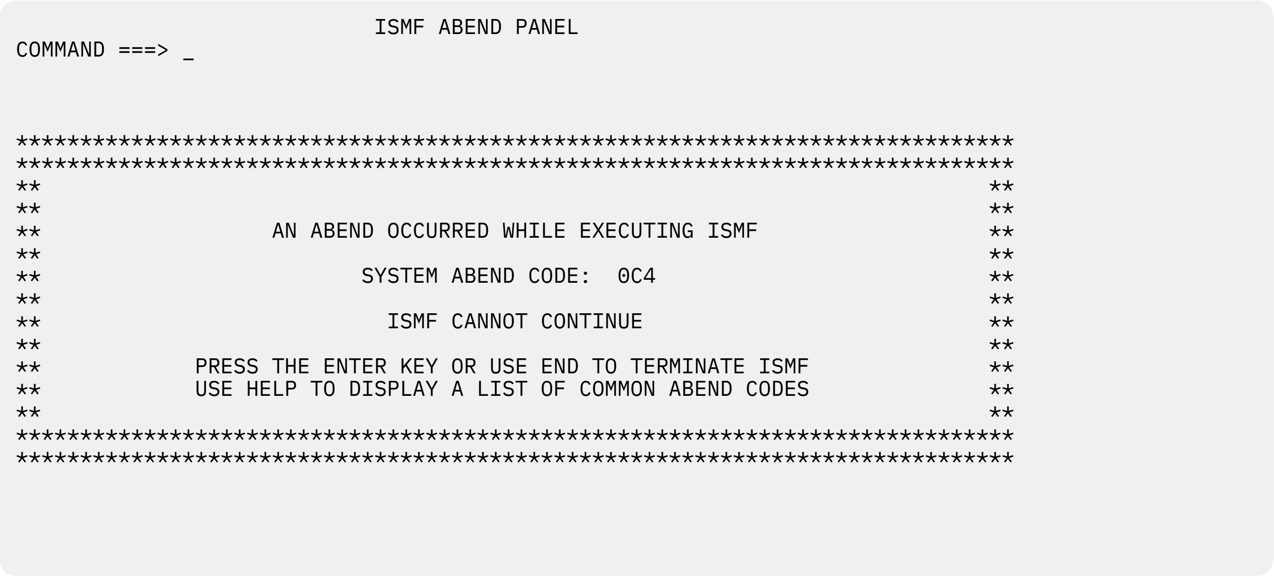
4. See "Module keyword" on page 33 (directly to the subcomponent-specific section, if one exists).

ISMF/ISPF—Abend panels

Use this section when your program ends abnormally, and ISMF or ISPF seems to be the cause of the failure.

You can recognize an abend failure from one of the following symptoms:

- An ISMF or ISPF abend panel appears on the screen, providing the ABEND CODE. Figure 3 on page 12 , which follows, and Figure 4 on page 12 show sample abend panels. Record the ABEND CODE and see "Building the Abend keyword" on page 11.
- A TSO/E message appears, indicating an abend condition. See "TSO/E messages for ISMF abends" on page 12.

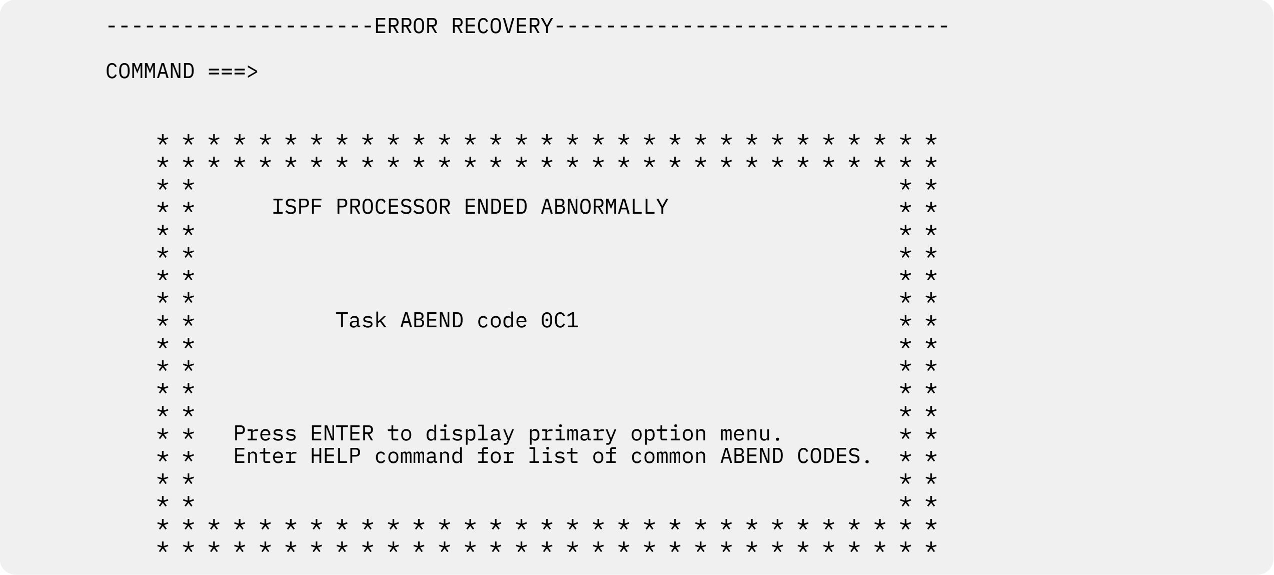


```
                            ISMF ABEND PANEL
COMMAND ===> _



******************************************************************************
******************************************************************************
**                                                                          **
**                                                                          **
**                  AN ABEND OCCURRED WHILE EXECUTING ISMF                  **
**                                                                          **
**                         SYSTEM ABEND CODE:  0C4                          **
**                                                                          **
**                           ISMF CANNOT CONTINUE                           **
**                                                                          **
**            PRESS THE ENTER KEY OR USE END TO TERMINATE ISMF              **
**            USE HELP TO DISPLAY A LIST OF COMMON ABEND CODES              **
**                                                                          **
******************************************************************************
******************************************************************************
```

Figure 3. ISMF—Abend Panel



```
       ---------------------ERROR RECOVERY-------------------------------

       COMMAND ===>

           * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
           * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
           * *                                                       * *
           * *      ISPF PROCESSOR ENDED ABNORMALLY                  * *
           * *                                                       * *
           * *                                                       * *
           * *                                                       * *
           * *                                                       * *
           * *           Task ABEND code 0C1                         * *
           * *                                                       * *
           * *                                                       * *
           * *                                                       * *
           * *                                                       * *
           * *   Press ENTER to display primary option menu.         * *
           * *   Enter HELP command for list of common ABEND CODES.  * *
           * *                                                       * *
           * *                                                       * *
           * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
           * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Figure 4. ISMF—ISPF Abend Panel

TSO/E messages for ISMF abends

When ISMF terminates abnormally and ISPF is in TEST/TRACE mode, the system issues a TSO/E message that indicates the failure.

Note: This is not an ISMF message. See Figure 5 on page 13 for an example of a TSO/E message.

Some problems cannot be recreated, so you might not always be able to get a dump or develop a full keyword string that describes the problem. You can, however, build the abend keyword by using the symptom dump displayed on your terminal screen and following the instructions in "Building the Abend keyword" on page 11.

```
* SPF  SUBTASK  ABEND *
ISPF      ENDED DUE TO ERROR+
READY
```

Figure 5. ISMF—TSO/E Error Message

Procedure

1. The plus sign (+) at the end of the message indicates that additional information is available. If you enter a question mark (?), TSO/E displays a message describing the error in greater detail. The question mark must be the next command from the terminal, or the supplemental information is lost.
2. To further diagnose the error, you need a dump of the abend. Press ENTER to generate the dump. If you have WTP message turned on in your TSO/E profile, the system displays a symptom dump on the terminal screen. (For information about the TSO/E profile, see *z/OS TSO/E Command Reference*.) Figure 6 on page 13 shows a sample symptom dump.

   ```
   IEA995I  SYMPTOM DUMP OUTPUT
    ABEND CODE  USER = 0222  TIME = 15.29.00 SEQ=22374 CPU = 000 ASID =007B
    PSW AT TIME OF ERROR  078D1000   00B5A992      ILC 2  INTC 0D
      NO ACTIVE MODULE FOUND
      DATA AT PSW  00B5A98C - 00181610  0A0D9180  A0644710
      GPR  0-3    80000000   800000DE   0000E060   000184A8
      GPR  4-7    000184AC   00018898   00000005   00000005
      GPR  8-11   00010B28   0000E810   0000DC30   00B5A788
      GPR 12-15   91B59788   000193D0   000185C7   00000000
    END OF SYMPTOM DUMP
   **LOGICAL SCREEN REQUEST FAILED - ABEND 0000DE**
   **CONTACT YOUR SYSTEMS PROGRAMMER OR DIALOG DEVELOPER**
   ***SPF MAIN TASK ABEND***
   ISPF    ENDED DUE TO ERROR+
   READY
   ```

   Figure 6. ISMF—Displayed System Symptom Dump

   If a SYSUDUMP, SYSMDUMP, or SYSABEND data set was allocated before the error, the system directs a full abend dump to the dump data set. You will use the dump later to determine the module keyword.
3. See "Building the Abend keyword" on page 11.

   Tip: If you did not have a dump data set allocated before the abend occurred, you can do the following after you build the abend keyword:

   a. Allocate a dump data set by using the TSO ALLOC command.
   b. Invoke ISPF with the TEST parameter by issuing the ISPF TEST command.
   c. Specify `RECOVER FROM ABENDS ===> N` in your ISMF profile.
   d. Try to recreate the problem by going through your ISMF session again.

Wait and loop keywords

Use this section when a DFSMSdfp function appears to be waiting or looping.

Symptoms of the failure

You can identify a wait or loop condition by means of one of the following conditions:

- Suspended activity while waiting for some condition to be satisfied

- Endless instruction loop
- Repetition of messages
- Repetition of a sequence of ISMF panels
- No system response; for example, an ISMF panel remains on the terminal screen indefinitely after the operator has taken action that should cause it to change

Because wait and loop failures are often hard to differentiate, this section documents both keywords.

Procedure

1. The failure indications should enable you to establish whether ISMF is involved or not. If you suspect a failure in ISMF, see "ISMF—Wait and loop keywords" on page 14.
2. For all other DFSMSdfp components, obtain the following:
   - A console dump initiated by the MVS DUMP command before the operator cancels the job.
   - A copy of the SYS1.LOGREC data set. Use IEBGENER to copy the SYS1.LOGREC data set to a standard label tape. This data set contains the event history leading up to the wait problem.
3. Using the dump, scan the RBs to see if a WAIT SVC was issued. The INTCODE field of the RB contains 0001 for jobs that issued a WAIT SVC, if the high-order byte of the LINK field of the RB contains a nonzero value, indicating the number of events being waited or suspended on.
4. If these fields do not indicate a WAIT condition, assume that the failure is a LOOP.
5. In either case, record the current PSW address in the waiting or looping RB's OPSW field as the failing address. You will use it to determine the failing module. Specify either the WAIT keyword or the LOOP keyword.

   You might find the following information useful in isolating the source of a LOOP failure to a specific module:

   - If the FLCDE field for the RB is nonzero, it points to a cde entry of the entry point of a load module. That load module might contain the failing CSECT.
   - A system branch trace (invoked by the operator from the console). This makes an entry in the system trace table for every successful branch instruction. (See *z/OS MVS Diagnosis: Tools and Service Aids* for the format of the system trace table.)
   - An AMBLIST listing or SPZAP dump of a CSECT or CSECTs that you suspect (see *z/OS MVS Diagnosis: Tools and Service Aids*).
6. A wait on the ENQ and DEQ major resource names IGDCDS, IGDCDSXS, or SYSZIGDI, indicates that the Storage Management Subsystem (SMS) is involved in the failure. Record the indicated major resource name and the minor resource name as modifier keywords on the "Keyword worksheet" on page 4.
7. A wait on the ENQ/DEQ major resource names SYSZIGW0 and SYSZIGW1 indicates that the program manager binder, HFS, or PDSE is involved in the failure. Record the indicated major resource name and the minor resource name as modifier keywords on the "Keyword worksheet" on page 4.
8. For VSAM Record Management, see "VSAM record management—wait and loop keywords" on page 16.
9. For all other DFSMSdfp components, see "Module keyword" on page 33.

ISMF—Wait and loop keywords

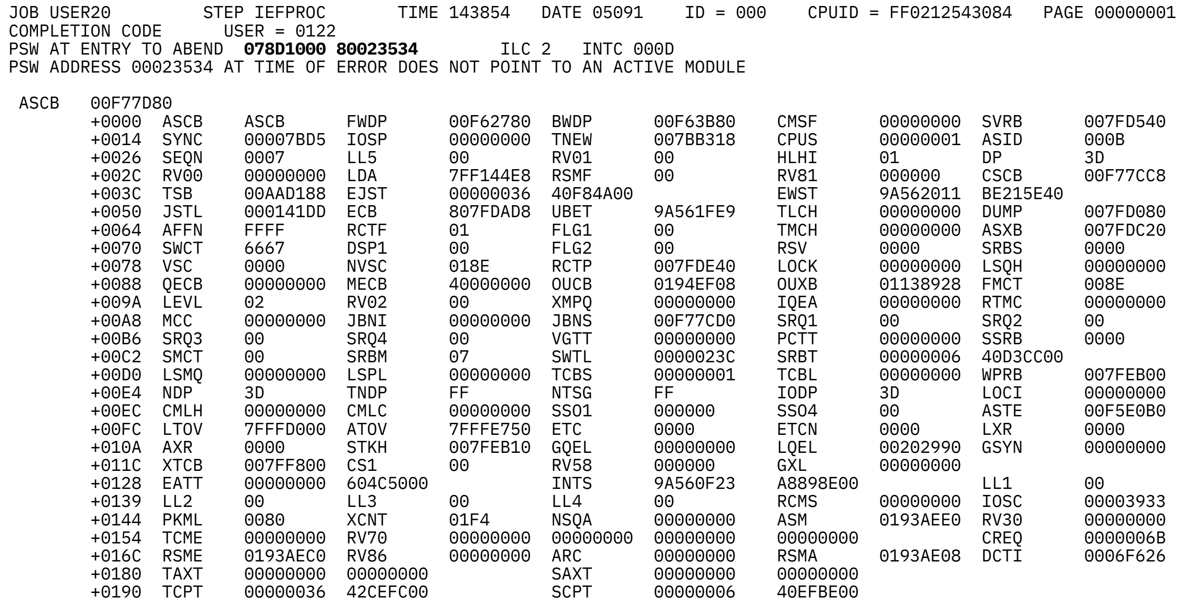
Use this section when a wait or loop condition seems to be occurring and you suspect ISMF of causing a failure.

If a request from a selection panel seems to be taking an abnormally long time to process, the delay could be related to legitimate selection criteria you have specified. Because the entries in the list generated must meet all of the criteria you establish on the selection panels, each value you specify must be verified before the list can be generated. To speed up processing you can try the following:

- Limit the number of entries in the list by making your selection criteria more specific.
- For data set selection, the time it takes to build a list is affected by the number of times ISMF must access the catalog or VTOC. If you are generating the list from the catalog, observe the following rules:
  - Do not acquire data for data sets migrated by DFSMShsm.
  - Do not acquire data from the volume.

Procedure

1. If you have made the distinction between WAIT and LOOP keywords, identifying the failure as either one or the other, specify the type-of-failure keyword as either WAIT or LOOP and see "ISMF—modifier keywords" on page 51. Otherwise, continue with "2" on page 15.
2. If you know the failure is either a wait or a loop, but cannot determine which, you need a dump to diagnose the problem. If the system has not produced a dump, see "Determining how ISMF processes abends and generates dumps" on page 221. If you have a dump, continue here.
3. Use IEBGENER to copy the SYS1.LOGREC data set to a standard label tape. This data set contains the event history leading up to the wait problem.
4. Find the Program Status Word (PSW) in the dump. See Figure 7 on page 15.



```
JOB USER20         STEP IEFPROC       TIME 143854   DATE 05091    ID = 000    CPUID = FF0212543084   PAGE 00000001
COMPLETION CODE      USER = 0122
PSW AT ENTRY TO ABEND  078D1000 80023534        ILC 2   INTC 000D
PSW ADDRESS 00023534 AT TIME OF ERROR DOES NOT POINT TO AN ACTIVE MODULE

 ASCB   00F77D80
        +0000  ASCB    ASCB      FWDP      00F62780  BWDP      00F63B80    CMSF      00000000  SVRB      007FD540
        +0014  SYNC    00007BD5  IOSP      00000000  TNEW      007BB318    CPUS      00000001  ASID      000B
        +0026  SEQN    0007      LL5       00        RV01      00          HLHI      01        DP        3D
        +002C  RV00    00000000  LDA       7FF144E8  RSMF      00          RV81      000000    CSCB      00F77CC8
        +003C  TSB     00AAD188  EJST      00000036  40F84A00              EWST      9A562011  BE215E40
        +0050  JSTL    000141DD  ECB       807FDAD8  UBET      9A561FE9    TLCH      00000000  DUMP      007FD080
        +0064  AFFN    FFFF      RCTF      01        FLG1      00          TMCH      00000000  ASXB      007FDC20
        +0070  SWCT    6667      DSP1      00        FLG2      00          RSV       0000      SRBS      0000
        +0078  VSC     0000      NVSC      018E      RCTP      007FDE40    LOCK      00000000  LSQH      00000000
        +0088  QECB    00000000  MECB      40000000  OUCB      0194EF08    OUXB      01138928  FMCT      008E
        +009A  LEVL    02        RV02      00        XMPQ      00000000    IQEA      00000000  RTMC      00000000
        +00A8  MCC     00000000  JBNI      00000000  JBNS      00F77CD0    SRQ1      00        SRQ2      00
        +00B6  SRQ3    00        SRQ4      00        VGTT      00000000    PCTT      00000000  SSRB      0000
        +00C2  SMCT    00        SRBM      07        SWTL      0000023C    SRBT      00000006  40D3CC00
        +00D0  LSMQ    00000000  LSPL      00000000  TCBS      00000001    TCBL      00000000  WPRB      007FEB00
        +00E4  NDP     3D        TNDP      FF        NTSG      FF          IODP      3D        LOCI      00000000
        +00EC  CMLH    00000000  CMLC      00000000  SSO1      000000      SSO4      00        ASTE      00F5E0B0
        +00FC  LTOV    7FFFD000  ATOV      7FFFE750  ETC       0000        ETCN      0000      LXR       0000
        +010A  AXR     0000      STKH      007FEB10  GQEL      00000000    LQEL      00202990  GSYN      00000000
        +011C  XTCB    007FF800  CS1       00        RV58      000000      GXL       00000000
        +0128  EATT    00000000  604C5000            INTS      9A560F23    A8898E00            LL1       00
        +0139  LL2     00        LL3       00        LL4       00          RCMS      00000000  IOSC      00003933
        +0144  PKML    0080      XCNT      01F4      NSQA      00000000    ASM       0193AEE0  RV30      00000000
        +0154  TCME    00000000  RV70      00000000  00000000  00000000    00000000            CREQ      0000006B
        +016C  RSME    0193AEC0  RV86      00000000  ARC       00000000    RSMA      0193AE08  DCTI      0006F626
        +0180  TAXT    00000000  00000000            SAXT      00000000    00000000
        +0190  TCPT    00000036  42CEFC00            SCPT      00000006    40EFBE00
```

Figure 7. ISMF—Locating the PSW in an ISMF Dump

5. Determine whether the WAIT bit (bit 14) of the PSW is on or off. For example, in Figure 7 on page 15, the PSW is `078D1000 80023534`. The hexadecimal digit `D` is bits 12 through 15. Expressed in bits, this value is `1101`. Bit 14 (the wait state bit) is off. If Bit 14 is off, specify the type-of-failure keyword as LOOP.
6. If Bit 14 is on, specify the type-of-failure keyword as WAIT.
7. See "ISMF—modifier keywords" on page 51.

Object access method—wait and loop keywords

Use this section when you feel that a wait or a loop seems to be occurring and you suspect OAM is causing the failure.

OAM storage management component—wait and loop keywords

Use the information presented below to identify a potential system wait or loop. Then contact the IBM Support Center for assistance with analyzing the dump.

OSMC processes storage groups individually and sends a message after each storage group has completed processing:

- The message `CBR9048I` is displayed upon successful processing of a storage group.
- The message `CBR9049I` is displayed upon unsuccessful processing of a storage group.

If the system does not issue a message, the system could be waiting or looping.

To diagnose the problem, you must generate a dump of the OAM address space by issuing the DUMP command. The following entries should be produced in the dump:

```
SDATA=(TRT,LPA,CSA,PSA,GRSQ,SUM,SQA,ALLNUC,LSQA,RGN,SWA)
```

Refer to *z/OS MVS System Commands* for information on how to generate a dump.

After producing a dump, use IEBGENER to copy the SYS1.LOGREC data set to a standard label tape. This data set contains the event history leading up to the wait problem.

Recommendation: If you have the DFSORT product installed, you should be using ICEGENER as an alternative to IEBGENER when making an unedited copy of a data set or member. It might already be installed in your system under the name IEBGENER. It generally provides better performance.

VSAM record management—wait and loop keywords

Use this section to determine whether additional analysis is required when either a wait or loop condition occurs and you suspect the VSAM Record Management subcomponents of causing the failure. This section does not apply to VSAM RLS.

Procedure

1. Some wait and loop failures involve reading or writing data to a damaged VSAM data set. To determine whether you have a damaged data set, use the IDCAMS EXAMINE command as described in the topic on functional command format in *z/OS DFSMS Access Method Services Commands* and the topic on checking a VSAM key-sequenced data set cluster for structural errors in *z/OS DFSMS Using Data Sets*. The EXAMINE command provides details about the nature of data set damage.

   Waits or loops can occur in several VSAM modules because of a damaged VSAM data set. The most commonly involved modules are:

   - IDA019RA
   - IDA019RI
   - IDA019R2
   - IDA019RB
   - IDA019RJ
   - IDAM19R3 (IDA019R3)
   - IDA019RC
   - IDA019RN
   - IDA019SE
   - IDA019RE
   - IDA019RW
   - IDA019SY
   - IDA019RH
   - IDA019RY

   A loop might also be indicated by the reissuing of either SVC121 (X'79') or Start Subchannel (SSCH). If an apparent wait occurs and a console dump is taken, the following symptom might indicate the involvement of a damaged data set:

   - SVC01 WAITs issued by module IDA019RZ

2. Use IEBGENER to copy the SYS1.LOGREC data set to a standard label tape. This data set contains the event history leading up to the wait problem.
3. See "Module keyword" on page 33.

Message keyword

Use this section for all DFSMSdfp message-related problems.

Symptoms of the failure

You can identify a message type-of-failure when one of the following conditions occurs:

- Message reports program or operation failure.
- Message is missing data, or contains invalid data.
- Message reports a data failure (catalog, user data).
- No message appears when one should have been issued.

Procedure

Before using this section, examine *z/OS MVS System Codes* and *z/OS MVS System Messages, Vol 1 (ABA-AOM)*, through *z/OS MVS System Messages, Vol 10 (IXC-IZP)*. These might help you generate additional keywords by identifying failure-related functions and providing message-to-module cross-reference tables.

Definition of message keyword related terms

The component-specific message keyword sections use the terms defined in Table 2 on page 17.

Table 2. Definition of Message Keyword Related Terms

Term	Definition
Message identifier	A three-letter prefix to identify the component that produced the message and a message serial number to identify the individual message (for example, IDC3009I).
Message keyword prefix	The characters MSG, to which the message identifier is appended. This comprises the message type-of-failure keyword.
Return or reason code	A numeric code contained in the message text. Either the message text or *z/OS MVS System Messages, Vol 1 (ABA-AOM)*, through *z/OS MVS System Messages, Vol 10 (IXC-IZP)*, can identify the type of code.
Return or reason code keyword prefix	The characters RC, to which each return or reason code (exactly as it appears in the message) is appended. (Each code in the text requires its own keyword prefix.) This comprises a modifier keyword to specify the failure-related symptom.

Go to one of the procedures indicated in the following table.

Subcomponent	Procedure
DADSM/CVAF	See "DADSM/CVAF—message keyword" on page 18.
ISMF	See "ISMF—message keyword" on page 20.
OAM	See "Object Access Method—message keyword" on page 23.

Subcomponent	Procedure
SMS	See "Storage Management Subsystem—message keyword" on page 24.
VSAM Record Management	See "VSAM, transactional VSAM and VSAM RLS record management—message keyword" on page 27.
All other DFSMSdfp components	Continue below.

1. Append the message identifier to the keyword prefix MSG. Include in the keyword string any return codes and reason codes from the message text. Append the codes, exactly as they appear in the message to the keyword prefix RC.

   Example: If the message identifier is IDC3009I, the return code is 04, and the reason code is 032, specify the keyword string as follows:

   ```
   MSGIDC3009I RC04 RC032
   ```

   Rule: For PDSE-related failures (IGW messages), use RSN as the prefix for reason codes and RC as the prefix for return codes.
2. Message text might contain additional information that you can use as modifier keywords (function, subfunction, device-related information, and so forth); record it on the "Keyword worksheet" on page 4.
3. For input/output or hardware-related errors, review the SYS1.LOGREC for keyword information.
4. See "Module keyword" on page 33.

DADSM/CVAF—message keyword

Use this section when a message indicates a failure in DADSM/CVAF.

Procedure

Before using this section, examine *z/OS MVS System Codes* and *z/OS MVS System Messages, Vol 1 (ABA-AOM)*, through *z/OS MVS System Messages, Vol 10 (IXC-IZP)*. These might help you generate additional keywords by identifying failure-related functions and providing message-to-module cross-reference tables.

See "Definition of message keyword related terms" on page 17 for definitions of the following terms used in this section:

Message identifier
Message keyword prefix
Return or reason code
Return or reason code keyword prefix

Messages issued by DADSM/CVAF

IEC614I—DADSM issues this message.

1. Append the message identifier and function return code to their appropriate prefixes. Append the 4-byte diagnostic information field to the prefix RSN and record on the "Keyword worksheet" on page 4 all failure-related function modifier keywords.

   Example: Suppose the message is the following one:

   ```
   IEC614I RENAME FAILED - RC4, DIAGNOSTIC INFORMATION IS
   (040D002D),STEP01,VOL0002,DATASET1
   ```

   Specify the keyword string as follows:

   ```
   MSGIEC614I RENAME RC4 RSN040D002D
   ```

2. See “DADSM/CVAF—module keyword” on page 34.

IEC603I—DADSM issues this message.

1. Append the message identifier to the keyword prefix MSG and record it on the “Keyword worksheet” on page 4 as the type-of-failure keyword. Append the cde code to the keyword prefix RC and record it on the “Keyword worksheet” on page 4 as a modifier keyword.

   Example: If the cde field contains 04, specify the keyword string as follows:

   ```
   MSGIEC603I RC04
   ```

2. If a dump entitled “DADSM (OBTAIN, SCRATCH, ALLOCATE, OR EXTEND) ERROR” is taken, then DADSM caused the dump because of an unexpected Common VTOC Access Facility (CVAF) error. Continue below, using “DADSM dumps” on page 185 for CVAF dump analysis assistance.
3. Record either the DADSM function name from the dump title (for example, OBTAIN) or the 2-character code for the CVAF function last issued (and the 2-character subfunction code, if applicable), as indicated by the 1-byte CVFCTN field (offset X'06' in the CVPL), as a modifier keyword. The CVPL is in the DADSM work area. See Table 11 on page 49 for a list of valid CVFCTN field values.

   Example: If the CVFCTN field contains X'07', record the modifier keywords as:

   ```
   IX DEL
   ```

4. See “DADSM/CVAF—module keyword” on page 34.

IEC606I—CVAF issues this message. It is related to CVAF ABEND18B.

1. Append the message identifier to the keyword prefix MSG and record it on the “Keyword worksheet” on page 4 as the type-of-failure keyword. Append the cde code to the keyword prefix RC and record it on the “Keyword worksheet” on page 4 as a modifier keyword.

   Example: If the cde field contains 153, specify the keyword string as follows:

   ```
   MSGIEC606I RC153
   ```

   Continue below, using “DADSM dumps” on page 185 for CVAF dump analysis assistance.
2. If the message type-of-failure keyword search (without the abend keyword) is inconclusive, perform software database searches by specifying keywords as follows:
   - Both ABEND18B and MSGIEC606I together as type-of-failure keywords
   - ABEND18B alone as the type-of-failure keyword
3. Record the CVAF module name identified in the system dump title on the “Keyword worksheet” on page 4.
4. See “DADSM/CVAF—module keyword” on page 34.

IEC608I or IEC609I—If DADSM issued either message and disabled the VTOC index, CVAF issues message IEC606I, error code 153, and causes a system dump.

1. IEC608I—Use the explanation of the message's cde field and the message-to-module cross-reference table in *z/OS MVS System Messages, Vol 7 (IEB-IEE)*, to determine the related module and function. If available, append these symptoms to their respective prefixes and record them on the “Keyword worksheet” on page 4. Continue below, using “DADSM dumps” on page 185 for CVAF dump analysis assistance.
2. IEC609I—Append the single-digit function code from the message text to the prefix IGGVRF0 to determine the related module name. Append the message identifier and the message's cde field to their respective identifiers and record these keywords and the function keyword (identified by the value in the message's `fctn` field) on the “Keyword worksheet” on page 4. Continue below, using “DADSM dumps” on page 185 for CVAF dump analysis assistance.

Example: Append the symptoms to their respective keyword prefixes as shown:

```
MSGIEC608I SCRATCH RC12
```

```
MSGIEC609I IGGVRF01 ALLOCATE RC04
```

3. See “DADSM/CVAF—module keyword” on page 34.

ISMF—message keyword

Use this section when a message indicates an ISMF problem.

Procedure

Before using this section, examine *z/OS MVS System Codes* and *z/OS MVS System Messages, Vol 1 (ABA-AOM)*, through *z/OS MVS System Messages, Vol 10 (IXC-IZP)*. These might help you generate additional keywords by identifying failure-related functions and providing message-to-module cross-reference tables.

See “Definition of message keyword related terms” on page 17 for definitions of the following terms used in this section:

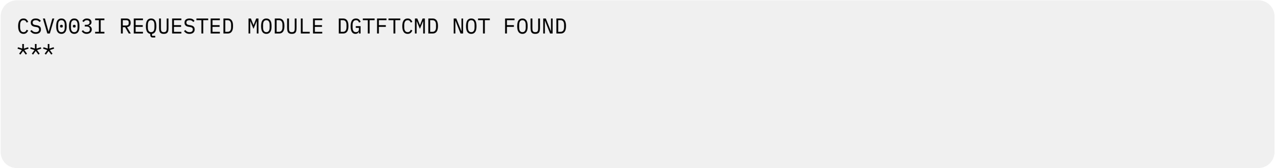
- Message identifier
- Message keyword prefix
- Return or reason code
- Return or reason code keyword prefix

Identifying the error message source

Two types of error messages can occur when you run ISMF:

- Error messages issued by other subcomponents that interrupt the screen

  See Figure 8 on page 20 for an example of a TSO/E message that can interrupt the screen. See “Messages issued by other subcomponents” on page 22 to diagnose the error.



```
CSV003I REQUESTED MODULE DGTFTCMD NOT FOUND
***
```

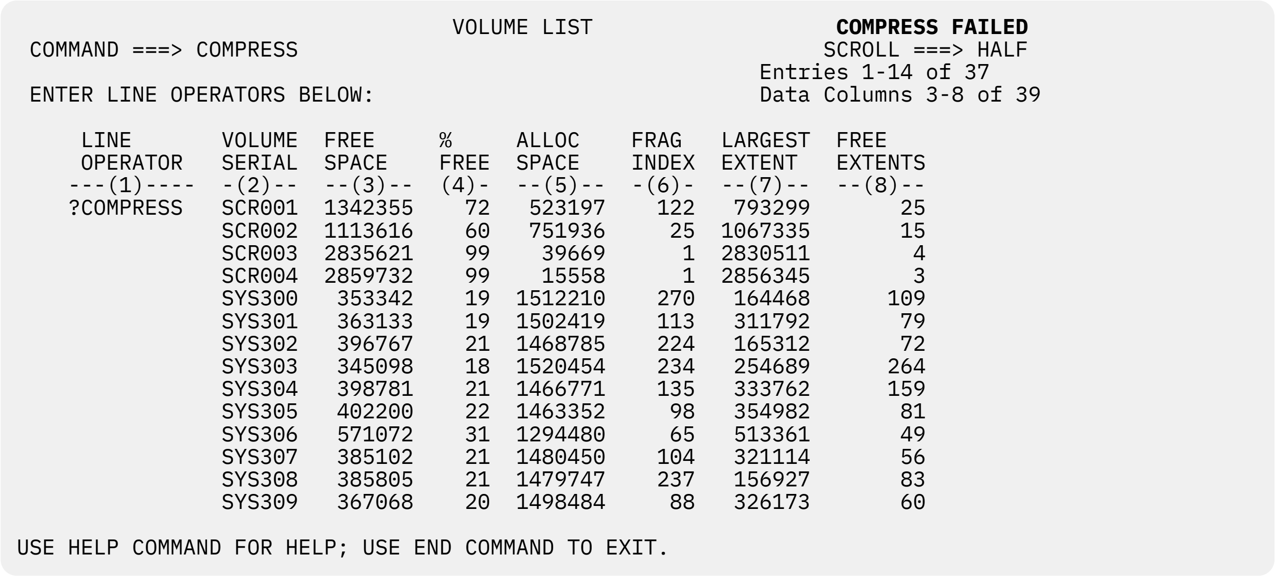
Figure 8. ISMF—TSO/E Messages That Interrupt the Screen

- ISMF or ISPF short error messages

  Figure 9 on page 21 is an example of an end-user ISMF panel with the short message:

  ```
  COMPRESS FAILED
  ```

  ISPF short error messages have the same format. To determine whether ISMF or ISPF issued the error message, see “Identifying an ISMF error message” on page 21.



```
                                 VOLUME LIST                  COMPRESS FAILED
COMMAND ===> COMPRESS                                         SCROLL ===> HALF
                                                         Entries 1-14 of 37
ENTER LINE OPERATORS BELOW:                              Data Columns 3-8 of 39

    LINE       VOLUME  FREE     %     ALLOC    FRAG   LARGEST  FREE
    OPERATOR   SERIAL  SPACE    FREE  SPACE    INDEX  EXTENT   EXTENTS
   ---(1)----  -(2)--  --(3)--  (4)-  --(5)--  -(6)-  --(7)--  --(8)--
   ?COMPRESS   SCR001  1342355    72   523197    122   793299       25
               SCR002  1113616    60   751936     25  1067335       15
               SCR003  2835621    99    39669      1  2830511        4
               SCR004  2859732    99    15558      1  2856345        3
               SYS300   353342    19  1512210    270   164468      109
               SYS301   363133    19  1502419    113   311792       79
               SYS302   396767    21  1468785    224   165312       72
               SYS303   345098    18  1520454    234   254689      264
               SYS304   398781    21  1466771    135   333762      159
               SYS305   402200    22  1463352     98   354982       81
               SYS306   571072    31  1294480     65   513361       49
               SYS307   385102    21  1480450    104   321114       56
               SYS308   385805    21  1479747    237   156927       83
               SYS309   367068    20  1498484     88   326173       60

USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

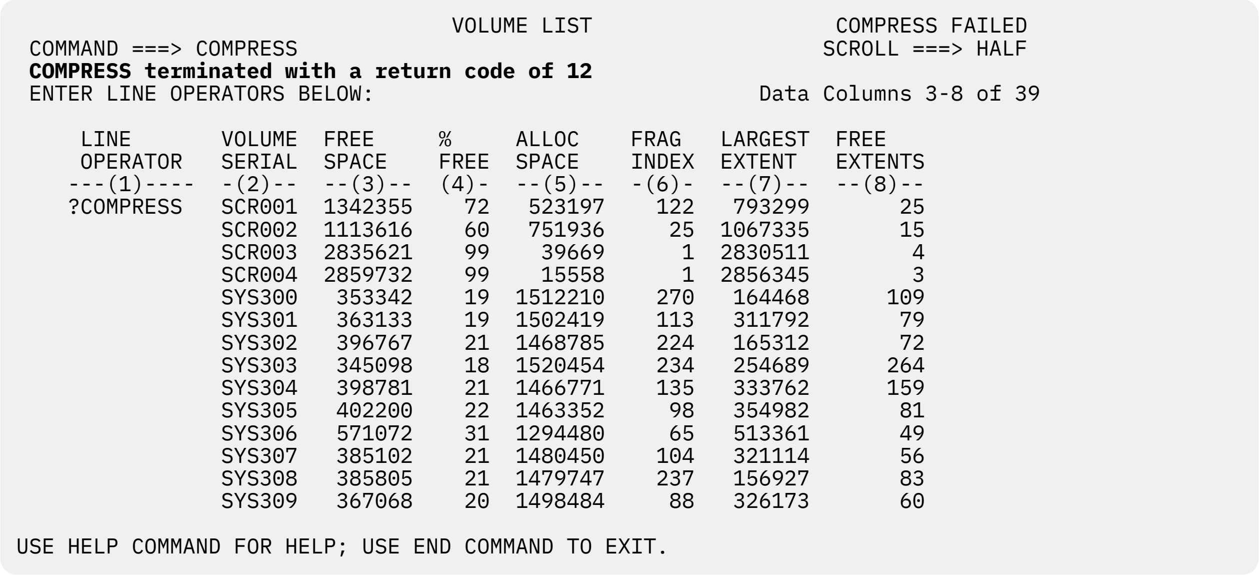
Figure 9. ISMF—End User Short Error Message

Identifying an ISMF error message

1. To determine whether ISMF or ISPF issued the error message, press the HELP key or enter the HELP command. The system displays a long message related to the original short error message. Figure 10 on page 21 is an example of an end-user ISMF panel with the long error message:

   ```
   COMPRESS terminated with a return code of 12
   ```

   ISPF long messages have the same format.



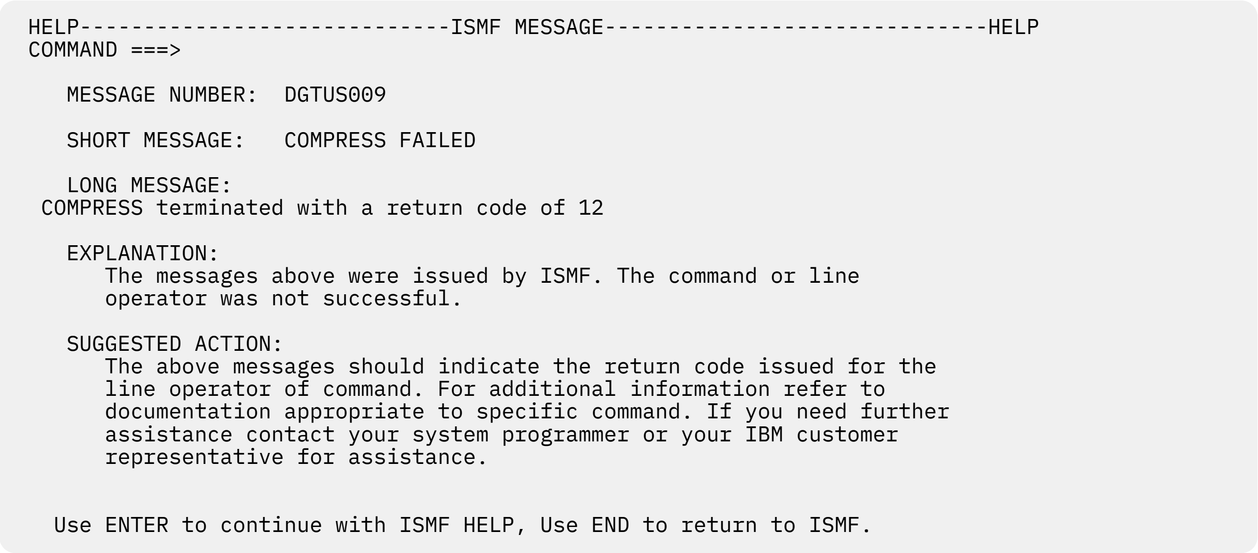
```
                                 VOLUME LIST                  COMPRESS FAILED
COMMAND ===> COMPRESS                                         SCROLL ===> HALF
COMPRESS terminated with a return code of 12
ENTER LINE OPERATORS BELOW:                              Data Columns 3-8 of 39

    LINE       VOLUME  FREE     %     ALLOC    FRAG   LARGEST  FREE
    OPERATOR   SERIAL  SPACE    FREE  SPACE    INDEX  EXTENT   EXTENTS
   ---(1)----  -(2)--  --(3)--  (4)-  --(5)--  -(6)-  --(7)--  --(8)--
   ?COMPRESS   SCR001  1342355    72   523197    122   793299       25
               SCR002  1113616    60   751936     25  1067335       15
               SCR003  2835621    99    39669      1  2830511        4
               SCR004  2859732    99    15558      1  2856345        3
               SYS300   353342    19  1512210    270   164468      109
               SYS301   363133    19  1502419    113   311792       79
               SYS302   396767    21  1468785    224   165312       72
               SYS303   345098    18  1520454    234   254689      264
               SYS304   398781    21  1466771    135   333762      159
               SYS305   402200    22  1463352     98   354982       81
               SYS306   571072    31  1294480     65   513361       49
               SYS307   385102    21  1480450    104   321114       56
               SYS308   385805    21  1479747    237   156927       83
               SYS309   367068    20  1498484     88   326173       60

USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 10. ISMF—End User Long Error Message

2. Press the HELP key again to see the help panel for the short and long messages. The help panel further explains the problem. Figure 11 on page 22 is an example of an ISMF message help panel.



```
HELP-----------------------------ISMF MESSAGE------------------------------HELP
COMMAND ===>

   MESSAGE NUMBER:  DGTUS009

   SHORT MESSAGE:   COMPRESS FAILED

   LONG MESSAGE:
 COMPRESS terminated with a return code of 12

   EXPLANATION:
      The messages above were issued by ISMF. The command or line
      operator was not successful.

   SUGGESTED ACTION:
      The above messages should indicate the return code issued for the
      line operator of command. For additional information refer to
      documentation appropriate to specific command. If you need further
      assistance contact your system programmer or your IBM customer
      representative for assistance.


  Use ENTER to continue with ISMF HELP, Use END to return to ISMF.
```

Figure 11. ISMF—Message HELP Panel

ISMF message identifiers begin with the prefix DGT. ISPF message identifiers begin with ISP or ISR. On the help panel, the message identifier is in the MESSAGE NUMBER field, located at the upper left.

3. If ISMF issued the message, follow the procedure in "Messages issued by ISMF" on page 23 to build the message type-of-failure keyword.
4. If ISPF issued the message, read the help panel to determine whether there is something you can do to resolve the problem.
5. If the help panel does not indicate how to correct the error, see *z/OS ISPF Messages and Codes*.

Messages issued by other subcomponents

1. Record on the "Keyword worksheet" on page 4 the message that interrupted the ISMF operation screen. For an example, see Figure 8 on page 20.
2. Press the ENTER key to get the ISMF short error message. For an example, see Figure 9 on page 21.
3. Enter HELP to get the ISMF long error message. For an example, see Figure 10 on page 21.
4. Enter HELP again to get the help panel that defines the problem further. For an example, see Figure 11 on page 22.

   Read the help panel to determine if you can do something to resolve the problem. If the explanation provided does not indicatehow to correct the error, continue with this procedure.
5. Examine the ISPF log for the message identifier. (See "ISMF's use of the ISPF log" on page 226 for an example of the ISPF log.) The message identifier appears with the original error message. A list of possible three-letter prefixes for messages issued by some other product appears in Table 3 on page 22.
6. If the message starts with any of these prefixes, use the information indicated in Table 3 on page 22 to determine the recommended action. For messages that start with other prefixes, see *z/OS MVS System Messages, Vol 1 (ABA-AOM)*, through *z/OS MVS System Messages, Vol 10 (IXC-IZP)*, to determine the recommended action.

Table 3. ISMF—Message Prefixes for Other Products

Prefix	Subcomponent Name	Information
ADR	DFSMSdss	*z/OS MVS System Messages, Vol 1 (ABA-AOM)*
ARC	DFSMShsm	*z/OS MVS System Messages, Vol 2 (ARC-ASA)*

Table 3. ISMF—Message Prefixes for Other Products (continued)

Prefix	Subcomponent Name	Information
DFQ or DGT	ISMF	(User response options are given on message panel.)
ICK	ICKDSF	*Device Support Facilities (ICKDSF) User's Guide and Reference*
IKJ	TSO/E	*z/OS MVS System Messages, Vol 9 (IGF-IWM)*
ISP or ISR	ISPF	*z/OS ISPF Messages and Codes*

7. If the message has no message identifier, issue the TSO PROFILE command specifying the keyword 'MSGID' and retry the failing function.
8. See "ISMF—module keyword" on page 36.

Messages issued by ISMF

1. Read the help panel to determine if you can do something to resolve the problem. If the explanation provided does not indicate how to correct the error, continue this procedure.
2. Build the message keyword by appending the message identifier to the keyword prefix MSG.

   Example: If the message identifier is DGTMD006, specify the message type-of-failure keyword as follows:

   ```
   MSGDGTMD006
   ```

   If ISPF logging was in effect when the message was issued, specify the return code and reason code related to the message exactly as the system presents them. Continue with Step "3" on page 23.

   If ISPF logging was not in effect when the message was issued, the type-of-failure keyword is complete. See "ISMF—module keyword" on page 36.
3. If the ISPF log was turned on when the message was issued, you must specify the message-related return code and reason code. To determine the return code and reason code, examine the ISPF log for the entry associated with the message identifier. (See "ISMF's use of the ISPF log" on page 226 for a description of the ISPF log.) Record the return code and reason code exactly as they appear in the log, on the "Keyword worksheet" on page 4 as modifier keywords.
4. See "ISMF—modifier keywords" on page 51.

Object Access Method—message keyword

Use this section for OAM-related failures.

Non-OAM components—message keywords

This section is for messages that contain OAM information only. The OAM component does not issue these messages.

Whereas messages issued by OAM have the prefix CBR, software applications that use OAM services issue their own messages while referring to OAM processing information.

The following are valid message keywords:

- External return code keyword prefix: The characters XRC, to which each external OAM return code is appended.
- External reason code keyword prefix: The characters XRSN, to which each external OAM reason code is appended.

Example: Suppose the message is the following one:

```
ECK0703E  OAM ERROR OCCURRED ON OBJECT objectname MODERR:
          modname RETCODE: 12 REACODE: 20
```

```
          XRETCODE: 16 XREACODE: A0020D00
```

Specify the keyword string as follows:

```
MSGEKC0703E RC12 RSN20 XRC16 XRSNA0020D00
```

OAM passes the return, reason, external return, and external reason codes to the application code that uses OAM services. These codes are in the syslog or in an application log. For more information, refer to Chapter 13, “OAM diagnostic aids,” on page 281.

Storage Management Subsystem—message keyword

Use this section for problems related to Storage Management Subsystems.

Procedure

Before using this section, examine *z/OS MVS System Codes* and *z/OS MVS System Messages, Vol 1 (ABA-AOM)*, through *z/OS MVS System Messages, Vol 10 (IXC-IZP)*. These might help you generate additional keywords by identifying failure-related functions and providing message-to-module cross-reference tables.

See “Definition of message keyword related terms” on page 17 for definitions of the following terms used in this section:

- Message identifier
- Message keyword prefix
- Return or reason code
- Return or reason code keyword prefix

Message IGD300I

This message always accompanies an SMS abend. If SMS cannot find the SDWA and recover, message IGD312I might appear. This message usually does not appear.

1. Figure 12 on page 25 shows a sample SMS abend-related job log, which always contains message `IGD300I`.
2. Because this message reports an abend type-of-failure, do not specify the message identifier as a keyword. See “Procedure for SVC dump” on page 8 and use that section to identify the failure-related keywords.

```
*******************************************************
                    J E S 2  J O B  L O G  --  S Y S T E M  3 0 8 1  --  N O D E  N 1
07.39.08 JOB   29  $HASP373 S2RAS031 STARTED - INIT  1 - CLASS A - SYS 3081
07.40.13 JOB   29      S2RAS031    STEP 0      IKJEFT01    0000
07.40.19 JOB   29  DFPWTX30 ISSUING COMMAND.
07.40.20 JOB   29 *41 S2RAS031--REPLY GO STEP1
07.40.46 JOB   29  R 41,U
07.40.52 JOB   29      S2RAS031    STEP1       WTORPGM     0000
07.42.31 JOB   29      S2RAS031    STEP2       AMBLIST     0000
07.42.58 JOB   29      S2RAS031    STEP2       AMBLIST     0000
                   IGD300I AN ABEND OCCURRED DURING SMS PROCESSING
                     ABEND SYSTEM CODE=06F ASID=0010
                     COMPONENT NAME=SMS  COMPONENT ID=28462
                     ACTIVE LOAD MODULE  NAME=IGDZILLA ADDRESS=01BE1000
                     CSECT IN ERROR      DESCRIPTION=BUILD MSG RTN 2
                                         NAME=IGDMCSC2 ADDRESS=01BF5758 OFFSET=00000026
                                         ASSEMBLY DATE=032487   PTF LEVEL=HDP3310
                     PSW AT TIME OF ERROR 071C0000 81BF577E
                     DATA AT PSW 01BF5778 - C5404040 400090EC D00C18CF
                     GPR  0-3  008C7444  7F70FC78  00000010  00000000
                     GPR  4-7  00000018  7F70FD10  7F70F434  7F70FD14
                     GPR  8-11 7F70FDD8  01BF58B2  01BF48B3  7F70FAB0
                     GPR 12-15 81BF38B4  7F70FAB0  81BF41B6  81BF5758
07.43.03 JOB   29  IGD306I UNEXPECTED ERROR DURING IGDMCSCN PROCESSING
                   RETURN CODE IS 8, REASON CODE IS 12008
                   THE MODULE THAT DETECTED THE ERROR IS IGDMCSCN
                   SMS MODULE TRACE BACK - MCSCM DSP00 SSIRT
                   SYMPTOM RECORD CREATED, PROBLEM ID IS IGD00025
07.43.04 JOB   29      S2RAS031    STEP3       IGDRAS00    0000
07.43.04 JOB   29  $HASP395 S2RAS031 ENDED
```

Figure 12. Sample SMS Failure Related Job Log Output

Message IGD306I

This message indicates that an SMS failure has occurred and that SMS has written a symptom record to SYS1.LOGREC.

1. Print SYS1.LOGREC to obtain structured search keyword information. See *EREP User's Guide* for instructions on how to print SYS1.LOGREC. See the sample record in "Storage Management Subsystem records written to logrec data set" on page 522. Use these keywords as the complete keyword string.
2. See "Searching with the keyword string" on page 2.

If the CSECT IN ERROR is not an SMS load module or CSECT, then SMS detected an error from a component or subcomponent that was called to perform a service. The return and reason codes might also belong to the called component or subcomponent.

SMS return and reason codes are in "Storage Management Subsystem return and reason codes" on page 525. Errors returned from the called component or subcomponent should be pursued with that component or subcomponent. Use Table 4 on page 25 to help identify the failing component or subcomponent.

Note: The called components or subcomponents might issue messages preceding the IGD306I message. These initial messages can describe the error.

Table 4. Components or Subcomponents Called by SMS.

CSECT IN ERROR Name or Prefix	Component or Subcomponent	Reference
AOM	Asynchronous Operations Manager (AOM)	Chapter 3, "AOM diagnostic aids," on page 69
CATALOG or IGG0CL	Catalog (SVC 26)	*z/OS MVS System Messages, Vol 8 (IEF-IGD)* (IGD3009I) "Catalog management diagnostic aids" on page 576
CBR	Object Access Method (OAM)	Chapter 13, "OAM diagnostic aids," on page 281

Table 4. Components or Subcomponents Called by SMS. (continued)

CSECT IN ERROR Name or Prefix	Component or Subcomponent	Reference
DEVINFO	Device Information Services	"Device information services return, reason, and response codes" on page 191
DEQUEUE or DEQ	DEQUEUE	*z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN*
DIV	Data in virtual	*z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN*
ENQUEUE or ENQ	ENQUEUE	*z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG*
GETMAIN	GETMAIN	*z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG*
IEFAB, SVC99 or SVC_99	Dynamic Allocation (SVC 99)	*z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN*
IEFSJRTE or SJF	Scheduler JCL Facility (SJF)	*z/OS MVS Programming: Authorized Assembler Services Guide*
IGGDA	Direct Access Device Space Management (DADSM)	Chapter 7, "DADSM/CVAF diagnostic aids," on page 109
IGW	Buffer Manager Facility (BMF) or System Support Facilities (SSF)	Chapter 16, "PDSE diagnostic aids," on page 487
RAC or SAF	Resource Access Control Facility (RACF®, a component of the Security Server for z/OS) or System Authorization Facility (SAF)	*z/OS MVS Programming: Authorized Assembler Services Guide*
SWAREQ	SWAREQ macro	*z/OS MVS Programming: Authorized Assembler Services Reference SET-WTO*

Message IGD311I

This message reports the same condition as IGD306I, except that SMS was unable to write a symptom record to SYS1.LOGREC.

1. Record the module in error, the return code, and the reason code as keywords.

   The failure to write a symptom record to SYS1.LOGREC is usually unrelated to the problem for which the record was being written. Message IGD311I also contains the return and reason codes from the unsuccessful attempt to write the SYS1.LOGREC record. Do not specify these codes in the keyword string.
2. If you want to do a keyword search for the failure to write to SYS1.LOGREC, do not include the failing module name, return code, or reason code associated with the original failure. Specify only the message identifier, return code, and reason code related to the SYS1.LOGREC write failure.
3. If you want to do a keyword search for the original failure, specify the original failure-related module name, return code, and reason code.
4. See "Modifier keywords" on page 42.

VSAM, transactional VSAM and VSAM RLS record management—message keyword

VSAM record management does not issue any messages directly. However, the results of a record management request can be translated into a message that the user of record management issues. Use this section when your program or the system indicates that a VSAM data set is being processed.

Before using this section, examine *z/OS MVS System Codes* and *z/OS MVS System Messages, Vol 1 (ABA-AOM)*, through *z/OS MVS System Messages, Vol 10 (IXC-IZP)*. These might help you generate additional keywords because they identify failure-related functions and provide message-to-module cross-reference tables.

See "Definition of message keyword related terms" on page 17 for definitions of the following terms used in this section:

- Message identifier
- Message keyword prefix
- Return or reason code
- Return or reason code keyword prefix

Procedure

1. A damaged data set can cause one of the following messages to be issued by the caller of VSAM record management or by a system service routine (for example, EOV or IOS) that was invoked by record management:
   - MSGIDC3302I—Action error
   - MSGIDC3308I—Duplicate records
   - MSGIDC3314I—Out-of-sequence records, missing records, duplicate records, no record found
   - MSGIDC3351I—VSAM logic I/O error RC156, RC24, or RC32
   - MSGIDC3350I—No record found or incorrect length
   - MSGIEC070I—RC32, RC202, RC104, or RC203
   - MSGIOS000I—Command reject
2. If the system issues one of these messages while processing a key-sequenced data set (KSDS), determine whether you have a damaged data set. Issue the IDCAMS EXAMINE command as described in the topic on functional command format in *z/OS DFSMS Access Method Services Commands* and the topic on checking a VSAM key-sequenced data set cluster for structural errors in *z/OS DFSMS Using Data Sets*. The EXAMINE command provides details about the nature of data set damage.

   Example: If a damaged data set caused message IDC3302I to be issued, specify the message type-of-failure keyword as follows:

   ```
   MSGIDC3302I
   ```
3. See "Modifier keywords" on page 42.

Incorrect output keyword

Use this section when a program or the system does not produce the expected output.

Symptoms of the failure

Incorrect output failures can be identified by the following:

- Expected output is missing.
- Output is different than expected.
- Output should not have been generated.
- System indicates damage to the VTOC or VTOC index.

- ISMF panel information or flow is erroneous.

Incorrect output can be the result of a previous failure and can often be difficult to analyze because the component affected might not be the one that caused the problem. Review previous messages, abends, console logs, or other system responses. They could indicate the source of the failure.

Procedure

1. If a message accompanied the failure, append the message identifier to the prefix MSG according to the procedures starting on "Message keyword" on page 17 and add this keyword to the keyword string. If the system did not issue a message, try to identify any failure-related control blocks, user areas, or data records and record them on the "Keyword worksheet" on page 4 as modifier keywords.

   Specify the incorrect output keyword as INCORROUT.
2. If the system indicates damage to the VTOC or VTOC index, then DADSM or CVAF normally issues an error message. In this case, examine the "Standard modifier keyword list" on page 43 and go to the DADSM/CVAF-related "Incorrect output failure modifier keywords procedure" on page 48 to identify applicable symptom keywords. If VTOC problems are not indicated, continue with this procedure.
3. Accumulate as much of the following information as possible. It can help you isolate or resolve your problem, and the IBM Support Center will request it if trap or trace information is needed.
   - When was the problem first noticed?
   - How was the problem identified (good output versus bad output)?
   - Were any system changes or maintenance recently applied? For example, a new device, software product, APAR, or PTF?
   - Does the problem occur with a specific data set, device, time of day, and so forth?
   - Does the problem occur in batch or in TSO mode?
   - Is the problem solid or intermittent?
   - Can the problem be re-created?
4. Select the procedure for the failure-related component from the following table.

Subcomponent	Procedure
Catalog Management	See "Catalog Management—incorrect output keyword" on page 32
Device Console Services	See "Device console services—Incorrect output keyword" on page 28
ISMF	See "ISMF—incorrect output keyword" on page 29
Media Manager	See "Media Manager—incorrect output keyword" on page 29
O/C/EOV (Common)	See "OPEN/CLOSE/End-of-Volume (common)—modifier keywords" on page 55
Storage Management Subsystem (SMS)	See "Storage Management Subsystem—incorrect output keyword" on page 30
Record Management	See "VSAM record management—incorrect output keyword" on page 30
All Other DFSMSdfp Subcomponents	See "Other DFSMSdfp subcomponents—modifier keywords" on page 60

Device console services—Incorrect output keyword

Use this section when your program or the system produces other than the expected output and you suspect a failure in device console services.

Procedure

1. If the DEVSERV command causes the issuance of messages that contain incorrect information, specify the incorrect output type-of-failure keyword as INCORROUT.
2. Record the DEVSERV command-line parameters as modifier keywords on the "Keyword worksheet" on page 4.
3. The message text might contain additional information that you can use as modifier keywords (function, subfunction, device-related information, and so forth); record it on the "Keyword worksheet" on page 4.
4. See "Other DFSMSdfp subcomponents—module keyword" on page 41.

ISMF—incorrect output keyword

Use this section to determine the required action if an incorrect output condition occurs with an ISMF panel.

Procedure

Before reporting an ISMF panel problem, ensure that the incorrect output is not the result of a customized panel or message. If the panel is modified, retry the operation with the IBM-supplied copy. If the failure still occurs, specify the incorrect output type-of-failure keyword as INCORROUT. See "ISMF—modifier keywords" on page 51.

Media Manager—incorrect output keyword

Use this section when the system produces other than the expected output and you suspect a failure in the media manager.

The media manager processes read and write requests from the following subcomponents for the types of records indicated.

Subcomponent	Record Type
Catalog	VVDS records contained in the SYS1.VVDS data set
CVAF	Indexed VTOC records contained in the SYS1.VTOCIX data set
DB2®	Db2 records contained in the Db2 table/index spaces, and the Db2 log data set
HFS	Hierarchical file system
IMS Fast Path	IMS records contained in the IMS data entry database.
PDSE	Partitioned data set extended
Sequential extended format data sets	Sequential data set
VSAM	VSAM data

Incorrect output could be the result of a previous failure.

Procedure

1. If the media manager provides output that is missing or different than expected, the subcomponent that requested services from the media manager issues an explanatory message. If this occurs, specify the message keyword as the type-of-failure keyword instead of INCORROUT, and continue below. If no such message is issued, specify INCORROUT as the type-of-failure keyword and see "Module keyword" on page 33.
2. Try to identify any significant failure-related keywords that appear in the message prefix, number, or text. Record them as modifier keywords on the "Keyword worksheet" on page 4 for use in the software support database search.

3. See “Module keyword” on page 33.

Storage Management Subsystem—incorrect output keyword

Use this section when the output of an SMS command or SMS-related function is incorrect. SMS-related processing includes:

- SMS operator commands
- ACS processing

Procedure

1. If the incorrect output includes a message, see “Message keyword” on page 17 and describe the problem as a message type-of-failure, but include the INCORROUT keyword in the keyword string.
2. Identify the failure-related SMS function, for example, which SMS command was issued, which SMS class was assigned, and so forth.
3. See “Storage Management Subsystem—modifier keywords” on page 56.

VSAM record management—incorrect output keyword

Use this section to gather detailed information about an incorrect output type-of-failure related to VSAM record management.

Incorrect output could have been caused by a previous failure. Examine the system and console logs for failure-related abends, messages, or return codes. A damaged VSAM data set can also cause incorrect output. Add any failure-related return codes to the keyword string, exactly as the system presents them. You can also add the abend or message type-of-failure keywords to the incorrect output keyword string to define the symptoms more closely.

Procedure

1. Determine whether failure-related record management return codes and reason codes exist.

   VSAM provides return codes in register 15 and reason codes in either the access method control block (ACB) or the request parameter list (RPL). Reason codes in the ACB indicate VSAM open or close errors. Reason codes in the RPL indicate VSAM record management error indications returned to the caller of record management. Reason codes returned to the caller of record management in the RPL indicate VSAM record management errors.

2. Record any failure-related RPL feedback word (a hexadecimal full word) and RPL return code on the “Keyword worksheet” on page 4 as modifier keywords. The IBM Support Center can use these values to identify a failure-related module and the nature of the incorrect output.

   Example: If the RPL feedback word is X'000C0010', specify the following keywords:

   ```
   RPLFDBWD 000C0010
   ```

3. Determine whether you have a damaged VSAM data set.

   Some incorrect output failures involve a damaged VSAM data set. To determine whether you have a damaged data set, use the IDCAMS EXAMINE command as described in the topic on functional command format in *z/OS DFSMS Access Method Services Commands* and the topic on checking a VSAM key-sequenced data set cluster for structural errors in *z/OS DFSMS Using Data Sets*. The EXAMINE command provides details about the nature of data set damage.

   If these service aids indicate that the data set is not damaged, inform the IBM Support Center if you call for assistance. If they indicate that the data set is damaged, keep a copy of the output for possible use by the IBM Support Center. Be prepared to describe the type of data set damage. You should attempt to recover the data set and rerun the failing job to determine whether the problem is resolved.

   The system can indicate a damaged data set by one of the following:

   - Messages (discussed in Message section)

- ABEND0C4 (discussed in ABEND section)
- Wait/Loop (discussed in Wait/Loop section)
- RPL feedback word: *nn*X'08'*nn*X'9C' or *nn*X'08'*nn*X'20'

4. If the data set is damaged, rebuild it as directed in “VSAM record management (R/M) diagnostic aids” on page 593, and rerun the job.
5. See “Modifier keywords” on page 42.

Transactional VSAM—incorrect output keyword

Use this section to gather detailed information about an incorrect output type-of-failure related to transactional VSAM.

Incorrect output could have been caused by a previous failure. Examine the system and console logs for failure-related abends, messages, or return codes. A damaged VSAM data set can also cause incorrect output. Add any failure-related return codes to the keyword string, exactly as the system presents them. You can also add the abend or message type-of-failure keywords to the incorrect output keyword string to define the symptoms more closely.

Procedure

1. Determine whether failure-related transactional VSAM return codes and reason codes exist.

   Look for transactional VSAM RLS return codes in register 15. Look for reason codes in either the access method control block (ACB) or the request parameter list (RPL). Reason codes in the ACB indicate VSAM open or close errors. Reason codes in the RPL indicate record management error indications returned to the caller of RLS.
2. Record any failure-related RPL feedback word (a hexadecimal full word) and RPL return code on the “Keyword worksheet” on page 4 as modifier keywords. The IBM Support Center can use these values to identify a failure-related module and the nature of the incorrect output.

   Example: If the RPL feedback word is X'000C0010', specify the following keywords:

   ```
   RPLFDBWD 000C0010
   ```
3. Determine whether you have a damaged VSAM data set.

   Some incorrect output failures involve a damaged VSAM data set. To determine whether you have a damaged data set, use the IDCAMS EXAMINE command as described in the topic on functional command format in *z/OS DFSMS Access Method Services Commands* and the topic on checking a VSAM key-sequenced data set cluster for structural errors in *z/OS DFSMS Using Data Sets*. The EXAMINE command provides details about the nature of data set damage.

   If these service aids indicate that the data set is not damaged, inform the IBM Support Center if you call for assistance. If they indicate that the data set is damaged, keep a copy of the output for possible use by the IBM Support Center. Be prepared to describe the type of data set damage. You should attempt to recover the data set and rerun the failing job to determine whether the problem is resolved.

   The system can indicate a damaged data set by one of the following:

   - Messages (discussed in Message section)
   - ABEND0C4 (discussed in ABEND section)
   - Wait/Loop (discussed in Wait/Loop section)
   - RPL feedback word: *nn*X'08'*nn*X'9C' or *nn*X'08'*nn*X'20'.
4. If the data set is damaged, rebuild it as directed in “VSAM record management (R/M) diagnostic aids” on page 593, and rerun the job.
5. See “Modifier keywords” on page 42.

Catalog Management—incorrect output keyword

Use this section to define the keyword when the system produces other than the expected output and you suspect a failure in the catalog management area.

Procedure

Determine the extent of the incorrect output.

1. Use the LISTCAT command as described in the topic on functional command format in *z/OS DFSMS Access Method Services Commands* to obtain a complete listing of the catalog.
2. Use the IEHLIST program as described in *z/OS DFSMSdfp Utilities* to obtain a listing of the VTOC. This might be useful when diagnosing problems in managing DASD volume space or in using access method services commands.
3. Use the DIAGNOSE command as described in the topic on functional command format in *z/OS DFSMS Access Method Services Commands* to determine whether a catalog structure is correct. Include any reason codes produced by DIAGNOSE in your search argument.

   Example: If the reason code is 23, specify it as follows:

   ```
   DIAGNOSE   RC23
   ```

4. Use the IDCAMS EXAMINE command as described in the topic on functional command format in *z/OS DFSMS Access Method Services Commands* and the topic on checking a VSAM key-sequenced data set cluster for structural errors in *z/OS DFSMS Using Data Sets* to determine whether the catalog being used has been damaged and the nature of any damage.

   If the output of these service aids (LISTCAT, IEHLIST, DIAGNOSE, or EXAMINE) indicates that the catalog is not damaged, inform the IBM Support Center if you call for assistance. If they indicate that the catalog is damaged, keep a copy of the output for possible use by the IBM Support Center. Be prepared to describe the type of catalog damage. You should attempt to recover the catalog and rerun the failing job to determine whether the problem is resolved.
5. See “Catalog management—modifier keywords” on page 59.

Performance keyword

Use this section to define the performance keyword when you suspect that a DFSMSdfp component or subcomponent is causing poor system performance.

Performance failure definition

Performance is defined as a measurement of either work or time:

- System-related work accomplished within a given time
- The time required to complete a task or job, based upon past performance.

Many performance problems are related to system tuning.

Modifications to your system or its environment can influence system performance. Consult with IBM for system tuning assistance.

Procedure

1. Use the performance keyword only when tuning cannot improve system performance and performance is below explicitly stated expectations.
2. Record actual performance, expected performance, and the source of the expected performance criteria. Include the order number and page of the source document. If you contact the IBM Support Center, you will be asked for this information.
3. Specify the performance keyword as PERFM.

4. If the problem is related to a function or module, specify the function or module name as a modifier keyword.

   Example: If the module is IFG0194A and the function is OPEN, specify the keywords as follows:

   ```
   IFG0194A OPEN
   ```

5. If possible, isolate the probable source of degraded performance to a specific DFSMSdfp subcomponent.
6. See Table 19 on page 60.

Documentation keyword

Use this section when you encounter incorrect or missing information in DFSMSdfp documentation.

Procedure

For a minor error, submit a Reader's Comment Form for the documentation in error. If the error is serious and of general concern to other users, continue with the procedure below.

Restriction: IBM does not accept APARs for minor documentation errors (spelling, punctuation, or grammar).

1. Record the document page in error. Be prepared to describe the error and the problem it caused.
2. Specify the order number of the document after the prefix DOC, omitting the hyphen and level number.

   Example: If the order number is GY27-7618-06, specify the type-of-failure keyword as follows:

   ```
   DOC GY277618
   ```

3. For documentation problems related to a system message, append the message identifier to the keyword prefix MSG, add it to the keyword string as a modifier keyword.

   Example: If the message is IEC147I, specify the modifier keyword as follows:

   ```
   MSGIEC147I
   ```

4. For problems related to a function or module, specify the function or module name as a modifier keyword.

   Example: If the module is IFG0194A and the function is OPEN, specify the modifier keywords as follows:

   ```
   IFG0194A OPEN
   ```

5. See "Searching with the keyword string" on page 2.

Module keyword

Use this section to build the module keyword for your keyword string. This keyword identifies the failure-related Control Section (CSECT). A load module consists of one or more individually replaceable CSECTs.

The CSECT name describes the area of suspected failure more precisely than the load module name, but you can specify either or both when varying the keyword string during the software support database search.

Procedure

- If the system has given some indication of a failure-related module in SVCDUMP TITLE, LOGREC RECORD, DUMP SUMMARY PAGE, or MESSAGES output, specify the complete module name as the module keyword and see "Modifier keywords" on page 42.

- You should be able to infer the probable failing component from a failure-related indication (such as a message or unique abend code).
- If you have a dump that contains the nucleus and LPA, select the procedure for the failure-related component indicated in Table 5 on page 34.
- For some failures, you might find it impossible to identify a probable failing module. In this case, select the procedure for the failure-related component from Table 5 on page 34.

Table 5. Building the Module Keyword

Subcomponent	Procedure
Catalog Management	See “Catalog management—module keyword” on page 40.
DADSM/CVAF	See “DADSM/CVAF—module keyword” on page 34.
DASD Common Services	See “DASD common services—module keyword” on page 35.
ISMF	See “ISMF—module keyword” on page 36.
Media Manager	See “Media Manager—module keyword” on page 38.
OPEN/CLOSE/End-of-Volume	See “OPEN/CLOSE/End-of-Volume (common)—module keyword” on page 39.
SAM	See “SAM—module keyword” on page 40.
All other DFSMSdfp subcomponents	See “Other DFSMSdfp subcomponents—module keyword” on page 41.

DADSM/CVAF—module keyword

Use this section to build the module keyword for either the DADSM or CVAF subcomponents.

Procedure

Use either of the following procedures.

Abend or wait/loop type-of-failure

1. For ABEND types-of-failure, the system might have issued message IEC999I, identifying the module in which the error occurred. If this occurs, go to Step “5” on page 35 to determine if it is a DADSM module.
2. Using the PSW from the formatted part of the dump obtained earlier, or from the symptom dump (summary) information in the system job log, extract the data in the last four bytes of the PSW. These bytes contain the address of the instruction that follows the last instruction executed before the abend.
3. Using a map of the LPA and the instruction address, identify the name of the module that was in control when the abend occurred.
4. If the module name begins with the letters ICV, CVAF is the failing component. Locate the CVAF module name in Table 36 on page 109 and record the related function and subfunction on the “Keyword worksheet” on page 4 as modifier keywords.

   Example: If the module name is ICVCMB00, specify the keyword as follows:

   ```
   ICVCMB00
   ```

   - For ABEND types-of-failure, see “DADSM/CVAF—modifier keywords” on page 46. For WAIT or LOOP types-of-failure, see “Release level keyword” on page 63.

5. Table 37 on page 126 contains a complete list of DADSM module names and their related functions and subfunctions. If the module name you identified in Step "1" on page 34 or Step "3" on page 34 is in the list, the subcomponent is DADSM. Record the function and subfunction on the "Keyword worksheet" on page 4 as modifier keywords.

   Example: If the module name is IGC0002G, specify the keyword as follows:

   ```
   IGC0002G
   ```

   - For ABEND types-of-failure, see "DADSM/CVAF—modifier keywords" on page 46. For WAIT or LOOP types-of-failure, see "Release level keyword" on page 63.

6. Use the Component Identification Keyword 5695DF133 to develop a keyword string for the DADSM/CVAF subcomponents. Include the suspected module name in the keyword string. The problem might have been previously reported with the same module name. Code the string as follows:

   ```
   5695DF133 ABEND0C4 ICVCMB00
   ```

   If the search keyword string does not yield usable results, and if the error might exist outside this component, you can broaden the search by deleting the component identification keyword. Build a keyword string containing only the abend code and the module name as:

   ```
   ABEND0C4 ICVCMB00
   ```

   This keyword string might yield a previously reported problem (and its fix) in some subcomponent other than DADSM/CVAF.

   - For ABEND types-of-failure, see "DADSM/CVAF—modifier keywords" on page 46. For WAIT or LOOP types-of-failure, see "Release level keyword" on page 63.

Message type-of-failure

1. Specify the module keyword by using the DADSM or CVAF module name identified at "DADSM/CVAF—message keyword" on page 18.

   Example: If the module name is IGG020P2, specify the module keyword as follows:

   ```
   IGG020P2
   ```

2. See "DADSM/CVAF—modifier keywords" on page 46.

DASD common services—module keyword

Use this section to build the module keyword for the DASD common services subcomponent. This keyword identifies the module related to a failure which results in the unsuccessful execution of one of the services.

Procedure

You can determine the name of the module that detected the error by using the problem determination area (PDA), except for common filter services, which neither employ an ESTAE nor use a PDA.

The failure-related ISPF log entry's FEEDBACK field contains the PDA information. See Figure 28 on page 227 for a sample entry containing PDA information. See Chapter 8, "DASD common services diagnostic aids," on page 187 for a description of the PDA contents.

1. Use the module name identified in the PDA as the module keyword.
2. Record any other significant keywords on the "Keyword worksheet" on page 4 with prefixes applicable to their nature. (For example, specify the return code as RC*nn*.)

Building the module keyword

1. **Example**: If the module name is IGBDIS01, specify the module keyword as `IGBDIS01`.

2. See “DASD common services—modifier keywords” on page 50.

ISMF—module keyword

For an abend type-of-failure, continue with the following procedure.

For all other types-of-failure, see “ISMF—modifier keywords” on page 51.

Procedure

You can determine the name of the failing module for an abend in several ways, depending on how the system indicated the failure and the conditions under which you invoked ISPF and ISMF. Table 6 on page 36 summarizes abend processing. To determine the module name, go to the section indicated under the "Type of Dump" column.

Table 6. Summary of ABEND Processing

ISPF Mode	ISMF Profile	Abend Indicator	Type of Dump
Normal	RECOVER FROM ABENDS ===> Y	ISMF abend panel	ISMF symptom dump, see “Using the ISMF symptom dump” on page 36
Normal	RECOVER FROM ABENDS ===> N	ISPF abend panel	ISPF symptom dump, see “Using the ISPF symptom dump” on page 37
Test/Trace	RECOVER FROM ABENDS ===> Y	ISMF abend panel	ISMF symptom dump, see “Using the ISMF symptom dump” on page 36
Test/Trace	RECOVER FROM ABENDS ===> N	TSO/E error message	System abend dump, see “Using the abend dump” on page 37

See “Determining how ISMF processes abends and generates dumps” on page 221 for information on controlling logging and dumps.

Using the ISMF symptom dump

If an abend occurs while executing ISMF, and logging is active, the system places an ISMF symptom dump (mini dump) in the ISPF log. If ISMF is able to capture the name of the failing module, it appears in the symptom dump. The format of the dump is shown in Figure 13 on page 36.

```
14:32   *****  ISMF ERROR  ***** - APPLICATION(DGTA - LIBRARY); FUNCTION(DEFINE)
14:32                            - WORK.SCDS
14:32                            - ISMF 1.1.0   FMID JDZ1110   SYSTEM ABEND CODE 0C1
14:32                            - XXXXXX NAME DGTFLCDA   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 13. ISMF—Symptom Dump in the ISPF Log

The name field in the symptom dump identifies the name of the failing module if it is available.

1. If you can determine the module name from the symptom dump:

    a. Use the module name as the module keyword.

Example: If the module name is DGTFLCDA, specify the module keyword as shown:

```
DGTFLCDA
```

b. See “ISMF—modifier keywords” on page 51.

2. If you cannot determine the module name from the symptom dump:

   a. Continue with the procedures for determining the module name described in “Using the abend dump” on page 37.

   b. If you still cannot determine the module name, see “ISMF—modifier keywords” on page 51.

Using the ISPF symptom dump

If the ISMF profile variable indicates that ISMF is not to recover from abends, and ISPF is executing in normal mode, the system displays an ISPF abend panel and places an ISPF symptom dump in the ISPF log. See “ISMF's use of the ISPF log” on page 226 for a description of the ISPF log. The format of the ISPF symptom dump is shown in Figure 14 on page 37.

```
14:45      ISP SUBTASK ABEND     - VS 03.8 ISP 3.3.0000  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 14. ISPF Symptom Dump in the ISPF Log

The name field in the symptom dump identifies the name of the failing module if it is available.

1. If you can determine the module name from the symptom dump:

   a. Use the module name as the keyword.

      Example: If the module name is DGTFLCDA, (this is not the case in the sample ISPF symptom dump shown in Figure 14 on page 37), specify the module keyword as:

      ```
      DGTFLCDA
      ```

   b. See “ISMF—modifier keywords” on page 51.

2. If you cannot determine the module name from the symptom dump:

   a. Continue with the procedures for determining the module name described in “Using the abend dump” on page 37. If you still cannot determine the module name, see “ISMF—modifier keywords” on page 51.

Using the abend dump

When the symptom of the failure is a TSO/E error message, use the save area trace section of the SYSUDUMP to determine the name of the failure-related module. See Figure 15 on page 38 for a sample save area trace.

```
SAVE AREA TRACE

DGTFMD01 WAS ENTERED VIA LINK         AT EP DGTFMD01..90.349
SA   0002EFD8  WD1 000000D0   HSA 0002E818   LSA 05301C50   RET 80FD2C38   EPA 8538C630   R0  03178AEC
               R1  0002F084   R2  0002EB84   R3  FFFFFFFF   R4  0002EB84   R5  0002A4D0   R6  00000000
               R7  00000001   R8  0002EB80   R9  00029740   R10 00000000   R11 00000000   R12 83178790
DGTFMD01 WAS ENTERED VIA CALL         AT EP DGTFMD05..90.349
SA   05301C50  WD1 00000000   HSA 0002EFD8   LSA 053018E0   RET 8538C832   EPA 8538DA38   R0  03178AEC
               R1  05301D70   R2  0000000C   R3  00000048   R4  05301F54   R5  05301E74   R6  00000000
               R7  00000000   R8  8002EB9A   R9  00029 740  R10 0539213C   R11 05301C50   R12 8538C630
UNKNOWN  WAS ENTERED VIA CALL         AT EP ISPDIR.912 17
SA   00030AC0  WD1 000004C0   HSA 00030300   LSA 0003C010   RET 831708A6   EPA 8316ABC8   R0  00000000
               R1  00030B10   R2  00030E80   R3  000301B8   R4  000000FE   R5  00017C14   R6  00015218
               R7  00030668   R8  00030D60   R9  00029740   R10 00030DD8   R11 00000000   R12 83170018
UNKNOWN  WAS ENTERED VIA CALL         AT EP ISPDIL.92014.OY51175.3.3
SA   0003C010  WD1 000007C0   HSA 00030AC0   LSA 0003C7D0   RET 8316B89A   EPA 83178790   R0  05300FCC
               R1  0003C0E0   R2  00000000   R3  00000000   R4  00017000   R5  00017000   R6  00015218
               R7  0003C378   R8  0316DBC8   R9  00029740   R10 0316CBC8   R11 0316BBC8   R12 8316ABC8
DGTFVA00 WAS ENTERED VIA LINK
SA   0003C7D0  WD1 000000D0   HSA 0003C010   LSA 05301638   RET 80FD2C38   EPA 853AA7A0   R0  03178AEC
               R1  0003C87C   R2  0003C37C   R3  FFFFFFFF   R4  0003C37C   R5  0002A4D0   R6  00000000
               R7  00000001   R8  0003C378   R9  00029740   R10 00000000   R11 00000000   R12 83178790
DGTFVA00 WAS ENTERED VIA CALL         AT EP DGTFVA11..91.221
SA   053966F4  WD1 00000000   HSA 05301638   LSA 053B90BC   RET 853AB422   EPA 8539FCB0   R0  00000000
               R1  05396A20   R2  00000001   R3  00000000   R4  0003C37C   R5  0002A4D0   R6  00000000
               R7  00000001   R8  05396DB8   R9  053AC03D   R10 0539213C   R11 053966F4   R12 853AB03E
DGTFMD01 WAS ENTERED VIA CALL         AT EP DGTFCTPR..91.227
SA   053E80A4  WD1 00000000   HSA 053EA4E4   LSA 053E305C   RET 853FFB5A   EPA 0533F540   R0  053E83F4
               R1  053E827C   R2  053E8404   R3  0005BCD9   R4  053F3E80   R5  053EA874   R6  0005B832
               R7  0005A833   R8  053010CC   R9  053F3145   R10 0539213C   R11 053E80A4   R12 853FF476
```

Figure 15. ISMF—Finding the Module Name in a SYSUDUMP

1. Examine the SAVE AREA TRACE section of the SYSUDUMP.
2. Scan the left-hand margin, looking for the words INTERRUPT AT.
3. Look at each entry in the SAVE AREA TRACE section of the dump (scanning upward from the interrupt address) and find the EPA (entry point address) that is less than (but closest to) the interrupt address.
4. The name of the failing module is located in the row above its EPA (the one that you found in Step "3" on page 38) in the center of the page. The module name should appear in one of the following two sets of words:

   ```
   WAS ENTERED VIA CALL    AT EP module name
   ```

   ```
   WAS ENTERED VIA LINK    AT EP module name
   ```

   In Figure 15 on page 38, the module name for the failure is highlighted.
5. Specify the module name as the module keyword.

   Example: If the module name is DGTFTCMD, specify the module keyword as:

   ```
   DGTFTCMD
   ```
6. See "ISMF—modifier keywords" on page 51.

Media Manager—module keyword

For abend, wait, or loop types-of-failure in the media manager, use the following procedure to build the module keyword.

Procedure

1. Locate in the dump obtained earlier the media manager process block (MMPB). The MMPB is near the beginning of the summary dump, and the acronym MMPB in EBCDIC representation is at offset 0 of the MMPB. See the sample media manager dump in Chapter 12, "Media Manager diagnostic aids," on page 275.

   Use the 1-byte MMPMODID field at offset X'1C' in the MMPB to identify the media manager module in control when the failure occurred. The code values in the field are the same as the values in the *ff* field of the media manager return codes, listed in "Media Manager return codes" on page 277.

Example: If the MMPMODID field value is 05, specify the module keyword as:

```
ICYCCHHR
```

2. See “Modifier keywords” on page 42.

OPEN/CLOSE/End-of-Volume (common)—module keyword

Use this section to build the module keyword for common O/C/EOV.

Procedure

Use one of the following three procedures.

Abend type-of-failure

1. If the failure is indicated by an abend 001, the failure is in the SAM subcomponent. Change the component identification keyword to indicate the SAM subcomponent (see Table 19 on page 60) and see “SAM—module keyword” on page 40 to build the module keyword.
2. If the system issued a message identifying an abend condition, and the module name appears in the message text, specify the module name keyword as shown in the fourth step below. If the name does not appear in the message, do the following:
   a. Using the formatted section of the dump, scan the RBs for the job in question, looking for the one representing the failing user program.
   b. The interrupt code field in the user's RB should indicate an SVC code representing the call to the DFSMSdfp service that abended. The next RB represents the failing DFSMSdfp service. Its interrupt code field (IC portion of the WC-L-IC field) should match the abend code.
   c. Using the address portion of the PSW field in that RB, locate that address in the dump and scan toward either the lower or higher addresses, looking in the translated EBCDIC in the right-hand column for a module name. Common O/C/EOV modules contain the CSECT name in the copyright information at the start of each CSECT and in the XCTL table at the end of each CSECT. Typically, register 6 will point to the name of the current CSECT.

      You can also determine the load module name by matching the PSW instruction address with the addresses in an LPA map (all common O/C/EOV load modules and CSECTs reside in the LPA).
   d. Specify the entire load module or CSECT name as the module keyword.

      Example: If the name is IFG0194C, specify the module keyword as `IFG0194C`.
3. See “OPEN/CLOSE/End-of-Volume (common)—modifier keywords” on page 55.

Wait/Loop type-of-failure

You should have obtained a system dump and determined the instruction address as directed earlier. If the problem is a loop, it might be necessary to obtain either a module-to-module trace, a GTF trace, or a SLIP trace. (See Chapter 14, “OPEN/CLOSE/EOV (common) diagnostic aids,” on page 345 for details.)

1. Once you determine a failure-related module, specify it as the module keyword.

   Example: If the module name is IFG0194C, specify the keyword as:

   ```
   IFG0194C
   ```
2. See “OPEN/CLOSE/End-of-Volume (common)—modifier keywords” on page 55.

Message type-of-failure

1. If a system message contains a failure-related module name or significant modifier keyword information, record these items on the “Keyword worksheet” on page 4.
2. See “OPEN/CLOSE/End-of-Volume (common)—modifier keywords” on page 55.

SAM—module keyword

Use this section to build the module keyword for the SAM subcomponent.

Procedure

If the failure is indicated by an ABEND001, use the following procedure to determine the module name. Otherwise, see Table 19 on page 60.

1. Using the formatted portion of the dump obtained earlier, scan the RBs for the job in question, looking for the one that issued an EOV SVC just prior to the abend. The INTCODE field of the RB contains X'0037' for jobs that issued an EOV SVC. Check the dump and locate the INTCODE field. Refer to *z/OS MVS Diagnosis: Tools and Service Aids* for information on dumps.
2. The instruction address in the RB's PSW is the failure-related address. Determine the module name by matching that address with the addresses in an LPA map. Most SAM modules reside in the LPA. (The exceptions are module IGG019V6 in SYS1.SVCLIB and SAM modules in SYS1.LINKLIB that support IBM 3505/3525 Reader/Punch.)
3. Use the entire module name as the module keyword.

   Example: If the module name is IGG019BB, specify the module name keyword as `IGG019BB`.

   If the system issued message IEC020I to inform the system operator of the abend, the message text identifies the O/C/EOV module that processes SAM abend 001s. Record the module name. If your keyword string does not produce any similar problems in the software support database, replace the SAM module name with the O/C/EOV module name that appears in the IEC020I message and perform the search again.
4. See “Modifier keywords” on page 42.

Catalog management—module keyword

Use this section to build the module keyword for a catalog.

Procedure

1. To determine which module is related to the failure, scan backward in the dump. Using the failing PSW address from either the abend or the wait/loop keyword procedure as a starting point (see the EPSW in the dump or the R/TM work area summary), scan the translated EBCDIC output in the right-hand column for a module name. (In the sample formatted dump portion in Figure 16 on page 40 it is IGG0CLAT.) Specify the entire module name as the module keyword.

```
000192A0 00 00000008 00015246 00070B9E 0001815A  000183CE 0001DE66 0001C13E 0004ED16  *..........................A.....*
000192C0 00 0003B8AA 00010A42 0001F3BE 0001FD36  000201EE 000230CE 0002D316 00038C2E  *..........3...............L.....*
000192E0 00 000458FE 0004A40A 0004A97E 00000000  000475CE 00038056 0001A14E 00057B8E  *................................*
00019300 00 00053822 0005582A E2E8E2F1 48E5E5C4  E24BE5C7 C5D5C4E2 D7404000 00000000  *........SYS1.VVDS.VGENDSP  .....*
00019320 00 47F0F024 C9C7C7F0 C3D3C1E3 F0F861F3  F161F0F5 C8C4D7F3 F3F1F040 D5D6D5C5  *.00.IGG0CLAT08.31.05JDZ1110 NONE*
00019340 00 40404040 C9C7C7D7 C3C4E5D9 07FE90CE  D00C41DD 000C05C0 41A0CFFF 4190AFFF  *    IGGPCDVR....................*
00019360 00 50D0B12C D202B40D A5A64110 B13092FC  B13991A0 10080A04 181F1211 4780C040  *....K...........................*
00019380 00 D203B004 A5B958F0 A65D05EF 41A0CFFF  4190AFFF 47F0C052 9180B314 47E0C04C  *K......0.............0..........*
000193A0 00 45E0CDF6 D2F7B314 B31C9500 B0074780  C06292A4 B00D45E0 CD2E5850 B4105050  *...6K7..........................*
```

Figure 16. Catalog Management Sample SVC Dump—Module Name in EBCDIC Section

2. To obtain a map of the catalog load module, take either of the following actions:

   - Execute the LISTLOAD XREF option of AMBLIST for modules IGG0CLHA and IGG0CLX0.
   - Use the map of modules IGG0CLHA and IGG0CLX0 provided in all CAS SVC dumps. Figure 17 on page 41 shows part of such a map. For each CSECT in load modules IGG0CLHA and IGG0CLX0, the map includes the following information:

     – The starting address of the CSECT

     – The CSECT name

  - The CSECT FMID
  - The CSECT service level.
3. See “Catalog management—modifier keywords” on page 59.

```
CAS Module CSECT MAP&colon.

01AEA000   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
01AEA020   TO NEXT LINE ADDRESS SAME AS ABOVE
01AEA980   C9C7C7D4 D6C4D4D7 00001680 01AEA9A0   01AEABC0 00000000 00000000 00000000  *IGGMODMP........................*
01AEA9A0   C9C7C7F0 C3D3C8C1 00CC4000 00CCEE48   00000010 00000000 00000000 00000000  *IGG0CLHA........................*
01AEA9C0   00CC4000 C9C7C7F0 C3D3C8C1 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *.. .IGG0CLHAJDZ1110 NONE    ....*
01AEA9E0   00CC6BB8 C9C7C7F0 C3D3C8C2 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *....IGG0CLHBJDZ1110 NONE    ....*
00CCAA00   00CC5760 C9C7C7F0 C3D3C8C3 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *....IGG0CLHCJDZ1110 NONE    ....*
01AEAA20   00CC74D8 C9C7C7F0 C3D3C8C4 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *...QIGG0CLHDJDZ1110 NONE    ....*
01AEAA40   00CC7A20 C9C7C7F0 C3D3C8C5 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *....IGG0CLHEJDZ1110 NONE    ....*
01AEAA60   00CC7BF0 C9C7C7F0 C3D3C8C6 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *...0IGG0CLHFJDZ1110 NONE    ....*
01AEAA80   00CC8288 C9C7C7F0 C3D3C8E5 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *....IGG0CLHVJDZ1110 NONE    ....*
01AEAAA0   00CC8428 C9C7C7F0 C3D3C8F0 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *....IGG0CLH0JDZ1110 NONE    ....*
01AEAAC0   00CC8A60 C9C7C7F0 C3D3C8F1 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *....IGG0CLH1JDZ1110 NONE    ....*
01AEAAE0   00CC9590 C9C7C7F0 C3D3C8F2 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *....IGG0CLH2JDZ1110 NONE    ....*
01AEAB00   00CC9E70 C9C7C7F0 C3D3E7C1 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *....IGG0CLXAJDZ1110 NONE    ....*
01AEAB20   00CCB2B8 C9C7C7F0 C3D3E7C2 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *....IGG0CLXBJDZ1110 NONE    ....*
01AEAB40   00CCC898 C9C7C7F0 C3D3E7C3 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *..H.IGG0CLXCJDZ1110 NONE    ....*
01AEAB60   00CCD200 C9C7C7F0 C3D3E7C4 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *..K.IGG0CLXDJDZ1110 NONE    ....*
01AEAB80   00CCE180 C9C7C7F0 C3D3E7C5 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *....IGG0CLXEJDZ1110 NONE    ....*
01AEABA0   00CCE7D8 C9C7C7F0 E3D9D4F1 C8C4D7F3   F3F1F040 61F0F461 40404040 00000000  *..XQIGG0TRM1JDZ1110 NONE    ....*
01AEABC0   C9C7C7F0 C3D3E7F0 000098C8 00081000   00000084 00000000 00000000 00000000  *IGG0CLX0...J....................*
01AEABE0   000098C8 C9C7C7F0 C3D3E7F0 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *...HIGG0CLX0JDZ1110 NONE    ....*
01AEAC00   0000AC30 C9C7C7F0 C3D3C6F0 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *....IGG0CLF0JDZ1110 NONE    ....*
01AEAC20   0000BFE8 C9C7C7F0 C3D3C1C8 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *...YIGG0CLAHJDZ1110 NONE    ....*
01AEAC40   0000D628 C9C7C7F0 C3D3C1F3 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *..O.IGG0CLA3JDZ1110 NONE    ....*
01AEAC60   0000F5C0 C9C7C7F0 C3D3C6F5 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *..5.IGG0CLF5JDZ1110 NONE    ....*
01AEAC80   00010900 C9C7C7F0 C3D3C5C7 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *....IGG0CLEGJDZ1110 NONE    ....*
01AEACA0   00011DB0 C9C7C7F0 C3D3C6F4 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *....IGG0CLF4JDZ1110 NONE    ....*
01AEACC0   00012CD8 C9C7C7F0 C3D3C6F6 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *...QIGG0CLF6JDZ1110 NONE    ....*
01AEACE0   00014768 C9C7C7F0 C3D3C2F6 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *....IGG0CLB6JDZ1110 NONE    ....*
01AEAD00   00015218 C9C7C7F0 C3D3C5C1 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *....IGG0CLEAJDZ1110 NONE    ....*
01AEAD20   00017288 C9C7C7F0 C3D3C6E3 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *....IGG0CLFTJDZ1110 NONE    ....*
01AEAD40   00018990 C9C7C7F0 C3D3C5C8 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *....IGG0CLEHJDZ1110 NONE    ....*
01AEAD60   00019320 C9C7C7F0 C3D3C1E3 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *....IGG0CLATJDZ1110 NONE    ....*
01AEAD80   0001AA00 C9C7C7F0 C3D3C6F1 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *....IGG0CLF1JDZ1110 NONE    ....*
01AEADA0   0001B460 C9C7C7F0 C3D3C6F3 C8C4D7F3   F3F1F040 D5D6D5C5 40404040 00000000  *....IGG0CLF3JDZ1110 NONE    ....*
    .
    .
    .
```

Figure 17. Catalog Management—Sample CAS SVC Dump of CSECT Address Map

Other DFSMSdfp subcomponents—module keyword

Use this section to build the module keyword for DFSMSdfp subcomponents for which no unique procedure exists.

Procedure

Use one of the following procedures:

Abend type-of-failure procedure

1. If the system issued a message identifying an abend condition, the module name might appear in the message text. The CSECT name or module name might also appear in the dump title in a console message or after executing the DISPLAY DUMP,TITLE system command.
2. If an EREP software record exists for the failure, use the failing PSW address with maps of the nucleus and LPA to determine the failing module, and then go to Step “3.d” on page 42. (See *EREP User's Guide* for details.)
3. If neither condition is true, continue as follows:
   a. Using the formatted section of the dump, scan the RBs for the job in question, looking for the one representing the failing user program.
   b. The interrupt code field in the user's RB should indicate an SVC code representing the call to the DFSMSdfp service that abended. The next RB represents the failing DFSMSdfp service. Its interrupt code field (IC portion of the WC-L-IC field) should match the abend code.

   c. Using the address portion of the PSW field in that RB, locate that address in the dump and scan toward the lower addresses, looking in the translated EBCDIC in the right-hand column for a module name. Most DFSMSdfp modules contain the module name at the start of each CSECT. You can also determine the module name by matching that address with the addresses in a map of the nucleus or LPA.
   d. Specify the entire module name as the module keyword.
4. See "Modifier keywords" on page 42.

Wait/Loop type-of-failure procedure

You should have obtained a system dump and determined the instruction address as directed earlier. If the problem is a loop, it might be necessary to obtain a GTF trace.

1. Using the instruction address from the waiting or looping RB's OPSW field and the NUCMAP or LPAMAP (as directed in Step "5" on page 14), determine which module resides at that address, and specify the entire module name as the module keyword.
2. See "Modifier keywords" on page 42.

Message type-of-failure procedure

1. If the message text or format is incorrect, use the message-to-module table in *z/OS MVS System Messages, Vol 1 (ABA-AOM)*, through *z/OS MVS System Messages, Vol 10 (IXC-IZP)*, to identify the issuing module, and specify the module name as the module keyword.

   If the message was issued at the wrong time or under the wrong conditions, use the same table and specify the name of the detecting module as the module keyword.
2. See "Other DFSMSdfp subcomponents—modifier keywords" on page 60.

Modifier keywords

Use this section to build the modifier keywords that can include function, subfunction, or other modifying information. The modifier keywords are optional but might be necessary to restrict the scope of a software database search to a reasonable number of matches. Use them to describe the unique characteristics of the failure.

You can often identify the failing function and subfunction when you determine the type-of-failure. You might be able to identify a failure-related general register or control block (invalid data, same value every time the failure occurs, and so forth). You should be able to identify the probable failing component from a failure-related indication (message, unique abend code, and so forth).

Procedure

1. Examine the "Standard modifier keyword list" on page 43 to see if any modifier keywords describe an aspect of the failure. Record any such keywords on the "Keyword worksheet" on page 4.
2. For incorrect output or performance types-of-failure, the system might not provide enough information to identify a failure-related module. If, however, you can identify a failure-related function or other modifier, specify them as the modifier keyword(s). See Table 19 on page 60.
3. For all other types-of-failure, select the procedure for the failure-related component from Table 7 on page 42.

Table 7. Procedure for the Failure-Related Component

Subcomponent	Procedure
Catalog Management	See "Catalog management—modifier keywords" on page 59.
DADSM/CVAF	See "DADSM/CVAF—modifier keywords" on page 46.

Table 7. Procedure for the Failure-Related Component (continued)

Subcomponent	Procedure
DASD Common Services	See "DASD common services—modifier keywords" on page 50.
ISMF	See "ISMF—modifier keywords" on page 51.
Media Manager	See "Media Manager—modifier keywords" on page 52.
OAM	See "Object Access Method —modifier keywords" on page 55.
OPEN/CLOSE/End-of-Volume (Common)	See "OPEN/CLOSE/End-of-Volume (common)—modifier keywords" on page 55.
SMS	See "Storage Management Subsystem—modifier keywords" on page 56.
All other DFSMSdfp subcomponents	See "Other DFSMSdfp subcomponents—modifier keywords" on page 60.

4. If SYS1.LOGREC contains a failure-related hardware record, identify the status and hardware device type using the following list of modifier keywords.

Standard modifier keyword list

Table 8 on page 43 lists modifier keywords that demonstrates the standards you must follow when specifying them. Keywords preceded by an asterisk (*) indicate invalid keywords. The invalid keyword description provides the correct keyword.

Table 8. Standards for Modifier Keywords

Keyword	Meaning
ABEND	Use for a system abend in a context that does not relate to a specific code
ABEND*nnn*	Use ABEND*nnn* for a system abend (for example, ABEND0C1, ABEND0F4)
ABENDU*nnnn*	Use ABENDU*nnnn* for a user abend (for example ABENDU0845)
***Access method services**	Use IDCAMS
APAR *nnnnn	Use OY*nnnnn* (for example, OY12345)
***CA split**	Use CI/CA SPLIT
CC*xx*	Condition code (exactly as the system presents it)
CHAN	Channel
***CI split**	Use CI/CA SPLIT
CI/CA SPLIT	Control interval and/or control area split
***Catalog**	Use VSAMCAT
***Catalog Management**	Use CATMGT
CATMGT	Catalog management function of VSAM
***Checkpoint/Restart**	Use CHKPT
CHKPT	Checkpoint/Restart
CMD REJ	Command reject (hardware sense bit)

Table 8. Standards for Modifier Keywords (continued)

Keyword	Meaning
***Command reject**	Use CMD REJ
***Condition code**	Use CC*xx* (exactly as the system presents it)
***Control block**	Use the standard control block identifier, for example, DCB, AMB, RPL, and so forth
COREOVERLAY	Main storage overlay or overlaid
CPC	Channel program check (hardware sense bit)
CREG*nn*	Control register
***Cross-Memory mode**	Use XMM
DCK	Data check (hardware sense bit)
***Data Language/I function of IMS**	DL/I
Device Type *nnnn	Use D/T*nnnn* (for example, D/T3380)
DI*nnnnnnnn*	DADSM diagnostic information (*nnnnnnnn* is four hex bytes)
D/T*nnnn*	Device type related to problem, if only one device type experiences the failure
***Duplicate Records**	Use DUPREC
DUPREC	Failure causes duplicate records in a data set
EQC	Equipment check (hardware sense bit)
ESDS	Entry sequenced data set
***Format-*n* DSCB**	Use F*n*DSCB
F*n*DSCB	Type of DSCB (*n* is a decimal digit, for example F4DSCB)
***Hardware program check**	Use PROGCK
***High-Impact Pervasive problems**	Use HIPER
ICC	Interface control check (hardware sense bit)
IDCAMS	Access method services
Incorrect output, unrelated to performance degradation	Use INCORROUT
INDEX	Data set index (used when describing data set damage)
***Instruction**	Use INST/*xxx*
INST/*xxx*	Failing instruction mnemonic (for example, INST/MVCL)
Initial Program Load of the system	Use IPL
KSDS	Key sequenced data set
Label *xxxxxx	Use LBL*xxxxxx* (for example, LBLNAME01). **Note:** For PLS-generated labels, omit the @.
LBL*xxxxxx*	Program listing statement label *xxxxxx*
LDS	Linear data set
***Lost records**	Use MISSREC
***Manuals**	Use manual order number without hyphens (for example, GC264149)

Table 8. Standards for Modifier Keywords (continued)

Keyword	Meaning
***Master catalog**	Use MASTCAT or MCAT
MASTCAT	Master catalog (MCAT is also permissible)
MCAT	Master catalog (MASTCAT is also permissible)
***Missing records**	Use MISSREC
MISSREC	Missing records from a data set
***Model number (hardware device)**	Use D/T with a three-digit model number (D/T3380 023)
NCA	Not capable (tape hardware sense bit)
***No record found**	Use NRF
NRF	No record found in data set
***OPEN/CLOSE/EOV**	Use O/C/EOV
O/C/EOV	OPEN/CLOSE/End-of-Volume (common) subcomponent of DFSMSdfp
***Out of sequence**	Use OUTOFSEQ
OUTOFSEQ	Record keys out of sequence
OVR	Overrun (hardware sense bit)
PROGCK*xx*	Hardware program check (for example, PROGCK01, PROGCK0C)
PTF *nnnnn	Use UY*nnnnn* (for example, UY12345)
***Publications**	Do not use hyphens (for example, GC264149)
***Return Code, decimal value**	Use RC*nn* (exactly as the system presents it, for example RC04, RC12)
RC*nnn*	Return code, reason code, function code, error code, and so forth (exactly as the system presents it)
***Reason code, hex value**	RS*xxxxxxxx* (for example, RS0418003C). Use RSN*xxxxxxxx* for PDSE
***Record Management**	Use RECMGT
RECMGT	Record management function of VSAM
***Record Management Trace**	Use RMTRACE
***Record-Level Sharing**	Use RLS or VSAMRLS
***Registers**	Use REG*nn* for general purpose registers, CREG*nn* for control registers, and FPREG*nn* for floating point registers, where (*nn* is either one or two decimal digits with leading zeros.)
REG*nn*	General purpose register (use two decimal digits, for example, REG02 or REG14)
***Release Level**	Use R*nnn*
RLS	Record-Level Sharing
RMTRACE	Record management trace facility of VSAM
RPLFDBWD*nnnnnnnn*	RPL feedback-word (*nnnnnnnn* is four hexadecimal bytes)
RRDS	Relative record data set

Table 8. Standards for Modifier Keywords (continued)

Keyword	Meaning
***Share options**	Use SHROPT
SHROPT	VSAM share options
SKC	Seek check (DASD hardware sense bit)
***Storage Overlay**	Use COREOVERLAY
***Supervisor Call**	Use SVC*nn* or 0A*nn*
SVC*nn*	Supervisor call instruction (*nn* is decimal, no leading zeros). (See 0A*nn* for hexadecimal
UCAT	User catalog (USERCAT is also permissible)
UCK	Unit check (hardware sense bit)
UEX	Unit exception (hardware sense bit)
USERCAT	User catalog (UCAT is also permissible)
***User catalog**	Use UCAT or USERCAT
***Volume table of contents**	Use VTOC
***VSAM Record-Level Sharing**	Use VSAMRLS or RLS
VSAMCAT	Catalog
VSAMRLS	VSAM Record-Level Sharing
VTOC	DASD volume table of contents
WAIT	Task or job waiting for an event
WAIT*xxx*	System disabled wait with hex code of 'xxx' (for example, WAIT04F)
WRI	Write inhibit switch should be checked (DASD hardware sense bit)
***Wrong record**	Use WROREC
WROREC	Wrong record in data set
0A*nn*	Supervisor call instruction (*nn* is hexadecimal)

DADSM/CVAF—modifier keywords

Use this section to build the modifier keywords (function and subfunction) for DADSM and CVAF. The modifier keywords are optional but might be necessary to restrict the scope of a software database search to a reasonable number of matches. Use them to identify a specific problem area.

Common procedure

For a description of the conditions associated with nonzero return codes and status codes related to failures in the DADSM functions, see Chapter 7, “DADSM/CVAF diagnostic aids,” on page 109. Some DADSM functions can return 4 bytes of diagnostic information (in the accompanying IEC614I message), which you should specify as modifier keywords. This diagnostic information consists of the DADSM error code, the subfunction identifier, the subfunction return code, and the subfunction reason code.

1. The DADSM return code is provided in register 15. Sometimes, it is provided in register 13 for extended data sets. Specify the return code exactly as the system provides it.

   Example: If the return code is 04, specify the modifier keyword as RC04.

2. The 4 bytes of diagnostic information are provided in register 0. See "DADSM return and diagnostic codes" on page 126 for further information. If the diagnostic information bytes are relevant to your failure, specify them as a single modifier keyword in the following order:

   - DADSM error code
   - Subfunction identifier
   - Subfunction return code
   - Subfunction reason code

   Example: If the 4 bytes are 08170486, specify the modifier keyword as RSN08170486.
3. Continue below with the applicable procedure.

Abend failure modifier keywords procedure

Two function tables exist—one for DADSM functions (see Table 9 on page 47), and one for CVAF functions (see Table 10 on page 48).

- If the failure occurred in a DADSM function, use the DADSM function table; if it occurred in a CVAF function, use the CVAF function table.
- Record the function and/or subfunction as modifier keywords on the "Keyword worksheet" on page 4.

  Example: If the function is ALLOC and the subfunction is INIT, specify the modifier keywords as shown.

  ```
  ALLOC INIT
  ```
- See Table 19 on page 60.

Table 9. DADSM Functions

Function	Subfunction	Description
COMMON		Miscellaneous internal DADSM functions
CREATE (ALLOCATE)	INIT	Initializes space allocation and routes requests to the appropriate subfunction
	NONIALL	Allocates space
	CONV	Converts a VSE VTOC, or an indexed VTOC whose VTOC index has been disabled, to the standard nonindexed VTOC format, enabling DADSM routines to be used,
EXTEND		Allocates secondary space for data sets
LSPACE		Calculates available space on a direct access volume and optionally writes SMF record 19
OBTAIN		Reads DSCBs from the VTOC
PRLSE		Frees unused space
PROTECT		Maintains the system password data set
RENAME		Renames data sets
SCRATCH		Deletes a data set or VSAM data space
VRF		Recovers from interruptions in DADSM for indexed VTOCs

Table 10. CVAF Functions

Function	Subfunction	Description
CM		Initializes CVAF processing and provides services to other CVAF functions
DM		Uses the maps in the VTOC index to manage space on a direct access device
DS		Reads and writes DSCBs and VTOC index records
FI		Reads sets of DSCBs for one or more data sets specified by a qualifier list
GT		Prompts the system operator through WTOR for: • Unit addresses for which GTF tracing of CVAF events is to be started or stopped. • CVAF debug/trap options.
IX	ADD	Adds an index entry to an indexed VTOC
	DEL	Deletes an index entry from an indexed VTOC
	SRCH	Retrieves a specified entry in a VTOC index entry record
LS		Responds to a volume demount system event and invalidates the VTOC information block
SM		Notifies SMS of space usage changes on a system-managed volume
TS		Tests for the presence of CVAF on the system and for an indexed VTOC on a specific volume
VO		Initializes the VTOC information block
VR		Records and retrieves VTOC recovery data in the VTOC index

Incorrect output failure modifier keywords procedure

You might find it difficult to determine either the function orthe module that caused the problem. One DADSM or CVAF function might modify and write the VTOC or VTOC index record, that causes a problem some other unrelated function detects later. Or, you might detect the problem in a listing of the VTOC or VTOC index.

1. Use the CVAF component trace as described in “CVAF component trace” on page 109 to trace the CVAF events.
    a. Examine the trace records to determine the events leading to the failure as described in “CVAF component trace” on page 109.
    b. You can usually identify the function that caused the incorrect output by determining the macro that was issued. If it was a DADSM macro, the failing subcomponent is DADSM; go to Step “1.d” on page 49. If it was a CVAF macro, the failing subcomponent is CVAF; go to Step “1.e” on page 49. If you cannot determine whether the failing subcomponent is DADSM or CVAF, obtain the

component identifiers for both subcomponents in Table 19 on page 60. Record them for use in the software database search procedure. Build two keyword strings, one with each component identifier.

c. See Table 19 on page 60.

d. See Table 9 on page 47 to determine the failing DADSM function. Code the keywords exactly as they are shown in the table.

   The DADSM function modifier keywords are merely abbreviations of the related DADSM functions. The SCRATCH, RENAME, OBTAIN, PARTREL, and PROTECT macros are DADSM macros and they invoke DADSM functions. SVCs exist for each of these macros, as well as for ALLOC and LSPACE. System internal calls invoke the DADSM extend and partial release functions so you must identify these functions from failure-related events.

   See Table 19 on page 60.

e. See Table 10 on page 48 to determine the failing CVAF function; code the keyword(s) exactly as they are shown in the table. All CVAF macros except CVAFTST are associated with a CVPL. For those macros, go to Step "1.j" on page 49. For CVAFTST, continue with the next step.

f. If the CVAFTST macro returned an invalid return code, specify TS as the modifier keyword.

g. If the CVAF GTF trace data is invalid, specify GT as the modifier keyword.

h. If a volume demounted by the system did not cause CVAF to rebuild the VIB, either the CVAF LS function or the function that issued the demount (OPEN, EOV, VSAM OPEN, Scheduler) failed. Specify the modifier keyword accordingly.

   Note: Demount is not a DADSM/CVAF function.

i. See Table 19 on page 60.

j. Locate the CVPL, which is in the DADSM work area. Use the eye catcher by scanning the readable portion of the DADSM work area. The CVFCTN field (offset 6 in the CVPL) contains a 1-byte CVAF function code.

k. Identify the CVAF function keyword corresponding to the value in the CVAF function byte and specify it as the modifier keyword. The possible values of the CVAF function byte and the corresponding function keywords are shown in Table 11 on page 49.

Table 11. CVAF Function Byte Values in the CVPL

Value of CVAF Function Byte	Function Keyword	Subfunction Keyword
01 (X'01')	DS	—
02 (X'02')	DS	—
03 (X'03')	DS	—
04 (X'04')	DS	—
05 (X'05')	DS	—
06 (X'06')	IX	ADD
07 (X'07')	IX	DEL
08 (X'08')	DM	—
09 (X'09')	DM	—
10 (X'0A')	DM	—
11 (X'0B')	VO	—
12 (X'0C')	VR	—
13 (X'0D')	VR	—

Table 11. CVAF Function Byte Values in the CVPL (continued)

Value of CVAF Function Byte	Function Keyword	Subfunction Keyword
14 (X'0E')	FI	—
15 (X'0F')	FI	—
16 (X'10')	FI	—

See Table 10 on page 48 for a detailed description of the operations performed by the CVAF functions.

l. A previous VTOC or index update might have caused the present problem. Use IEHLIST to obtain a listing of the contents of the VTOC and the VTOC index. To list the index, specify INDEXDSN=SYS1.VTOCIX.V*nnnnnn*, in which *nnnnnn* is the name of your VTOC index.

m. See Table 19 on page 60.

Message failure modifier keywords procedure

1. Using the module name determined in “DADSM/CVAF—module keyword” on page 34, locate that module name in either “CVAF module-to-function cross-reference” on page 109 or “DADSM module-to-function cross-reference” on page 125 to determine any applicable modifier (function and subfunction) keywords.

   Example: If the module name is IGG0325B, specify the modifier keywords as shown:

   ```
   ALLOC NONIALL
   ```

2. See Table 19 on page 60.

DASD common services—modifier keywords

Use this section to build the modifier keywords for DASD common services. The modifier keywords are optional but might be necessary to restrict the scope of a software database search to a reasonable number of matches. Use them to identify a specific problem area.

Procedure

Use the following fields in the problem determination area (PDA) as modifier keywords except for common filter services, which neither employ an ESTAE nor use a PDA:

- Failure-related module
- Most recent calling function
- Return code
- Reason code.

The failure-related ISPF log entry's FEEDBACK field contains the PDA information. See Figure 28 on page 227 for a sample entry containing PDA information. See “DASD common services—problem determination area (PDA)” on page 187 for a description of the PDA contents.

1. Specify the contents of the PDA fields as modifier keywords.

   Example: If the module name is IGBDIS01, the return code is 10, and the reason code is 038, specify the modifier keywords as:

   ```
   IGBDIS01 RC10 RC038
   ```

2. See Table 19 on page 60.

ISMF—modifier keywords

Use this section to build the modifier keywords for ISMF. The modifier keywords are optional but might be necessary to restrict the scope of a software database search to a reasonable number of matches. Use them to identify a specific problem area.

Procedure

ISMF uses an entry in the ISPF log data set to record information that can help in diagnosing a problem. See "ISMF's use of the ISPF log" on page 226 for a description of the ISPF log and ISMF logging options.

ISMF also provides the ERTB command, which displays failure symptoms that you can specify as search keywords. See "Using the logging/abend control entry panel" on page 222 and "Using the ERTB display panel" on page 223 for details.

1. You can use the contents of the following ISPF log entry fields as modifier keywords:

   Application identifier
   Function identifier
   Module name
   Procedure identifier
   Last panel identifier
   Service
   Feedback
   Return code
   Reason code

2. If you can relate a log entry to the error, use Table 12 on page 51 to determine which log information to specify as modifier keywords.

   - For abend failures, specify the module you determined in "ISMF—module keyword" on page 36 as the module name keyword.
   - For message-indicated failures, specify the return code and reason code you determined in "ISMF—message keyword" on page 20 as modifier keywords. (Specify them exactly as the system presented them.)

Table 12. ISMF—Information Needed to Build the Modifier Keywords

Type-of-Failure	Modifier
Abend	Application identifier Function identifier Last panel
Message	Application identifier Function identifier Module identifier Procedure Last panel Service Feedback Return code Reason code
Incorrect Output	Application identifier Function identifier Return code Reason code Last panel
Wait/Loop	Application identifier Function identifier Return code Reason code Last Panel
Performance	Application identifier Function identifier Return code Reason code Last panel Service Feedback

3. If you cannot relate an ISPF log entry to the failure (incorrectoutput, wait, or loop types-of-failure), use Table 13 on page 51 to determine which modifier keywords to specify.

Table 13. ISMF—Modifier Keywords to Use When the Log Is Not Available

Failing Function	Keyword
Primary Option Menu	MAIN DLG

Table 13. ISMF—Modifier Keywords to Use When the Log Is Not Available (continued)

Failing Function	**Keyword**
Profile	PROFILE
Application selection	SELECT
Initializing an application	INIT
Displaying a list	DISPLIST
Line operators	Line operator name
Commands	Command name
Exiting ISMF	TERM

4. Continue with “Adding modifiers to the keyword string” on page 52.

Adding modifiers to the keyword string

1. Specify (in any order) the appropriate modifiers as keywords.

   The sample log entry in Figure 18 on page 52 illustrates a message-indicated failure.After specifying MSGDGTDS004 as the type-of-failure keyword, specify the values listed after the applicable labels (as identified in Table 12 on page 51). The application modifier keyword is DATA SET.

```
*****  ISMF ERROR  ***** - APPLICATION(DGT1 - DATA SET); FUNCTION(SELECT)
                         - RETURN CODE(0008); REASON CODE(0006)
                         - MODULE(DGTFDS03); PROCEDURE(GENAARAY)
                         - MESSAGE ID(DGTDS004 - DGTDS004); LAST PANEL(DGTDDDS1)
                         - SERVICE(DGTFDS04); FEEDBACK(NONE)
```

   Figure 18. ISMF—Sample ISMF Entry in ISPF Log

2. To add modifier keywords to the keyword string, use only the significant log entry field values as keywords, ignoring any fields containing duplicate values. For example, specify the keyword string for the log entry sample in Figure 18 on page 52 as follows:

```
MSGDGTDS004 DATA SET SELECT RC8 RC6 DGTFDS03 GENAARAY DGTDDDS1 DGTFDS04
```

3. See Table 19 on page 60.

Media Manager—modifier keywords

Use this section to build the modifier keyword (function) for the media manager. The modifier keywords are optional but might be necessary to restrict the scope of a software database search to a reasonable number of matches. Use them to identify a specific problem area.

Two procedures follow. The first is for abend, wait, or loop types-of-failure. The second is for identifying a probable failing function if the first procedure does not, or if the failure is other than an abend, wait, or loop.

Abend/Wait/Loop modifier keyword procedure

To determine the failing media manager function, use the following procedure to locate the calling program and identify the type of call.

1. Use the dump obtained earlier for either the abend or wait/loop type-of-failure.
2. Register 13 at the time of the abend points to the save area of the media manager's caller. Standard save area conventions are followed. Linkage to media manager is by registers 14 and 15. Parameters are passed in registers 0, 1, and 2. In DFSMS, media manager uses the BAKR instruction, and the registers are saved on the linkage stack. The registers are not saved in the standard storage area. It is possible to format the linkage stack with IPCS.

3. Locate in the dump the save area pointed to by register 13. Locate register 14 in the save area; it points to the calling program's next sequential instruction following the instruction that called media manager.
4. Locate in the dump the instruction in the calling program that passed control to media manager. The instruction is similar to the one shown:

```
Machine         Assembler
Language        Language
Instruction     Instruction     Usage

440E044A        EX 0,1098(14)   (Common for all linkages)
```

5. Trace backward from that instruction, looking for the following sequence of linkage instructions:

```
Machine         Assembler
Language        Language
Instruction     Instruction

58E00010        L 14,16(0)
58F0E148        L 15,328(,14)
58F0F010        L 15,16(,15)
```

6. One of the instructions shown in Table 14 on page 53 follows this sequence of instructions. It identifies the media manager function invoked by the caller.

Table 14. Identifying the Media Manager Function Invoked

Machine Language Instruction	Assembler Language Instruction	Modifier (Function) keyword
58F0F008	L 15,8(,15)	INIT
58F0F00C	L 15,12(,15)	RDWR
58F0F010	L 15,16(,15)	FMTWR
58F0F014	L 15,20(,15)	PFMT
58F0F018	L 15,24(,15)	CNVT (RBA TO CCHHR)
58F0F01C	L 15,28(,15)	CNVT (CCHHR TO RBA)
58F0F020	L 15,32(,15)	SRV
58F0F028	L 15,40(,15)	COMMIT, DISCARD
58F0F030	L 15,48(,15)	WRITE
58F0F034	L 15,52(,15)	PIO
58F0F038	L 15,56(,15)	PIOPG
58F0F03C	L 15,60(,15)	PIOC

Tip: Because I/O and program processing are asynchronous, the most recent program activity might not involve a media manager call. However, by looking for these linkage instructions, you can locate the most recent media manager calls to the INIT, CNVT, or SRV functions.

7. Specify the media manager function as the modifier keyword.

   Example: If the function is CNVT, specify the keyword as CNVT.
8. If the function is neither INIT, CNVT, nor SRV, continue with this process. Otherwise, see Table 19 on page 60.
9. The function must be either RDWR, FMTWR, or PFMT. Continue with this procedure.
10. If functional recovery routines are in effect and a program check or abend occurs either in the media manager or in an exit routine from the media manager, the system automatically directs a summary dump of media manager control blocks to the SYS1.DUMPxx data set.

    When printing a dump from a SYS1.DUMPxx data set, print at least the summary dump portion of the dump, using the IPCS dump formatter (see *z/OS MVS IPCS User's Guide* for details on using this service). If ICYFRR issued the SDUMP, use the sample media manager dump in Chapter 12, “Media Manager diagnostic aids,” on page 275 for the following steps:

11. Locate in the dump the media manager process block (MMPB). It is near the beginning of the summary dump, and the acronym MMPB in EBCDIC representation appears at offset 0 in the MMPB itself.

    If you cannot locate the media manager process block this way, use the following procedure:

    a. Locate the MMVT by using the CVTMMVT field in the CVT and look for the first storage vector MMSV by using the eye catcher ICYMMSV1. See the sample media manager dump in Chapter 12, “Media Manager diagnostic aids,” on page 275.
    b. Each two-word MMSV entry following the eye catcher consists of a lockword (unused or caller's ASID) followed by a pointer to the media manager storage block containing the MMPB(s). Multiple MMPBs exist, some of which are active and some inactive; to find the one that might be related to the failure, examine the MMPASID and MMPBPARM fields for the ASID and input parameters, respectively.
12. The MMPFLG2 field in the MMPB is a 1-byte field that identifies which media manager function was in control.

MMPFLG2 Value	Modifier (Function) Keyword
X'08'	PFMT
X'10'	FMTWR

    Example: If the function is PFMT, specify the modifier keyword as PFMT.

    If neither value is present, the function, by default, is RDWR.
13. See Table 19 on page 60.

Alternate modifier keyword procedure

Use this procedure if the previous procedure has not isolated the failing media manager function, or if the type-of-failure is other than abend, wait, or loop.

Function keywords divide the media manager into logical areas. A unique acronym identifies each of these functions.

1. From the following list, select the modifier (function) keyword that describes the area you suspect of failure.

Function	Description
CNVT	Converts relative byte addresses to absolute device addresses (CCHHR), and vice versa. The CNVT function uses modules ICYCCHHR and ICYRBA.
FMTWR	Processes requests to format and write tracks sequentially. The FMTWR function uses modules ICYABN, ICYABN, ICYBLDCP, ICYBLIST, ICYDIE, ICYFW, ICYNRM, ICYPGAD, ICYRBA, and ICYSTOR.
INIT	Initializes the media manager control blocks. The INIT function uses modules ICYIEDB, ICYILPMB, and ICYINIT.
MMSYSTEM	Manages media manager resources. The MMSYSTEM function uses modules ICYELE, ICYFRR, and ICYPURG.
PFMT	Processes requests to preformat tracks within a range of relative byte addresses. The PFMT function uses modules ICYCCHHR, ICYPFCP, ICYPFMT, ICYRBA, ICYSTOR, ICYPFDIE, and ICYPFAPP.

Function	Description
RDWR	Processes requests to read or write the data portion of records. The RDWR function uses modules ICYABN, ICYBLDCP, ICYBLIST, ICYDIE, ICYNRM, ICYPGAD, ICYRBA, ICYRDWR, ICYSORT, and ICYSTOR.
SRV	Set of functions that interface with a catalog to obtain data set information and to update that information when explicitly requested. The SRV function uses modules ICYMMSRV and ICYMSTAE.

2. Specify the modifier keyword as FMTWR.
3. See Table 19 on page 60.

Object Access Method —modifier keywords

The following common modifier keywords are used to diagnose specific problem areas of OAM subcomponents.

Function	Description
OVTOC	Optical volume table of contents
RC*nn.*	Return code
RSN*nn*	Reason code
XRC*nn*	External return code
XRSN*nnnnnnnn*	External reason code
STORCLASS	Storage class
MGTCLASS	Management class
STORGROUP	Storage group
OAM	Object access method
OSMC	Object access method storage management component
OTIS	Object access method thread isolation support
OSR	Object storage and retrieval
LCS	Library control system
3495	IBM TotalStorage™ Enterprise Automated Tape Library (3495)
3995	IBM 3995 Optical Library Dataserver
MO	Magneto-optic (erasable optical media)
WORM	Write once read many (nonerasable optical media)

OPEN/CLOSE/End-of-Volume (common)—modifier keywords

Use this section to build the modifier keywords for common OPEN/CLOSE/End-of-Volume functions. The modifier keywords are optional but might be necessary to restrict the scope of a software database search to a reasonable number of matches. Use them to identify a specific problem area.

Procedure

1. Except for the first load modules of common OPEN/CLOSE/End-of-Volume functions (listed below), the first 3 characters of most module names are IFG, and the fourth through sixth characters of the module name identify the function. Select a function from Table 15 on page 56.

Table 15. List of Entry Points

Module Name	Function Keyword
IFG019	OPEN
IFG020	CLOSE
IFG023	TCLOSE
IFG055	EOV
IFG0RR	RECOVERY
IFG0TC	TERMINATION
IGC0001I	OPEN
IGC00020	CLOSE
ICG0002C	TCLOSE
IGC0005E	EOV
IGC0002B	OPENJ
IGC0003A	FEOV (force end-of-volume)
IGC0006D	RDJFCB (read JFCB)
IGC0010C	XLATE (translate to and from ASCII (BSAM and QSAM))

Example: If the function is OPEN, specify the modifier keyword as OPEN.

2. See Table 19 on page 60.

Partitioned data set extended—modifier keywords

Use this section to build the modifier keywords for Partitioned Data Set Extended data sets when you have received ABEND0F4 dumps or LOGREC records. The ABEND0F4 dump title or LOGREC record will provide the failing CSECT name. A return code and reason code should always be associated with the module name when building keywords for an ABEND0F4.

Procedure

1. If a LOGREC record is available for an ABEND0F4, locate the registers at the time of abend in the LOGREC record. Specify the contents of register 15 as the return code, and the contents of register 0 as the reason code.
2. If an SVC dump is available for an ABEND0F4, locate the return code and reason code that appear in the dump title.

Storage Management Subsystem—modifier keywords

Use this section to build the modifier keywords for the Storage Management Subsystem. The modifier keywords are optional but might be necessary to restrict the scope of a software database search to a reasonable number of matches. Use them to identify a specific problem area.

If SMS issues either message IGD300I or IGD306I, use the SUMDUMP or SYS1.LOGREC information respectively to identify failure-related modifier keywords.

Procedure

1. Specify modifier keywords based on the failure-related SMS function or process. Use Table 16 on page 57 to identify any applicable keywords. Record them on the "Keyword worksheet" on page 4 as modifier keywords.

Table 16. SMS General Keywords

Keyword	Failure-Related Command
ACDS	N/A
ACS	N/A
AVGREC	N/A
COMMDS	N/A
DATACLAS	N/A
DEVSERVSMS	Operator command 'DEVSERV SMS'
DISPLAYSMS	Operator command 'DISPLAY SMS'
DO	N/A
DSNTYPE	N/A
END	N/A
EXIT	N/A
FILENUM	N/A
IART	N/A
IF	N/A
KEYLEN	N/A
KEYOFF	N/A
LABEL	N/A
LIBNAME	N/A
LIKE	N/A
MGMTCLAS	N/A
PGM	N/A
PROC	N/A
REFDD	N/A
SCDS	N/A
SECMODEL	N/A
SETSMS	Operator command 'SETSMS'
SETSMSEQ	Operator command 'SET SMS='
SMSDATA	IPCS verbexit 'SMSDATA'
STORCLAS	N/A
STORGRP	N/A
VARY	Operator command 'VARY' (not 'VARY SMS')
VARYSMS	Operator command 'VARY SMS'

For problems involving ACS routine execution, specify as keywords any applicable variables and statements from Table 17 on page 58. For example, if the variable is ACSENVIR, specify ACSENVIR as the modifier keyword.

Table 17. ACS Variable and Statement Keywords

Keyword	Keyword	Keyword
ACCT_JOB	ACCT_STEP	ACSENVIR
ALLVOL	ANYVOL	APPLIC
BLKSIZE	DATACLAS	DD
DEF_DATACLAS	DEF_MGMTCLAS	DEF_STORCLAS
DO	DSN	DSNTYPE
DSORG	DSOWNER	DSTYPE
END	EXIT	EXPDT
FILENUM	FILTLIST	GROUP
HLQ	IF	JOB
LABEL	LIBNAME	LLQ
MAXSIZE	MEMHLQ	MEMLLQ
MEMN	MEMNQUAL	MGMTCLAS
MSPARM	MSPDEST	MSPOLICY
MSPOOL	MSVGP	NQUAL
NVOL	PGM	PROC
RECORG	RETPD	SECLABL
SELECT	SET	SIZE
STORCLAS	STORGRP	SYSNAME
SYSPLEX	UNIT	USER
WRITE	XMODE	

For problems related to serialization of SMS resources, specify the applicable resource names as shown in Table 18 on page 58.

Access to a PDSE is serialized with an enqueue that uses SYSZIGW0 and SYSZIGW1.

GRS can be used to propagate the enqueue names as global resources. This insures PDSE integrity across multiple systems.

Example:. If the major resource name is SYSZIGDI and the minor resource name is SMS VECTOR TABLE, specify the keywords as:

```
SYSZIGDI SMS VECTOR TABLE
```

Table 18. Resource Name Keywords

Major Name	Minor Name
IGDCDS	N/A
IGDCDSXS	N/A
SYSZIGDI	ICMRT CMDSADDR LOCKED
SYSZIGDI	IGDSSI00
SYSZIGDI	IGDSSI01
SYSZIGDI	SMS VECTOR TABLE
SYSZIGW0	N/A

Table 18. Resource Name Keywords (continued)

Major Name	Minor Name
SYSZIGW1	N/A
SYSIGWD5	PDSEASRESTART

2. See Table 19 on page 60.

Catalog management—modifier keywords

Use this section to build the modifier keywords for the failure-related catalog management subcomponent. The modifier keywords are optional but might be necessary to restrict the scope of a software database search to a reasonable number of matches. Use them to isolate a specific problem area. The keywords identify the following problems:

- Failure-related access method services commands
- Internal catalog management functions

The catalog management subcomponents follow:

- Catalog
- Access method services

Procedure

1. Select the keywords that describe the type of catalog in use at the time of failure. (More than one could be applicable.)
   - ICFCAT
   - MASTCAT (or MCAT)
   - USERCAT (or UCAT)
   - VSAMCAT
2. Select the keyword that identifies the area in which you suspect the failure occurred:
   - ALLOCATE
   - ALTER
   - BINDDATA
   - BLDINDEX
   - CNVTCAT
   - DEFINE ALIAS
   - DEFINE ALTERNATE INDEX
   - DEFINE CLUSTER
   - DEFINE GDG
   - DEFINE MASTERCATALOG
   - DEFINE NONVSAM
   - DEFINE PAGESPACE
   - DEFINE PATH
   - DEFINE SPACE
   - DEFINE USERCATALOG
   - DELETE
   - DIAGNOSE
   - EXAMINE
   - EXPORT

- EXPORTRA
- IMPORT
- IMPORTRA
- LISTCAT
- LISTCRA
- LISTDATA
- PRINT
- REPRO
- RESETCAT
- SETCACHE
- VERIFY

Example: If the failure occurred while you were executing the DEFINE USERCATALOG command, specify the modifier keywords as :

```
DEFINE USERCATALOG
```

3. See Table 19 on page 60.

Other DFSMSdfp subcomponents—modifier keywords

Use this section to build the modifier keywords for DFSMSdfp subcomponents for which no unique procedure exists. The modifier keywords are optional but might be necessary to restrict the scope of a software database search to a reasonable number of matches. Use them to identify a specific problem area. They could identify a function, command name, register name, register contents, and so forth.

Procedure

1. Scan the list of standard modifier keywords in "Modifier keywords" on page 42 and specify any that apply to the failure.
2. Specify the function, command name, or other modifier as it is spelled in a message, information, or other identifying material.
3. See Table 19 on page 60.

Component identification keyword

Use this section to identify the suspected failing DFSMSdfp component in your keyword string.

1. Table 19 on page 60 lists the component identification keyword for each DFSMS component. Specify the entire nine-digit component identification shown in the table as the keyword.
2. See "Release level keyword" on page 63.

Table 19. Component Identification Keywords

Component Name	Component Identification Keyword	Subcomponents
DFSMS Installation	5695DF100	DFSMSdfp INSTALL
Storage Management Subsystem	5695DF101	Storage Management Subsystem

Table 19. Component Identification Keywords (continued)

Component Name	Component Identification Keyword	Subcomponents
Base Access Methods	5695DF102	SAM
		DAM
		PAM
		Common Adapters
		VIO
		SAM Subsystem Interface
		3505/25 Reader/Punch
Access Method Services	5695DF103	Access Method Services (IDCAMS)
DFSMS Common Services	5695DF104	Common Measurement Manager
		DASD Common Services
Catalog	5695DF105	Catalog
VSAM/Media Manager	5695DF106	Media Manager
		VSAM O/C/EOV
		VSAM Record Management
Device Management Services	5695DF107	OPEN/CLOSE/End-of-Volume
Program Management	5695DF108	Linkage Editor
		LINK/LOADGO Prompter
		Loader
		AMBLIST
		Overlay Supervisor
		PROGRAM FETCH
		IGWSPZAP
		Program Management Binder
		Program Management Loader
Checkpoint Restart	5695DF109	Checkpoint Restart
Device Support: Tape/Unit Record	5695DF110	Tape ERP
		Unit Record ERP
		MSGDISP (Message Description Macros)
		TAPE/Unit Record Device Description Table
		SIO Exits— Tape
		IEHINITT
		IFHSTATR
Device Support: DASD	5695DF111	DASD ERP
		DASD Device Description Tables
		SIO EXITS— DASD

Table 19. Component Identification Keywords (continued)

Component Name	Component Identification Keyword	Subcomponents
Device Support Services	5695DF113	Device Console Services (DEVSERV command)
		Asynchronous Operations Manager (AOM)
		Cache Device Services
Utilities	5695DF114	IEHLIST
		IEHPROGM
		IEHMOVE
		IEBIMAGE
		IEBUPDTE
		IEBEDIT
		3800 OFFLINE Utility
		IEBPTPCH
		IEBGENER
		TSO/E Utilities
		IEBDG
		IEBCOMPR
		IEBCOPY
Partition Data Services Extended (PDSE)	5695DF115	Buffer Management Facility (BMF)
		Common Data Manager (CDM)
		Common Lock Manager (CLM)
		Index Management Facility (IMF)
		System Services Facility (SSF)
		File Attributes Manager (FAMS)
Volume Mount Analyzer	5695DF116	Volume Mount Analyzer (VMA)
		SMF extract utility (GFTAXTR)
		Volume Mount Analyzer (GFTAVWA)
System Data Mover	5695DF117	Concurrent Copy Manager
Compression Management Services	5695DF118	Compression Management Services (CMP)

Table 19. Component Identification Keywords (continued)

Component Name	Component Identification Keyword	Subcomponents
VSAM Record-Level Sharing	5695DF122	Buffer Management Facility - Local Cache (BLC)
		Common Error Recovery Services (CERS)
		Measure Management Facilities (MMF)
		RLS Diagnostic Services (RDS)
		Service Queue Manager (SQM)
		Shared Hierarchy Management (SHM)
		Shared Latch Services (SLCH)
		Sharing Control (SHC)
		Storage Management Locking Services (SMLS)
		Storage Management Protocol Manager (SMPM)
		VSAM RLS OPEN/CLOSE/End-of-Volume (VOC)
		VSAM RLS Quiesce (VQUI)
		VSAM RLS Record Management (VRM)
		VSAM RLS Server Services (VSS)
NaviQuest	5695DF123	None
Device Management Services	5695DF133	CVAF
		DADSM
		DEVMAN address space
		Password Protect
ISMF and HCD	5695DF161	Hardware Configuration Dialog
		ISMF
		ISMF English Panels
		ISMF Japanese Panels
OAM	5695DF180	Object Storage and Retrieval (OSR)
		OAM Storage Management Component (OSMC)
		Library Control System (LCS)

Release level keyword

Using the release level keyword when searching the software database is optional. However, the release level keyword is required for an APAR.

Including the release level keyword in your keyword string will limit the software database search to problems reported against that release.

Tip: While this might reduce the number of matches against your keyword string, it could inadvertently eliminate the software record you hoped to find. The problem might have been reported against a different release. Use “Techniques for varying the search argument” on page 3 to avoid this.

Use the following procedure to determine the release level keyword for the failure-related DFSMSdfp module on your system.

Procedure

Use one of the following methods to obtain the DFSMSdfp FMID. Refer to Table 20 on page 64 to find the corresponding release level keyword.

- Method A—Using SMP/E (requires knowing the module name)
- Method B—Using the Module Copyright Area (requires knowing the module name and having a dump containing the module)

If you cannot determine the module name, bypass this procedure and see “Searching with the keyword string” on page 2.

Method A—using SMP/E

If you suspect a specific module of causing the problem, you can use this procedure to determine the FMID of DFSMSdfp.

1. List the consolidated software inventory (CSI), using the following SMP/E control statements:

   ```
   SET BDY (tgtzone).
   LIST MOD (name) XREF.
   ```

   Be sure to include the period at the end of the control statements.
2. In the NAME column of the CSI, locate the name of the module causing the problem.
3. In the entry for the module, find the FMID field.
4. Find the corresponding release level keyword in Table 20 on page 64.
5. See “Service level keyword” on page 65.

Related reading: For more details on using this procedure, see *SMP/E for z/OS Reference*.

Method B—using the module copyright area

1. Locate the module in the dump by scanning the EBCDIC portion in the address range associated with the failure-related module. The first part of the module contains the following:
   - Copyright statement
   - Module name
   - FMID
   - Service level (PTF number or APAR number) of the module.
2. In the copyright area, find the FMID field.
3. Find the corresponding release level keyword in Table 20 on page 64.
4. See “Service level keyword” on page 65.

Table 20. FMIDs and Corresponding Release Level Keywords

FMID	Release Level Keyword	Product or Element
HDZ1A10	RA10	z/OS 1.10 DFSMS
HDZ1B10	RB10	z/OS 1.11 DFSMS
HDZ1C10	RC10	z/OS 1.12 DFSMS
HDZ1D10	RD10	z/OS 1.13 DFSMS
HDZ2210	R210	z/OS 2.1 DFSMS

Table 20. FMIDs and Corresponding Release Level Keywords (continued)

FMID	Release Level Keyword	Product or Element
HDZ2210	R220	z/OS 2.2 DFSMS

Service level keyword

Use this procedure to determine the service level of a DFSMSdfpmodule installed on your system. The service level of a module is defined as the most current fix applied to that module. It might be any one (but only one) of the following:

- The highest level APAR fix applied to a module.
- The highest level PTF fix applied to a module.
- The FMID of the DFSMSdfp product to which a module belongs (if no fixes have been applied to the module).

The service level keyword is optional in the keyword string, but is required when communicating with the IBM Support Center.

Use this keyword to identify any recently applied software service (APARs or PTFs) that seems to be failure-related.

Procedure

Use one of the following methods to obtain the service level of a DFSMSdfp module:

- Method A—Using SMP/E
- Method B—Using the Module Copyright Area
- Method C—Using the SDUMP Title Page.

The service level is normally adjacent to the module name in a dump.

Method A—using SMP/E

1. List the appropriate target zone of the CSI. The control statements are:

   ```
   SET BDY (tgtzone).
   LIST MOD (name) XREF.
   ```

   in which `tgtzone` is the target zone and `name` is the module name.
2. Using the output listing, locate the name of the failure-related module in the NAME column of the CSI.
3. In the entry for that module, locate the replacement module identifier (RMID) field. The RMID field contains one of the following:

   - An APAR number
   - A PTF number
   - The FMID of the DFSMSdfp product of which the module is a part (if no fixes have been applied to the module).

   Tip: You can also determine a module's service level by using the SMP/E query dialog. For more information, see *SMP/E for z/OS Reference*.
4. To determine whether a specific PTF is installed on your system without knowing what modules it affects, use your installation's SMP procedure as follows:

   a. Using SMP/E, list the SYSMODs; the control statement is:

   ```
   SET BDY (tgtzone).
   LIST SYSMOD (ptf#).
   ```

Related reading: For more details on using this procedure, see *SMP/E for z/OS Reference*.

5. Specify the service level keyword.

   Example: If the service level of the failure related module is UY12345, specify UY12345 as the service level keyword.

6. See "Searching with the keyword string" on page 2.

Method B—using the module copyright area

1. Locate the service level in the copyright information at the beginning of the failure-related CSECT.
2. Specify the service level keyword.

   Example: If the service level of the failure related module is UY12345, specify the service level keyword as UY12345.

3. See "Searching with the keyword string" on page 2.

Method C—using the SDUMP title page

1. Locate the service level in the SDUMP header title.
2. Specify the service level keyword.

   Example: If the service level of the failure related module is UY12345, specify the service level keyword as UY12345.

3. See "Searching with the keyword string" on page 2.

Contacting the IBM Support Center for assistance

Contact the IBM Support Center for assistance in gathering additional documentation. Be prepared to supply the following information:

- The keyword string (or strings) that you have built
- Customer number
- Current release level for MVS/ESA, SP, and DFSMSdfp or z/OS.
- Processor number (type, model, serial)
- Your program update tape level (PUT tape), which is a collection of PTFs and APARs, and any additional APARs and PTFs applied to this release
- An SMP/E listing of installed service available

Chapter 2. General diagnostic aids

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

Notational conventions

This document 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 document 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`}]`

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 "Abend keyword" on page 7.

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 through 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 through IPCS	Operator doing an IPL of the standalone dump program.
DYNAMIC (SVC Dump or SDUMP)	SYS1.DUMP*xx*	Machine readable	Printed and formatted through IPCS	Operator entering DUMP command.
FRR (SVC Dump or SDUMP)	SYS1.DUMP*xx*	Machine readable	Printed and formatted through 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.)

Note:

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 3. 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.

CTRACE subcommand

Table 21 on page 69 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 21. AOM—CTRACE Subcommand

Operation	Parameters

`CTRACE COMP(SYSAOM)`	`{SHORT|FULL}`
	`{GMT|LOCAL}`

Table 21. AOM—CTRACE Subcommand (continued)

Operation	Parameters

	{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.

Altering an AOM CTRACE without using an external output writer

To alter an AOM CTRACE without using an external output writer, perform the steps shown in the following example. This example will trace for JOBNAME AOMV2 and for ASID 3F. It is possible to selectively trace up to 16 JOBNAMES and 16 ASIDs.

1. Issue the operator TRACE command and reply to the resulting message.

   ```
   TRACE CT,ON,COMP=SYSAOM
   *05 ITT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND.
   R 5,JOBNAME=(AOMV2),ASID=(3F),END
   ```

2. Display the trace information for SYSAOM.

   ```
   DISPLAY TRACE,COMP=SYSAOM
   ```

   You should see a response similar to this example:

   ```
   RESPONSE=SYSTEM1
    IEE843I 16.46.07  TRACE DISPLAY 810
            SYSTEM STATUS INFORMATION
     ST=(ON,0001M,00001M) AS=ON  BR=OFF EX=ON  MO=OFF MT=(ON,999K)
      COMPONENT     MODE BUFFER HEAD SUBS
     -------------------------------------------------------------
      SYSAOM        ON
         ASIDS      003F
         JOBNAMES   AOMV2
         OPTIONS    YES
         WRITER     *NONE*
   ```

3. Restore AOM CTRACE to its normal trace level using the TRACE command.

   ```
   TRACE CT,OFF,COMP=SYSAOM
   ```

4. If the input source to IPCS is to be a dump, issue the following command:

   ```
   F DEVMAN,DUMP
   ```

5. Issue the IPCS CTRACE COMP(SYSAOM) subcommand and set the input source to either the dump created in the step above or to active.

   ```
   CTRACE COMP(SYSAOM) FULL LOCAL
   ```

Start an AOM CTRACE using an external output writer

To start an AOM CTRACE using an external output writer, perform the steps shown in the following example. This example will trace for JOBNAME AOMV2 and for ASID 3F. It is possible to selectively trace up to 16 JOBNAMES and 16 ASIDs.

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

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

3. Alter the AOM CTRACE by entering the operator TRACE command and replying to the resulting message.

   ```
   TRACE CT,ON,COMP=SYSAOM
   *05 ITT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND.
   R 5,WTR=CTWTR,JOBNAME=(AOMV2),ASID=(3F),END
   ```

4. Display the trace information for SYSAOM.

   ```
   DISPLAY TRACE,COMP=SYSAOM
   ```

   You should see a response similar to this example:

   ```
   RESPONSE=SYSTEM1
    IEE843I 17.20.13  TRACE DISPLAY 898
            SYSTEM STATUS INFORMATION
     ST=(ON,0001M,00001M) AS=ON  BR=OFF EX=ON  MO=OFF MT=(ON,999K)
      COMPONENT     MODE BUFFER HEAD SUBS
     --------------------------------------------------------------
      SYSAOM        ON
         ASIDS      003F
         JOBNAMES   AOMV2
         OPTIONS    YES
         WRITER     CTWTR
   ```

5. Restore AOM CTRACE to its normal trace level using the TRACE command.

   ```
   TRACE CT,OFF,COMP=SYSAOM
   TRACE CT,WTRSTOP=CTWTR
   ```

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

   ```
   CTRACE COMP(SYSAOM) FULL LOCAL
   ```

AOM trace table Abbreviations

Table 22 on page 71 describes some of the abbreviations that are used to describe the data that is contained in the AOM trace table.

Table 22. 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 22. 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.

Table 22. AOM Trace Table Abbreviations (continued)

Abbreviation	Description
PTHM	Path mask.
PTHI	Channel path identification.
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.
TKPT	Caller's IOC system token pointer
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.

Table 22. AOM Trace Table Abbreviations (continued)

Abbreviation	Description
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 23 on page 74 lists the explanation and actions to take for each code.

Table 23. 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 23. 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.
	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.

Table 23. AOM DASD Return and Reason Codes (continued)

Return Code	Reason Code	Description
	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. Check IOSCOD and IOSXRCOD, saved in AOMSERVL, for more details.
	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.
	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.

Table 23. AOM DASD Return and Reason Codes (continued)

Return Code	Reason Code	Description
	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.
	4 (X'04')	An asynchronous I/O operation ended with an unexpected completion status. For diagnostic purposes, the unexpected status is stored in the SERVL control block at offset '24'X (immediately following SERVRETC and SERVREAS). The SERVL is traced using CTRACE (refer to "Using IPCS to format the AOM trace table" on page 69).
	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 24 on page 77 describes AOM tape library return and reason codes.

Table 24. 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'0404'	REQTYPE(MEDINFO) medium sense does not contain required data (volume may not be mounted)

Table 24. AOM Tape Library Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description
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'0811'	Device not found in any library
		X'0812'	Invalid device type
		X'0813'	Library Manager returned invalid PTP device data
		X'0814'	All Device Data requested but not supported by microcode
		X'0815'	I/O error trying to read All Device Data record sets.
		X'0816'	OPM type request, but not supported by microcode.
		X'0817'	SERVEXIT address is invalid (must be DFVLAPR
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'0C07'	EJECT failed - 1600 library orders are already queued
		X'0C08'	Two 4 minute timeouts (library I/O did not complete in 4 minutes on the first device selected, a second device was selected and it too did not complete in 4 minutes).
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.
		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.

Table 24. AOM Tape Library Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description
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.

AOM QUERY FlashCopy return and reason codes

The Asynchronous Operations Manager QUERY FlashCopy® return codes and reason codes specify the condition that occurred. Table 25 on page 79 lists the explanation for each code.

Table 25. AOM QUERY Flashcopy Return and Reason Codes

Return Code	Reason Code	Description
0 (X'00')		The control volume and all of the volumes in the VOLLIST have been processed. All of the volume entries have been marked to indicate their degree of FlashCopy eligibility and separation.
	0 (X'00')	All volumes are eligible.
	1 (X'01')	One or more known volumes in the volume list are ineligible.
4 (X'04')		Processing cannot be accomplished. None of the volumes' entries have been marked to indicate their degree of FlashCopy eligibility and separation.
	1 (X'01')	The control volume does not support any version of FlashCopy.
	2 (X'02')	Control volume support for FlashCopy cannot be determined. I/O to read flash copy support status for uninitialized device (offline) did not complete or has incomplete status.
8 (X'08')		QUERYFC detected an error in the input parameters or in the contents of the VOLLIST.
	1 (X'01')	AOM02039 was passed a SERVL in which SERVVPTR is zero.
	2 (X'02')	AOM02039 was passed a SERVL in which AOMQFC_LISTNUM is zero.
	3 (X'03')	The chain of pointers needed to locate the SSCT contains a zero pointer.
	4 (X'04')	A number for extent pairs to be checked was passed without passing the extent pairs.
	5 (X'05')	A number for extent pairs to be checked exceeded max relations allowed on volume.
10 (X'0A')		ASYN PPRC primary is currently active on volume.
12 (X'0C')		A serious AOMQFC error occurred, such as an ABEND.
	1 (X'01')	Load module AOM02039 could not be loaded by module IGX00039. SERVFRC will contain the failure return code from LOAD.
	2 (X'02')	Load module AOM02039 was passed a SERVL that specified a request type other than SERVFVLS(QUERYFC).
	3 (X'03')	Getmain failed in AOM02039. SERVFRC set to getmain return code.

Table 25. AOM QUERY Flashcopy Return and Reason Codes (continued)

Return Code	Reason Code	Description
	4 (X'04')	Freemain failed in AOM02039. SERVFRC set to freemain return code.
	5 (X'05')	Program error in AOM02039. SERVFRC set to system completion code.
	6 (X'06')	Getmain failed in AOM02139. SERVFRC set to getmain return code.
	7 (X'07')	Freemain failed in AOM02139. SERVFRC set to freemain return code.
	8 (X'08')	Program error in AOM02139. SERVFRC set to system completion code.
	9 (X'09')	Load module AOM02139 could not be loaded by module AOM02039. SERVFRC will contain the failure return code from LOAD.

Table 26 on page 80 lists reason codes for the following conditions:

- FlashCopy volume capability
- FlashCopy being ineligible for volume-related reasons
- FlashCopy extent capability
- FlashCopy being ineligible for extent-related reasons

Table 26. FlashCopy Volume Capability and Ineligibility, and Extent Capability and Ineligibility

Reason Code	Description
FlashCopy Volume Capability (AOMQFC_VCAP):	
0 (X'00')	Volume is not capable.
1 (X'01')	Volume is source capable (source query).
2 (X'02')	Volume is target capable (target query).
FlashCopy not eligible for volume-related reasons (AOMQFC_VRSN): Reason message text (AOMQFC_MSGTEXT):	
0 (X'00')	Volume is capable. Reason msg text: VOLUME IS TARGET CAPABLE Or Reason msg text: VOLUME IS SOURCE CAPABLE
1 (X'01')	Total relations active is at maximum for the volume. Reason msg text: MAX VOL FC RELATIONS
2 (X'02')	PPRC primary is active on volume Reason msg text: PPRC PRIM CURRENTLY ACTIVE
3 (X'03')	PPRC secondary is active on volume Reason msg text: PPRC SEC CURRENTLY ACTIVE
4 (X'04')	Concurrent copy source is active on volume. Reason msg text: CC SRC CURRENTLY ACTIVE
5 (X'05')	XRC source is active on volume. Reason msg text: XRC SRC CURRENTLY ACTIVE
6 (X'06')	Volume inhibited for FlashCopy operations by command. Reason msg text: VOLUME INHIBITED FOR FC

Table 26. FlashCopy Volume Capability and Ineligibility, and Extent Capability and Ineligibility (continued)

Reason Code	Description
7 (X'07')	Version 1 FlashCopy relation exists on the volume. Reason msg text: VERSION 1 FC RELATION EXISTS
8 (X'08')	Total relations active is at maximum of the ESS. Reason msg text: MAX ESS FC RELATIONS
9 (X'09')	Volume is inaccessible. Reason msg text: VOLUME INACCESSIBLE
10 (X'0A')	ASYN PPRC primary is currently active on volume. Reason msg text: ASYNC PPRC PRIM ACTIVE
11 (X'0B')	Volume already active as a full volume target (target capability). Reason msg text: FULL VOL FC TARGET RELATION
12 (X'0C')	Volume already active as a full volume source (target capability). Reason msg text: FULL VOL FC SOURCE RELATION
13 (X'0D')	Volume already active in the maximum full volume source relations. Reason msg text: MAX FULL VOL FC RELATIONS
200 (X'C8')	Volume not found. Reason msg text: VOLUME NOT FOUND
201 (X'C9')	No versions of Flash Copy are supported on this volume. Reason msg text: FLASHCOPY NOT SUPPORTED
202 (X'CA')	Volume exceeds current supported FC boundary of the control volume. Reason msg text: BOUNDARY EXCEPTION
203 (X'CB')	I/O error obtaining current FlashCopy status. Reason msg text: I/O ERROR ON QUERY
204 (X'CC')	Volume geometry not compatible with control volume. Reason msg text: VOL GEOMETRY NOT COMPATIBLE
205 (X'CD')	MVS-formatted VM minidisk. Reason msg text: MVS-FORMATTED VM MINIDISK
FlashCopy Extent Capability (AOMQFC_XCAP):	
0 (X'00')	Volume is not capable.
1 (X'01')	Volume is source capable (source query).
2 (X'02')	Volume is target capable (target query).
FlashCopy not eligible for extent-related reasons (AOMQFC_XRSN): Reason text (AOMQFC_MSGTEXT):	
0 (X'00')	Volume is capable on extent basis. Reason msg text: VOLUME IS TARGET CAPABLE or VOLUME IS SOURCE CAPABLE

Table 26. FlashCopy Volume Capability and Ineligibility, and Extent Capability and Ineligibility (continued)

Reason Code	Description
1 (X'01')	Already active as a source extent (target capability) Reason msg text: ALREADY ACTIVE AS SOURCE
2 (X'02')	Already active as a target extent (source or target capability) Reason msg text: ALREADY ACTIVE AS TARGET
3 (X'03')	Total relations active is at maximum for targets (source capability). Reason msg text: ALREADY IN MAX RELATIONS
4 (X'04')	Relationship entry is in a validation-required state (source or target capability). Reason msg text: RELATION IN VALIDATION
Preserve Mirror (PM) FlashCopy (FC) Volume Capability (AOMQFC_PM_VCAP):	
0 (X'00')	PM volume is not capable.
1 (X'01')	PM volume is capable.
2 (X'02')	PM capability was not requested.
3 (X'03')	PM capability was not checked because not FC capable.
PM FlashCopy not eligible for volume-related reasons (AOMQFC_PM_VRSN): Reason message text (AOMQFC_MSGTEXT):	
0 (X'00')	PM Volume is capable or not checked. Reason msg text: PRESERVE MIRROR CAPABLE (if AOMQFC_PM_VCAP=X'01') or ADDRESSED VOL NOT FC CAPABLE(if AOMQFC_PM_VCAP=X'03')
1 (X'01')	Total relations active is at maximum for the secondary volume. Reason msg text: MAX VOL RELATIONS FOR SCNDRY
4 (X'04')	Concurrent copy src active for the secondary. Reason msg text: SCNDRY CC SRC ACTIVE
5 (X'05')	XRC src active for the secondary. Reason msg text: SCNDRY XRC SRC ACTIVE
6 (X'06')	Secondary volume is inhibited for FlashCopy. Reason msg text: FC INHIBITED ON SCNDRY VOL
7 (X'07')	Phase 1 FC relation exists on the secondary volume. Reason msg text: VER 1 FC RELATION ON SCNDRY
8 (X'08')	Total relations currently at the maximum for the secondary ESS. Reason msg text: MAX FC RELATIONS FOR SCNDRY
9 (X'09')	Secondary volume is inaccessible. Capability information could not be obtained. Reason msg text: SCNDRY VOLUME INACCESSIBLE
10 (X'0A')	Secondary volume is a Global Mirror primary. Reason msg text: SCNDRY IS GL MIRROR PRIMARY
11 (X'0B')	Full volume target relation exists on the secondary volume. Reason msg text: MAX FC RELATIONS FOR SCNDRY

Table 26. FlashCopy Volume Capability and Ineligibility, and Extent Capability and Ineligibility (continued)

Reason Code	Description
12 (X'0C')	Full volume source relation exists on the secondary volume and target capability was requested. Reason msg text: FULL VOL SRC RELATION ON SEC
13 (X'0D')	The maximum number of full volume source relations exist on the secondary volume and source capability was requested. Reason msg text: MAX FULL VOL RELATION ON SEC
14 (X'0E')	Secondary voume is a Track Space Efficient volume and target capability was requested. Track Space Efficient volumes are not allowed for Preserve Mirror operations. Reason msg text: SEC VOL IS SPACE-EFFICIENT
15 (X'0F')	Secondary volume is in the process of PPRC Cascading Failback Enabled. Reason msg text: INCRES OR CASC FB STARTED
16 (X'10')	Secondary is a cascaded PPRC. Reason msg text: SEC IS CASCADED PPRC
128 (X'80')	Volume is not a PPRC primary device. Reason msg text: NOT PPRC PRIMARY DEVICE
129 (X'81')	The secondary to the addressed volume and the secondary to the control volume are not in the same SFI. Reason msg text: SEC VOLS NOT IN THE SAME SFI
130 (X'82')	PPRC relationship between the addressed volume and its secondary volume is not in the DUPLEX state. Reason msg text: PPRC PAIR NOT A FULL DUPLEX
131 (X'83')	The secondary control unit does not have PM feature installed. Reason msg text: SEC CTRL UNIT NOT PM ENABLED
132 (X'84')	The control volume is not a PPRC primary device. Reason msg text: CTRL VOL IS NOT PPRC PRIMARY
133 (X'85')	The command was received while system was attempting to install a new microcode load and it does not support PM feature. Reason msg text: PARTNER CEC INCAPABLE OF PM
134 (X'86')	Error was detected while the PRSDX47 was sent to the addressed secondary volume. Reason msg text: PRSDX47 IN ERROR ON SCNDRY
135 (X'87')	The addressed volume is Track Space Efficient volume and target capability was requested. Reason msg text: VOLUME IS SPACE-EFFICIENT
208 (X'D0')	PPRC Primary and PM feature not installed. Reason msg text: PPRC PRIM WITHOUT PM FEATURE
209 (X'D1')	PPRC Primary not duplex or RPFCv2 feature not enabled. Reason msg text: PRIMARY NOT DUPLEX, no RPFCv2
256 (X'100')	PM feature not installed in storage server. Reason msg text: PM FEATURE NOT INSTALLED

Table 26. FlashCopy Volume Capability and Ineligibility, and Extent Capability and Ineligibility (continued)

Reason Code	Description
Preserve Mirror (PM) volume reason code for extent capability (AOMQFC_PM_XCAP):	
0 (X'00')	PM volume is not capable.
1 (X'01')	PM source capable (on source query).
2 (X'02')	PM target capable (on target query).
PM FlashCopy not eligible due to extent-related reasons (AOMQFC_PM_XRSN): Reason message text (AOMQFC_MSGTEXT):	
0 (X'00')	Volume is capable on extent basis. Reason msg text: VOLUME IS REMOTE TGT CAPABLE or Reason msg text: VOLUME IS REMOTE SRC CAPABLE
1 (X'01')	Already active as a remote source extent (target capability) Reason msg text: ALREADY ACTIVE AS REMOTE SRC
2 (X'02')	Already active as a remote target extent (source or target capability) Reason msg text: ALREADY ACTIVE AS REMOTE TGT
3 (X'03')	Total relations active is at maximum for targets (source capability). Reason msg text: REMOTE IN MAX RELATIONS
4 (X'04')	Relationship entry is in a validation-required state (source or target capability). Reason msg text: REMOTE REL IN VALIDATION

Chapter 4. BAM diagnostic aids

The following major diagnostic aids are provided for the basic access method (BAM):

- Extended format I/O routine diagnostic information
- The format of the 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 346

  For fields that BAM adds to the IOB, see "IOB" on page 430.

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

DFSMSdss calls SDM, system data mover, to process data residing in sequential extended-format data sets. SDM calls the extended format I/O routine 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 27 on page 85 lists the extended format I/O routine return and reason codes.

Table 27. Extended Format I/O Routine

Return Code	Reason Code	Description
0 (X'00')	0 (X'00')	The requested operation was performed successfully.

Table 27. 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 that 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 with 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.
	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.

Table 27. Extended Format I/O Routine (continued)

Return Code	Reason Code	Description
	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.
	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.

Finding open data sets

You can find information about open data sets for the failing task by examining the section named DATA MANAGEMENT CONTROL BLOCKS in a dump. This section appears in a SYSABEND or SYSUDUMP dump if the DM system option is in effect (which is typical). With a SYSMDUMP or standalone dump you can use IPCS option 2.4 to see this section.

Each DEB represents a data set that the current task opened. The header for each DEB identifies its DD name and data set name. Later lines identify the access method and record format and help you to find the buffers.

Finding buffers

You can find the buffers that are associated with an open data set. For QSAM, these buffers are in the buffer pool that OPEN normally builds. For BSAM and BPAM, these buffers are the data areas that the READ or WRITE macro passed. In both cases, you can find them by examining the section named `DATA MANAGEMENT CONTROL BLOCKS` in a dump.

Each DEB represents an open data set. Find the DEB for the open data set that you are interested in. The header line for each DEB identifies the DD name and the data set name. Then, look at the control blocks that are labeled `IOB` after the DEB and DCB. You should see between 1 and 30 IOBs. If you see ICBs, then use the ICBs instead.

The dump shows IOBs or ICBs with numbers. The numbers begin with the IOB or ICB that has the lowest storage address, but the IOBs or ICBs are printed in the order in which the access method is using them. There might be up to 255 IOBs or ICBs, but only the most recently used 30 are printed. Generally the last one is the current one, but if the failure occurred in an access method module, you might be interested in the first IOB or ICB.

Finding QSAM buffers

At the end of each IOB or ICB are some unformatted hexadecimal words. They represent a simulated format 0 CCW. Examine the first byte of the second word. It is a flag byte. The sixth bit (X’04’) is called the IDA bit. If the IDA bit is on, then the low order 3 bytes of the previous word point to a word that points to the buffer. In this case, the buffer is above the line. If the IDA bit is off, then the low order 3 bytes of the previous word point to the buffer.

Finding BSAM or BPAM buffers

After the IOB in a formatted dump, you can see a DECB. Each represents a READ or WRITE macro. If you do not see a DECB, it might be because the application program has not yet issued enough READ or WRITE macros to use the IOB or ICB. Another possibility is that the application program freed the DECB storage. The word labeled AREA points to the data area for the READ or WRITE macro. If the programmer coded SF64 on the READ or WRITE macro, then the word points to a doubleword that points to the data area.

Finding the buffer that got an I/O error

At the right end of each IOB or ICB header line, there are some abbreviations for the status of the I/O for that buffer. This status might come from the device, or be simulated by the access method. The normal status is `CE DE`, which means channel end and device end. You might also see UE, unit exception, which means that you have read to the end of the data set or you are writing at the end of a magnetic tape. The access method often marks UE also for later IOBs or ICBs.

These are the meanings of the other codes that you might see:

ATTN
attention

BUSY
busy

CC
chaining check

CCC
channel-control check

CDC
channel-data check

CUE
control unit end

ICC
interface-control check

IL
incorrect length

PCI
program-controlled interrupt

PROG
program check

PROT
protection check

STAT-MOD
status modifier

UC
unit check

The most common I/O error is UC, unit check. For a unit check, you should examine the 2 bytes that are labeled SENS at offset 2. They are real or emulated device sense bytes. The first 6 bits have these meanings:

Command reject
Intervention required
Bus-out check
Equipment check
Data check
Overrun

Meanings of the other bits are device-dependent.

Additional SYNAD information (BSAM and QSAM)

For a PDSE, extended format data set, or UNIX 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 that can contain a diagnostic code stored by BAM 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 28 on page 90 is a list of diagnostic codes.

Table 28. BAM 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 by using POINT (PDSE).
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). The application programmer can solve this by coding BLOCKTOKENSIZE=LARGE on the DCBE macro. This will require the programmer to examine all issuances of the NOTE and POINT macros for this DCB.
15 (X'0F')	An error was encountered while processing a UNIX 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.

BAM internal trace facility

The BAM internal trace facility provides an ongoing record of internal software events occurring during BSAM, BPAM or QSAM I/O operations in chronological order. When you activate the BAM internal tracing, BAM starts it during the data set's DCB OPEN processing and it runs continuously until the DCB is closed. The BAM internal trace facility creates records that are placed into the data set's trace table. This data set's DCB trace table resides in the private area of storage within the address space.

Invoking the BAM internal trace facility

To enable the BAM internal trace facility for a data set that will be accessed via BSAM, BPAM or QSAM, use any of the methods to start the OPEN/CLOSE/EOV work area trace. If you start GTF with appropriate options, you will get an OPEN/CLOSE/EOV work area trace. The BAM internal trace does not require GTF.

To start BAM internal tracing, use one of these methods:

- Specify DCB=DIAGNS=TRACE on the DD statement of a data set for which you wish to trace
- Specify the DALDIAGN text unit with dynamic allocation for a data set for which you wish to trace
- Use the problem determination optional work area trace started task, IFGOCETR. Refer to "Non-VSAM problem determination work area trace started task" on page 353 for information on the started task.

The BAM internal trace facility then operates on the data set or data sets.

Initialization and termination of the BAM internal trace facility

BAM internal tracing occurs for the duration of the data set's access by the application program, from OPEN until CLOSE. Tracing starts when the BAM OPEN executor determines that you requested OPEN/CLOSE/EOV work area tracing. You do not have to start GTF. BAM creates internal trace records until the data set's DCB is closed. The BAM internal trace table is destroyed (with FREEMAIN) during DCB CLOSE processing.

BAM internal trace records

The trace record types that may be created when the BAM internal trace facility is active are:

- ENTRY records that describe an entry into a module (CSECT)
- EXIT records that describe an exit from a module (CSECT)
- Numbered records, from 1 to 8, that describe optional trace records inserted by IBM service to gather additional diagnostic information.

The content of the trace records is as follows:

- All trace records contain the timestamp to indicate the record's time of creation.
- All trace records running under a TCB contain the TCB address, extracted from PSATOLD.
- All trace records running under an SRB contain zeros instead of a TCB address. This value is extracted from PSATOLD.
- All trace records contain the CSECT name, for example, IGG019V8, IGG019V9, and so on.
- All trace records contain the type identifier, for example, ENTRY or EXIT.
- All trace type ENTRY records contain the caller's return address. Other trace types may contain optional data.

The following example shows EXIT and ENTRY type trace records:

```
TRACE:      TIME     TCB      RET/DATA   CSECT    TYPE
17:34:54.882497  009E68C8  00009F28  IGG019BB  EXIT
17:34:54.880990  009E68C8  00009F28  IGG019BA  EXIT
17:34:54.880990  009E68C8  50020D42  IGG019BB  ENTRY
```

Displaying the BAM internal trace table

If BAM internal tracing is in effect and a dump occurs, the BAM internal trace table is displayed in the formatted output area along with the data set's DEB and DCB information. You may see it in any of these types of dumps:

- SYSUDUMP dump, when SDATA=DM is in effect in the IEADMP00 member of SYS1.PARMLIB. This is the normal case.
- SYSABEND dump, when SDATA=DM is in effect in the IEAABD00 member of SYS1.PARMLIB. This is the normal case.
- SNAP dump, when an application program issued a SNAP macro and specified SDATA=DM.
- System dump or stand-alone dump, when the user uses IPCS and specifies one of the following:
  - Option 2.4 (**Analysis** and **Summary**), with report type F for **Format**
  - An IPCS TCBEXIT IECDAFMT *xxxxxx* command, where *xxxxxx* is a TCB address that ends with a period

- An IPCS CBFORMAT *xxxxxx* command, where *xxxxxx* is a TCB address that ends with a period.

You can cause a dump with the operator SLIP or CANCEL command while the data set is open.

Interrupt control block (ICB)

The SAM ICB, used by BAM, 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 29. 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 29. 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.

Chapter 5. BDAM diagnostic aids

This topic 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 30 on page 95 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 *z/OS MVS System Messages, Vol 7 (IEB-IEE)*.

Table 30. 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	

Chapter 6. Compression management services diagnostic aids

This topic 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.

CMP002E
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.

CMP003E
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 CMP003E message with the return and reason codes for the error.

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 blocks that are available at the time of record
- The module that actually performed the record (CMPSVCSY)

Return code format

As Table 31 on page 98 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 = 000000xx

In the example, xx is the return code in hexadecimal format.

Table 31. 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 32 on page 99 for a list of module identifiers.

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

See Table 35 on page 100 for a complete list of CMP reason codes.

Table 32. 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 33. 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 35 on page 100 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 34. 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 35 on page 100 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 35. 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.
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		

Table 35. CMP Return and Reason Codes (continued)

Return Code	Reason Code	Explanation
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 level of MVS that supports compression is installed on this system.
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 CMP003E 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 35. 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. To be eligible for compression, a data set must: • Meet the requirements for Extended Format • Have a primary allocation of at least 5 megabytes in order to be allocated as a compressed data set, or 8 MBs if no secondary allocation is specified • Have a minimum record length of forty (40) bytes, not including the key offset or key length • Be SMS managed. The mechanism for requesting compression for VSAM data sets is through the SMS data class COMPACTION=Y parameter. **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. This also true for the 8 MB primary allocation, if no secondary is specified. • The maximum record length for nonspanned data sets is 3 bytes less than the maximum record length of data sets that do not contain compressed data (this length is CISIZE-10). • The relative byte address (RBA) of another record, or the address of the next record in a buffer, cannot be determined using the length of the current record or the length of the record provided to VSAM. • The length of the stored record can change when updating a record without any length change. • The key and any data in front of the key will not be compressed. Data sets with large key lengths and RKP data lengths might not be good candidates for compression. • Only the data component of the base cluster is eligible for compression. Alternate indexes are not eligible for compression. • The global shared resources (GSR) option is not permitted for compressed data sets.

Table 35. CMP Return and Reason Codes (continued)

Return Code	Reason Code	Explanation
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.

RC_CALR X'10' Reason Codes

Return Code	Reason Code	Explanation
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.

Table 35. CMP Return and Reason Codes (continued)

Return Code	Reason Code	Explanation
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.
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.

Table 35. CMP Return and Reason Codes (continued)

Return Code	Reason Code	Explanation
X'24'	DEC 1096	RS_GET_TOKEN_FAIL MPSTCGI's get of CMS's Global token failed.
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.
X'24'	DEC 4099	RS_BAD_DICTIONARY The dictionary has some format error. The MVS Compression Services cannot operate properly. The MVS return code will be propagated to the caller in the low order 2 bytes of the reason code.

Table 35. CMP Return and Reason Codes (continued)

Return Code	Reason Code	Explanation
X'24'	DEC 4100	RS_BAD_DICT_TOKEN Invalid Dictionary Token. The Dictionary Building Blocks 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 with 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.
X'24'	DEC 4111	RS_BAD_AS_TOKEN The CMSAS token retrieved by IGWFTOKM does not point to a valid CMSAS block.
X'24'	DEC 4112	RS_BAD_TASK_TOKEN The CMSTASK token retrieved by 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.

Table 35. CMP Return and Reason Codes (continued)

Return Code	Reason Code	Explanation
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 with 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 7. DADSM/CVAF diagnostic aids

This topic provides the following major diagnostic aids for DADSM/CVAF:

- CVAF module-to-function cross-reference
- CVAF component trace
- 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 36 on page 109, 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 "DADSM/CVAF—modifier keywords" on page 46.

Table 36. 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 component trace

The CVAF component traces the following events:

- CVAF entry and exit
- 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

CVAF captures information in ctrace format in the Device Manager address space (devman). The trace data can optionally be connected to a ctrace output writer using the component name SYSDMO. The devman buffer containing CVAF trace data is included in any system dumps that are taken by the CVAF recovery routines. In addition, the system operator can request that devman take a dump that includes the CVAF trace data. CVAF trace data provides diagnostic information that improves the first failure data capture of CVAF errors.

CVAF recording of trace events must be requested. It is not the system default. By default, no trace data is recorded by CVAF. When requesting this trace data, you can specify up to 16 job names and up to 16 ASIDs. If either the job name or ASID matches, the record will be traced. Not specifying the job name or ASID results in all records being traced. To capture CVAF trace data, enable tracing as follows:

```
TRACE CT,ON,COMP=SYSDMO
*07 ITT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND
R 7,OPTIONS=(CVAF1,DADSM1),END
```

Request all component trace records for specific job names and ASIDs as follows:

```
TRACE CT,ON,COMP=SYSDMO
*02 ITT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND.
R 2,OPTIONS=(CVAF1,DADSM1,DADSM2)
*03 ITT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND.
R 3,JOBNAME=(JOBA,JOBB,JOBC)
*04 ITT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND.
R 4,ASID=(1001,1002,1003),END
```

Note: To capture DADSM OSVTOC recorded records, you must add DADSM2 to the list of OPTIONS parameters.

When tracing is no longer required, it can be disabled with the following commands:

```
TRACE CT,ON,COMP=SYSDMO
*07 ITT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND
R 7,OPTIONS=(CVAF0,DADSM0),END
```

Use the following command to request a devman dump that includes the CVAF ctrace data:

```
F  DEVMAN,DUMP=DUMP1
```

To display the CVAF ctrace records use the following IPCS command:

```
CTRACE COMP(SYSDMO)  FULL  LOCAL
```

For additional information on the use of IPCS, see *z/OS MVS Diagnosis: Tools and Service Aids*.

CVAF Ctrace record format

When displayed by IPCS, CVAF ctrace records have the following general format:

```
SYSTEM1   TRDATA    40000001  10:33:16.467722  WRITE DSCB TO VTOC

   +0000  MODULE... ICVDSD01  LOCATION. WRITE DSCB TO VTOC
   +0010  SEQNO.... 00000000  CPUID.... 0041      ASID..... 0015
   +001C  TCB...... 009FD098  DATA@.... 7F03387C  DATASIZE. 0064
   +004C  USER..... (TRDATA)
   +006C  CVAFDSN..
   +0098  CVAFVOL.. USRPAK    CVAFCCHR. 00000000  00
   +00A3  CVAFRBA.. 00000000

SYSTEM1   TRDATA    00001801  10:33:16.467724

   +0000  MODULE... ICVDSD01  LOCATION. DUMP OF DSCB
   +0010  SEQNO.... 00000001  CPUID.... 0041      ASID..... 0015
   +001C  TCB...... 009FD098  DATA@.... 7F0117A0  DATASIZE. 0060
   +004C  USER..... (DSCB)
   +006C  F1E4E2D9  D7C1D200  015F0128  63016D01  | 1USRPAK..¬...._. |
   +007C  0000C9C2  D4D6E2E5  E2F24040  40404068  | ..IBMOSVS2     . |
   +008C  01400000  00000008  C0801000  00000000  | . ......{....... |
   +009C  0012C000  00000000  00E5A200  00810000  | ..{......Vs..a.. |
   +00AC  40000000  43000E00  00000000  00000000  |  ............... |
   +00BC  00000000  00000000  00000000  00000000  | ................ |
SYSTEM1   SCRATCH   40000020  10:29:33.904427  DADSM MODULE ENTRY   SCRATCH

   +0000  MODULE... IGGDAS02  LOCATION. SCRATCH:PARAMETER LIST
   +0010  SEQNO.... 0000004A  CPUID.... 0081      ASID..... 002F
   +001C  TCB...... 009CE188  DATA@.... 009A92E8  DATASIZE. 0058
   +0024  JOBNAME.. IBMUSER   CALLEDBY. 83B6DDEA
   +004C  USER.....        (IGGDASCR)
   +006C  DASPLID.. IGGDASCR  DASPVER.. 0000      DASPLEN.. 002C
   +0078  DASPKEY.. 50        DASHRTC.. 0000      DASDIAGI. 00050000
   +0081  DASFLAG2. 000000    DASFLAG6. 8080      DASUCB... 00000000
   +008C  DASAVOLL. 009A8A84  DASADSN.. 009A8B10  DASREFDT. 000000
   +0098  DASDSN... IBMUSER.X.Y.Z
```

Header information that is common to all trace records identifies the module that created the trace (in the field named MODULE). The header also contains a descriptive name of the trace (in the field named LOCATION). The field named USER identifies the data that is being traced.

The example above shows three ctrace records recorded.

- The first trace record format is created for all CVAF trace records and records data named TRDATA that includes DSNAME, VOLSER, CCHHR and RBA. Not all of the TRDATA is relevant to every trace record. In this example, only the VOLSER field contains valid data. Depending on the type of trace, a second record might sometimes be recorded in order to trace additional data.
- The second record traces the dataset control block (DSCB).
- The third record trace record shows the callers address in field CALLEDBY. A system programmer can use the callers address and from an IPCS dump find the program offset and name. For example, the following shows an example of using the caller's address, 83B6DDEA, from the trace record to find information about the calling program:

  ```
  IP WHERE 3B6DDEA===>  ASID(X'000D') 03B6DDEA. IGDZILLA+11BDEA IN EXTENDED PLPA

  03B6CF50   00000000   00000000   47F0F026   20C9C7C4   | .........00..IGD |
  03B6CF60   E5E3E2C4   C1F0F561   F2F761F0   F8C8C4E9   | VTSDA05/27/08HDZ |
  03B6CF70   F1C2F1F0   40404040   40404040   400090EC   | 1B10         ... |
  ```

CVAF Ctrace events

The following CVAF events are traced. The heading names such as 'Write DSCB to VTOC' are the names used by IPCS to describe each record.

- Write DSCB to VTOC

  Recorded when CVAF writes a DSCB to the VTOC. Trace data includes the VOLSER and DSCB. Recorded by module ICVDSD01 and ICVDSD02. Restriction: CVAF does not write format-0 and format-4 DSCBs. Therefore, no trace record is created for them.
- Write VIR to VTOC index

  Recorded when a VTOC index record (VIR) is written to the VTOC index. Trace data includes the VOLSER and Dataset Name and 1 or more VTOC information records (VIRs). Recorded by module ICVCMI00
- Add Index

  Recorded when a data set name and a CCHHR pointer are added to the VTOC index. Trace data includes the VOLSER and DSNAME. Recorded by module ICVIXA00
- Delete Index

  Recorded when a data set name is deleted from the VTOC index. Trace data includes the VOLSER and DSNAME. Recorded by module ICVIXD00
- Allocate Space from VPSM

  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. Trace data includes the VOLSER and the Extent Descriptor Table (recorded in a second trace record). Recorded by module ICVDMS01

- Release Space from VPSM

  Recorded when space on the volume is released from the volume pack space map (VPSM). Trace data includes the VOLSER and the Extent Descriptor Table (recorded in a second trace record). Recorded by module ICVDMS01
- Allocate VTOC Index Space

  Recorded when a VTOC index record is allocated and marked in the VTOC index map (VIXM) as available for allocation. Trace data includes the VOLSER and CCHHR. Recorded by module ICVDMS02
- Release VTOC Index Space

  Recorded when a VTOC index record is released and marked in the VIXM as unavailable for subsequent allocation. Trace data includes the VOLSER and CCHHR. Recorded by module ICVDMS02
- 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. Trace data includes the VOLSER and CCHHR. Recorded by module ICVDMS02
- Release VTOC Space

  Recorded when a DSCB (all formats) is released and marked in the VMDS as unallocated. Trace data includes the VOLSER and CCHHR. Recorded by module ICVDMS02
- CVAF Entry

  Recorded on entry to CVAF module ICVCMM00. Trace data includes the CVAF parameter list (CVPL). Recorded by module ICVMM00.
- CVAF Exit

  Recorded on exit from CVAF module ICVCMM00. Trace data includes the CVAF parameter list (CVPL). Recorded by module ICVCMM00.
- Read or Write multiple DSCBs

  Recorded when CVAF reads or writes multiple DSCBs to the VTOC. Trace data includes the VOLSER and DSCB and 1 or more DSCB processed by this CVAF request. Recorded by module ICVDSD02.

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 ENDT, OR DBUG
   ```

3. Enter command DBUG.
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 113. 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 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 *workarea* 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 118 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 *workarea* 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 118 for a description of these codes.

The dump title line is in the following format. See Figure 19 on page 118 for a sample title line.

```
MODNAME  JOBNAME  CVAF ERROR TYPE n   TCB=tcbaddr  W/A=workarea
```

where:

MODNAME is the CVAF module involved
n is the error type (1, 2, or 3)
tcbaddr is the TCB address
workarea is the work area address

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

Figure 19. 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 Table 11 on page 49. For a description of the CVSTAT codes, see "CVSTAT field codes" on page 118 .

CVSTAT field codes

The CVSTAT field codes are as follows.

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 or read-only volumes.

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 or HADSCB keyword omitted.

15 (X'0F')
UCB is not valid, volume was not mounted; VIO unit, not DASD.

16 (X'10')
The device has no paths and NOQUEUE=OK was specified.

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 RTA is invalid since the description of this macro refers to RTA values and not RBN values.

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.

81 (X'51')
HADSCB=*addr* requested, but could not be determined. VIB not initialized or could not be determined from the VMDS.

82 (X'52')
CVAF operation issued to a volume with more than 65,520 cylinders and caller did not indicate support for it with EADSCB=OK. Applies to the following calls:

- CVAFDIR, where the DSCB associated with the seek or search argument is a format-8 or format-9 DSCB
- CVAFDSM, for a request for unallocated space
- CVAFSEQ, where physical sequential access is requested or where index order is requested and a DSCB read is described by a format-8 or format-9 DSCB

83 (X'53')
CVAFDIR or CVAFSEQ read a format-8 DSCB from a non-EAV

84 (X'54')
CVAFDIR or CVAFSEQ read a format-8 DSCB, yet the data set was not extended addressing space (EAS) eligible.

85 (X'55')
Same as error code 82 but request issued to any volume with DIAG*xx* ICVTESTEADSCB trap set and EADSCB=OK not specified. Applies to the following calls:

- CVAFDIR, where the DSCB associated with the seek or search argument is an extended addressing space (EAS) eligible data set
- CVAFDSM, for a request for unallocated space
- CVAFSEQ, where physical sequential access is requested or where index order is requested and a DSCB read is an extended addressing space (EAS) eligible data set

86 (X'56')
CVAFFILT request, all FCL entries were processed, one or more DSCB access errors were found (for example, format-8 DSCB but the caller did not specify support for EAV, or format 1 DSCB for an EAV eligible data set and the ICVTESTEADSCB DIAGxx trap is in effect.

87 (X'57')
CVAFDIR or CVAFSEQ read a format-8 DSCB that did not point to a format-9 DSCB, or a format-8 DSCB was read that had a forward pointer of zeroes.

88 (X'58')
Request to write or modify an essential field in a DSCB using CVAFDIR ACCESS=WRITE and VALIDATE=YES. No write occurred.

89 (X'59')
CVAF ACCESS=WRITE is not permitted on a read-only volume.

90 (X'5A')
CVAF parameter list length is incorrect.

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').

167 (X'A7')
Request to write a format-1 DSCB. The extents are not valid. A beginning extent was larger than an ending extent.

168 (X'A8')
Request to write a format-4 DSCB. One of the following field or fields were invalid. The number of logical cylinders on the volume (DS4DSCYL); number of tracks in a logical cylinder (DS4DSTRK); device track length (DS4DEVTK)

169 (X'A9')
Request to write a VTOC Index record (VIR). A VIR did not have a valid ID on the front of the record. The ID should be VIXM, VPMS,VMDS, OR VIER.

170 (X'AA')
Request to write the first High Level VIER. One or more of the fields in the VIER was invalid.

171 (X'AB')
Invalid write attempted for format-5 DSCB - 140 byte DSCB

172 (X'AC')
Invalid write attempted for format-5 DSCB - 140 byte DSCB

173 (X'AD')
Invalid write attempted for format-5 DSCB - 96 byte DSCB

174 (X'AE')
Invalid write attempted for format-8 DSCB

175 (X'AF')
Invalid write attempted for format-3 DSCB

176 (X'B0')
Invalid write attempted for format-9 DSCB

177 (X'B1')
Invalid write attempted for DSCB

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

Table 37 on page 126 lists the function and subfunction keywords for DADSM and CVAF functions. In this table, 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 "DADSM/CVAF—modifier keywords" on page 46.

Table 37. 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	System provided dynamic exit routine associated with the IGGPOST0_EXIT dynamic exit.
IGGPRE00	System provided dynamic exit routine associated with the IGGPRE00_EXIT dynamic 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

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 38 on page 127 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 39 on page 129 shows.

Tip: For certain errors during DADSM Extend processing for secondary allocations on a new volume, the diagnostic information in Table 39 on page 129 describes the DADSM Extend failure.

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

Table 38 on page 127 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 38 on page 127.

The field S99INFO contains the right-most 2 bytes of the DADSM diagnostic code, which are in the DADSM diagnostic information table in Table 39 on page 129.

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 38. 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 39 on page 129 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.

Table 38. DADSM CREATE Return Codes (continued)

DADSM SVC Return Code (R15)	Description
20 (X'14')	Requested quantity not available.
24 (X'18')	Average record length exceeds 65535 bytes.
28 (X'1C')	ISAM is no longer supported and the request has been failed.
40 (X'28')	The CREATE request specified 'exact-required (DACEXREQ) amount', but the exact amount of space could not be returned. This request came either from the caller or internally from CREATE. If the request came from the caller, DACEXREQ will be on in the caller's DACRE. Internally, CREATE applies a request as an 'exact-required (DACEXREQ)' allocation to divide the amount into track-managed space (TMS) and cylinder-manager space (CMS). This may occur because the amount requested was more than 65,520 tracks, up to a data set limit of 65,535 tracks. The exact amount of space could not be returned from both TMS and CMS because either the TMS or CMS was full.
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.
69 (X'45')	UCB is read-only secondary device.
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.
104 (X'68')	Invalid space subparameter.
108 (X'6C')	Zero primary space request for ABSTR request.
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.
136 (X'88')	Invalid FMT3 DSCB pointer.
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.
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.

Table 38. DADSM CREATE Return Codes (continued)

DADSM SVC Return Code (R15)	Description
196 (X'C4')	Requested data set creation was not possible. Register 0 contains additional diagnostic information. See Table 39 on page 129 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 or 16777215 tracks. The specific error is qualified by the DADSM create diagnostic code, see Table 39 on page 129.
217 (X'D9')	Data set could not be created, DSNTYPE=LARGE not valid for this data set.
220 (X'DC')	VTOC conversion failed because the VTOC is full.

Table 39 on page 129 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 39 on page 129 are defined in Table 40 on page 142.

Table 39. 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'06'	X'00'	X'57'	Non-SMS managed extended-format was not allowed to allocate.
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'	ISAM data sets are no longer supported.
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'09'	X'00'	X'B0'	Creation of a PDS dataset with zero directory blocks was not allowed.
X'04'	X'0A'	X'00'	X'34'	Validate model DSCB extension; unmovable data sets not on SMS-managed volumes.

Table 39. 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'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.
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'12'	X'00'	X'74'	An Exact Allocation preferred is specified, but the amount of storage returned is not equal to the amount requested

Table 39. 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'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'39'	Validate CREATE request; data set name is VTOC name.
X'04'	X'14' or X'3E'	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.
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' or X'3E'	X'00'	X'3E'	Validate DADSM CREATE request; ADSCB request—directory quantity specified, but data set is not a PDS.
X'04'	X'14' or X'3E'	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' or X'3E'	X'00'	X'40'	Validate CREATE request; MDSCB or ADSCB address invalid.
X'04'	X'14' or X'3E'	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 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' or X'3E'	X'00'	X'45'	Validate DADSM CREATE request; ADSCB request—DSORG cannot be PDS.
X'04'	X'14' or X'3E'	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' or X'3E'	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' or X'3E'	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' or X'3E'	X'00'	X'49'	Validate DADSM CREATE request; ADSCB request—primary quantity is zero.

Table 39. 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'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' or X'3E'	X'00'	X'4D'	Validate DADSM CREATE request; ADSCB request—invalid FMT3 key.
X'04'	X'14' or X'3E'	X'00'	X'4E'	Validate DADSM CREATE request; ADSCB request—invalid FMT3 ID.
X'04'	X'14' or X'3E'	X'00'	X'4F'	Validate DADSM CREATE request; ADSCB request—number of extents is greater than 16.
X'04'	X'14' or X'3E'	X'00'	X'50'	Validate DADSM CREATE request; ADSCB request—number of extents is less than 4.
X'04'	X'14' or X'3E'	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'14'	X'00'	X'AA'	Data set could not be created, DSNTYPE=LARGE not valid for this data set.
X'04'	X'15'	PREREA1	PREREA2	DADSM CREATE exit routine associated with the IGGPRE00_EXIT preprocessing dynamic exit, such as IGGPRE00, 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').

Table 39. 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'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.
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.
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'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.

Table 39. 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'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'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.
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'	CVRET	CVSTAT	An Exact Allocation required is specified and the exact amount of storage could not be obtained.
X'04'	X'33'	X'00'	X'72'	The active SMS configuration parmlib UseEav(NO) is specified for a request on an EAV.

Table 39. 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'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'3B'	X'00'	X'F2'	VTOC Conversion; convert failed, no FMT0 DSCBs to use for FMT5 or FMT7 DSCBs. VTOC full.
X'04'	X'3D'	X'00'	X'00'	VTOC conversion; IGG0325P requested FMT5 space update.
X'04'	X'3E'	X'00'	X'70'	The caller specifies an F8 ID request for a data set with an ineligible data set organization on EAV.
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 501.
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 501.
X'04'	X'4E'	X'00'	X'8E'	Requested primary space amount exceeds 16777215 tracks for DSNTYPE=LARGE.
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.
X'04'	X'50'	COMMRC	COMMRS	System error. A DADSM Common service detected an error during free-space search.
X'04'	X'51'	X'00'	X'AB'	DADSM was unable to accept a non-VSAM ALLOC=ABS request with F1 ID and the extent pairs specified in EAV cylinder-managed space.
X'04'	X'51'	X'00'	X'AC'	DADSM was unable to accept a VSAM ALLOC=ABS request with F8 ID and the extent pairs specified straddling the EAV track managed-space and cylinder-managed space.
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.

Table 39. 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'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.
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.

Table 39. 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'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.
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.

Table 39. 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'	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'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'42'	X'00'	X'5D'	DADSM was unable to find FMT0s on OSVTOC to write F9 DSCBs.
X'08'	X'44'	ECBCC, SNS0	CHSTAT, SNS1	VTOC conversion; read FMT5 DSCB.
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.

Table 39. 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'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.
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.

Table 39. 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'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.
X'0C'	X'06'	X'00'	X'91'	Validate branch entry parameter list; DACRSRV6 must be zero.

Table 39. 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'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'B1'	Create was attempted for a data set on a read-only volume.
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'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'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 extended format data sets.
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.

Table 39. 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'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 40 on page 142 describes abbreviations for DADSM CREATE diagnostic information.

Table 40. Abbreviation Descriptions for DADSM CREATE Diagnostic Information (Register 0)

Abbreviation	Description
CDMRET	Return code from CDM. For explanation see “CDM reason codes” on page 501.
CDMRSN	Reason code from CDM. For explanation see “CDM reason codes” on page 501.
CDMRSN1,2	CDM reason code (low order 2 bytes). For explanation see “CDM reason codes” on page 501.

Table 40. Abbreviation Descriptions for DADSM CREATE Diagnostic Information (Register 0) (continued)

Abbreviation	Description
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 68 on page 182.
COMMRS	DADSM common reason code. For a description of that reason code, see Table 68 on page 182.
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 118.
DCSRET	See return codes in "DASD calculation services return, reason, and status codes" on page 190.
DCSRSN	See reason codes in "DASD calculation services return, reason, and status codes" on page 190.
DCSSTAT	See status codes in "DASD calculation services return, reason, and status codes" on page 190.
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	Reason code (low order 2 bytes) from the IGGPRE00_EXIT preprocessing dynamic exit. 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 526.

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 127 and Table 39 on page 129 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 41 on page 144, the Extend diagnostic code (Extend to same volume only) might appear in an IEC614I message. The abbreviations that are used in Table 41 on page 144 are defined in Table 42 on page 147.

Table 41. 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'00'	X'5D'	X'00'	X'27'	Extend on an EAV for a preferred exact request (DAEXAPRE). The requested space amount was allocated but rounded up by CVAF to a multicylinder unit multiple in order to satisfy the request. The return code in register 15 is set to zero. The reason code in register zero set to X'005D0027'.
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. The data set exceeds the size of volume or, for a non-EAS eligible data set, the size exceeds the track-managed space.
X'04'	X'21'	X'00'	X'11'	Validate space request size. Request size would cause the data set size to exceed X'FFFF' tracks or, for DSNTYPE=LARGE data sets, cause the data set size to exceed X'FFFFFF' tracks.
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 41. 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'32'	X'00'	X'25'	OSVTOC path. No space available for contiguous request.
X'04'	X'39'	X'00'	X'00'	VTOC conversion; build FMT5 DSCBs.
X'04'	X'3B'	X'00'	X'F2'	VTOC Conversion; convert failed, no FMT0 DSCBs to use for FMT5 or FMT7 DSCBs. VTOC full.
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'56'	X'04'	X'1D'	IXVTOC path - Invoke CVAFDSM to get F0 DSCB for FN DSCB. VTOC is full.
X'04'	X'57'	X'04'	X'01'	OSVTOC path - Invoke CVAFDIR to get F0 DSCB for FN DSCB. VTOC is full.
X'04'	X'58'	X'04'	X'1D'	IXVTOC path - Invoke CVAFDSM to get F0 DSCB for F3 DSCB. VTOC is full.
X'04'	X'59'	X'04'	X'01'	OSVTOC path - Invoke CVAFDIR to get F0 DSCB for F3 DSCB. VTOC is full.
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.

Table 41. 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'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.
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'4A'	ECBCC	CHSTAT	OSVTOC path - Invoke EXCP to read F0 DSCB.
X'08'	X'4D'	CVRET	CVSTAT	Write a format-9 DSCB during the conversion to extended address DSCBs (format 8, 9) for an extended address space (EAS) eligible data set.
X'08'	X'4E'	CVRET	CVSTAT	Write a format-8 DSCB during the conversion to extended address DSCBs (format 8, 9) for an extended address space (EAS) eligible data set.
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.

Table 41. 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'52'	CVRET	CVSTAT	IXVTOC path - Invoke CVAFDIR to zero VRF data.
X'08'	X'53'	CVRET	CVSTAT	Invoke CVAFDIR using the Multiple DSCB write facility to write out updated DSCBs. These could be format 1,8, 9 or 3 DSCBs.
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'60'	X'00'	X'30'	OSVTOC invalid selected extent found updating the free space chain.
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.
X'0C'	X'0A'	X'00'	X'0A'	Extend was attempted for a data set on a read-only volume.

Table 42 on page 147 describes abbreviations for DADSM EXTEND diagnostic information.

Table 42. 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 68 on page 182.
COMMRS	DADSM Common Function reason code. For a description of that reason code, see Table 68 on page 182.
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 118.
ECBCC	ECB completion code. For a description of the code, see *z/OS DFSMS Using Data Sets*.

Table 42. Abbreviation Descriptions for DADSM EXTEND Diagnostic Information (continued)

Abbreviation	Description
PREREA1,2	Reason code (low order 2 bytes) from an exit routine associated with the IGGPRE00_EXIT preprocessing dynamic exit, such as IGGPRE00. 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 43 on page 148, Table 44 on page 148, and Table 45 on page 149 describe DADSM extend return codes.

End-of-Volume

Table 43. 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 44. 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.

Table 44. DADSM Catalog Management Return Codes (continued)

Return Code (R15)	Description
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 45. 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 LSPACE return and diagnostic codes

Table 46 on page 150 describes the DADSM LSPACE return codes.

Table 46. 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. Parameter list error – Bad parameter list storage key, or parameter list ID is invalid (not set to ‘LSPA’) , or parameter list size is not sufficient.
20 (X'14')	ESTAE routine entered, processing error in LSPACE.

Table 47 on page 150 describes additional diagnostic information that is returned by LSPACE in register 0. The abbreviations that are used in Table 47 on page 150 are defined in Table 48 on page 152. If the return code is not 16, it is also returned in the LSPACE parameter list at the locations that are identified in Table 49 on page 152.

Table 47. 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'08'	X'00'	X'0D'	No ECBs on the ECBLIST were found to be posted. This is an internal error where neither the ISGENQ OBTAIN ECB or the STIMERM timeout ECB were posted.
04 (X'04')	X'08'	ISGENQRC	ISGENQRS	Return-code (RC) and reason-code (RS) from ISGENQ.
04 (X'04')	X'0C'	ISGQRC	ISGQRS	Return-code (RC) and reason-code (RS) from ISGQUERY.
04 (X'04')	X'0D'	X'00'	X'0E'	UCBVOLI is binary-zeroes (volume has gone offline).
04 (X'04')	X'0D'	X'00'	X'0F'	UCBVOLI doesn't match the volser in the volume-label.
04 (X'04')	X'0E'	X'00'	X'10'	ISGQUERY-results indicates that pointer-to the first 'RS' (Resource) record is zero.

Table 47. DADSM LSPACE 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**
04 (X'04')	X'0E'	X'00'	X'11'	ISGQUERY-results indicates that pointer-to the first 'RQ' (Requester) record is zero.
04 (X'04')	X'0E'	X'00'	X'12'	ISGQUERY-results indicates that no 'RQ' (requester) is the "owner" of the SYSVTOC resource.
04 (X'04')	X'0E'	X'00'	X'13'	The required "RESERVE" of SYSVTOC was not performed by the caller's address space.
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.
14 (X'0E')	X'86'	X'00'	X'00'	The LSPACE function has timed out, not waiting on the ISGENQ of SYSVTOC.
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.
16 (X'10')	X'01'	X'00'	X'0A'	The size of the referenced parameter list is not large enough to accommodate the specified parameters.
16 (X'10')	X'01'	X'00'	X'14'	The value specified for the TIMEOUT parameter is zero or negative
16 (X'10')	X'01'	X'00'	X'15'	The value specified for the TIMEOUT parameter exceeds the maximum allowed value of 86,400 seconds
20 (X'14')	X'85'	System completion code		Processing error in LSPACE.
20 (X'14')	X'86'	X'00'	X'0C'	The ISGENQ OBTAIN for the SYSVTOC volser resource timed out.

Table 48. 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 118.
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*.
ISGENQRC	Return code from ISGENQ. For a description of the return code, see *z/OS MVS Programming: Assembler Services Reference ABE-HSP*.
ISGENQRS	Reason code from ISGENQ. For a description of the return code, see *z/OS MVS Programming: Assembler Services Reference ABE-HSP*.
ISGQRC	Return code from ISGQUERY. For a description of the return code, see *z/OS MVS Programming: Assembler Services Reference IAR-XCT*.
ISGQRS	Reason code from ISGQUERY. For a description of the return code, see *z/OS MVS Programming: Assembler Services Reference IAR-XCT*.
N/A	Not Applicable.
VRFRET	Return code from CVAFVRF.

Table 49. 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 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 50 on page 152 shows the return codes that might appear in this message.

Table 50. 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.

Table 50. DADSM ISJF Return Codes (continued)

Return Code (R15)	Description
16 (X'10')	Invalid parameter list.

Table 51 on page 153 describes the diagnostic information that appears in message IEC614I, which contains the return code, subfunction identifier, subfunction return code, and subfunction reason code.

Table 51. 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 OBTAIN function return codes

Table 52 on page 154 describes DADSM OBTAIN function return codes.

Table 52. 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.
24 (X'18')	Data set has a format-8 DSCB and EADSCB=N0T0K is in effect
28 (X'1C')	Internal error with EADSCB=NOTOK in effect.

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 53 on page 154 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 54 on page 155 shows. Each volume's volume list entry contains the rename status code shown in Table 56 on page 161.

If an error occurs, DADSM issues message IEC614I, that 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 53 on page 154 describes the conditions indicated by the DADSM SVC return code.

Table 53. 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.

Table 53. DADSM RENAME Return Codes (continued)

DADSM SVC Return Code	Description
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 54 on page 155 describes the conditions that are indicated by the diagnostic information that is returned in register 0. The abbreviations used in Table 54 on page 155 are defined in Table 55 on page 160.

Table 54. 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 or more than the maximum number of volumes allowed for a data set.
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.

Table 54. 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'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.
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 Renaming a Data Set That Might be in Use in *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.

Table 54. 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'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.
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'04'	X'14'	X'00'	X'AA'	Data set could not be created, DSNTYPE=LARGE not valid for this data set.
X'08'	X'03'	X'00'	X'05'	Logical error on SSI call; bad SSOB format, length, and so forth.
X'08'	X'05'	X'00'	X'41'	One or more volumes in the volume list is a read-only volume.
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.

Table 54. 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'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; an exit routine associated with the IGGPRE00_EXIT preprocessing dynamic exit, such as IGGPRE00, rejected the request; the subfunction return code field contains the return code from the exit routine.
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.
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.

Table 54. 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'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.
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.

Table 54. 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'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 55 on page 160 describes abbreviations for DADSM RENAME diagnostic information.

Table 55. 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 118.
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 an exit routine, such as IGGPRE00, associated with the IGGPRE00_EXIT preprocessing dynamic exit. 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.

Table 55. Abbreviation Descriptions for DADSM RENAME Diagnostic Information (Register 0) (continued)

Abbreviation	Description
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 552.

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 56 on page 161.

Table 56. 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 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 57 on page 162.

DADSM PARTREL returns 4 bytes of diagnostic information in register 0. See Table 58 on page 163 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 57 on page 162 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 57. DADSM Partial Release Function Return Codes

Return Code (R15)	Description
0 (X'00')	Function successful.
2 (X'02')	Unable to find extent in format-1 or format-8 DSCB.
3 (X'03')	Exceeded maximum number format-3 pointers in format-9 DSCB.
4 (X'04')	Unable to find extent in format-3 DSCB.
5 (X'05')	Unable to release on a read-only volume.
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 an exit routine, such as IGGPRE00, associated with the IGGPRE00_EXIT preprocessing dynamic exit.
52 (X'34')	RTA to CCHH conversion error.

Table 57. DADSM Partial Release Function Return Codes (continued)

Return Code (R15)	Description
60 (X'3C')	SORT EDT ERROR
64 (X'40')	An error occurred during conversion from format-7 to extent descriptor table.
68 (X'44')	An error occurred during conversion from format-5 to extent descriptor table.
72 (X'48')	Input DSCB is not a format-5 or a format-7 DSCB.
76 (X'4C')	An error occurred during conversion from extent descriptor table to format-7.
80 (X'50')	An error occurred during conversion from extent descriptor table to format-5.
84 (X'54')	VSAM standard format data set incorrectly specified. If the data set is VSAM, it must be extended format.
88 (X'58')	VSAMTRKS address parameter must be specified when VSAMTYPE is specified.
92 (X'5C')	The name of the specified VSAM data set is not a VSAM cluster data component.
96 (X'60')	Catalog call to partial release failed.
100 (X'64')	Request was directed to a VSAM data set defined with guaranteed space.
104 (X'68')	Unexpected reason code received back from a call to catalog services.
108 (X'6C')	Unable to find enough F0's to complete the PARTREL request.
112 (X'70')	DAPCCHH passed is in cylinder-managed space but is not on a multi-cylinder unit boundary.
116 (X'74')	The first extent of a compressed data set is not large enough for space to be released.
178 (X'B2')	Refer to message IDC3009I, return code 178 for possible causes. See *z/OS MVS System Messages, Vol 6 (GOS-IEA)* for more information.

Table 58 on page 163 describes the conditions that are indicated by the diagnostic information that is returned in register 0. The abbreviations that are used in Table 58 on page 163 are defined in Table 59 on page 169.

Table 58. 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'01'	X'00'	X'1E'	Partial release was attempted for a data set on a read-only device.
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.

Table 58. 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'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'37'	X'00'	X'1D'	No space released for compressed data set.
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.
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.

Table 58. 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'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 that 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 that 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 that 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.
X'08'	X'30'	X'04'	X'14'	RTA to CCHH conversion routine encountered an input RTA that 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 that 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' that was not smaller than the maximum number of cylinders on the volume.

Table 58. 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'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'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 that 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 that 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 that 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 that 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 that 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 that 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.
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.

Table 58. 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'20'	CCHH to RTA conversion routine encountered an END 'CC' value that 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 that 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 that was not less than the maximum number of tracks per cylinder.
X'08'	X'41'	X'06'	X'01'	Converting from 4-byte EDT table to a 2-byte EDT table. Invalid eye-catcher detected.
X'08'	X'41'	X'06'	X'02'	Converting from 4-byte EDT table to a 2-byte EDT table. Invalid control block 'level.'
X'08'	X'41'	X'06'	X'03'	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'	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'	Converting from 4-byte EDT table to a 2-byte EDT table. Invalid eye-catcher detected.
X'08'	X'43'	X'06'	X'02'	Converting from 4-byte EDT table to a 2-byte EDT table. Invalid control block 'level.'
X'08'	X'43'	X'06'	X'03'	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'	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.
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.

Table 58. 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'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'	Sorting Extent Descriptor table. Eye-catcher is invalid.
X'08'	X'58'	X'07'	X'02'	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'	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'	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'	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'	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'	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'	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'	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'	Attempting to convert an EDT into a F7 DSCB. EDT eye-catcher is invalid.
X'08'	X'5E'	X'0B'	X'02'	Attempting to convert an EDT into a F7 DSCB. EDT control block 'level' is invalid.
X'08'	X'5E'	X'09'	X'01'	Attempting to convert an EDT into a F5 DSCB. EDT eye-catcher is invalid.
X'08'	X'5E'	X'09'	X'02'	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.

Table 58. 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'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 59. 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*.

Table 59. Abbreviation Descriptions for PARTREL Diagnostic Information (Register 0) (continued)

Abbreviation	Description
CVSTAT	CVSTAT field of the CVAF parameter list. For a description of the fields, see "CVSTAT field codes" on page 118.
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 an exit routine, such as IGGPRE00, associated with the IGGPRE00_EXIT preprocessing dynamic exit. 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 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 60 on page 171. 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 61 on page 171. Each volume list entry contains the scratch status code as shown in Table 63 on page 179.

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 60 on page 171 describes the conditions that are indicated by the DADSM SVC return code.

Table 60. 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.

Table 61 on page 171 describes the conditions that are indicated by the diagnostic information that is returned in register 0. The abbreviations that are used in Table 61 on page 171 are defined in Table 62 on page 178.

Table 61. 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'	Validate DADSM SCRATCH parameter list; the count of volumes in the volume list is less than or equal to 0 or more than the maximum number of volumes allowed for a data set.
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.

Table 61. 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'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.
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.

Table 61. 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'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.
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; an exit routine associated with the IGGPRE00_EXIT preprocessing dynamic exit, such as IGGPRE00, 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. Note that you should only use the two right most bytes when looking up the CDM reason codes. For explanation see “CDM reason codes” on page 501.
X'04'	X'3E'	COMMRC	COMMRS	System error. Unable to convert Extent Descriptor Table prior to calling the exit routines associated with the IGGPRE00_EXIT preprocessing dynamic exit.
X'04'	X'3F'	COMMRC	COMMRS	System error. Error converting CCHH to relative track address format.

Table 61. 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'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'05'	X'00'	X'4E'	One or more volumes in the volume list is a read-only volume.
X'08'	X'09'	X'00'	X'00'	VDSS detected an error.
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; an exit routine associated with the IGGPRE00_EXIT preprocessing dynamic exit, such as IGGPRE00, 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.

Table 61. 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'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.
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.

Table 61. 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'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.
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.

Table 61. 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'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.
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 62 on page 178 describes the abbreviations for DADSM SCRATCH diagnostic information.

Table 62. 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 501.
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 68 on page 182.
COMMRS	DADSM common reason code. For a description of that reason code, see Table 68 on page 182.
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 118.
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 an exit routine associated with the IGGPRE00_EXIT preprocessing dynamic exit, such as IGGPRE00. 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 552.

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 63 on page 179.

Table 63. 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 64 on page 179.

Table 64. 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 by 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 66 on page 182 contains the applicable DADSM SVC return codes returned in register 15. Table 65 on page 180 describes the conditions indicated by the diagnostic information returned in register 0. 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 65 on page 180 shows. The abbreviations used in Table 65 on page 180 are defined in Table 67 on page 182.

Table 65. 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 65. 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 that 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'03'	X'00'	X'2A'	Update was attempted for a DSCB on a read-only volume.
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.

Table 65. 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'13'	UCBCAPTRC	UCBCAPTRS	IOSCAPU failed when attempting to capture the UCB passed by the caller.
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 66. DADSM UPDATE Return Codes (R15)

DADSM SVC Return Code	Description
00(X'00')	Function Successful
04(X'04')	Categorized Error Condition
08(X'08')	Unexpected Error Condition
12(X'0C')	Invalid Parameter
16(X'10')	Invalid Parameter List

Table 67. 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 68 on page 182 describes the diagnostic codes for the DADSM common functions.

Table 68. DADSM Common Diagnostic Codes

Byte 2	Byte 3	
COMMRC (Function Number)	**COMMRS (Function Description)**	**Description**
X'01'	X'04'	RTA_XXYYZ: Zero value input.

Table 68. DADSM Common Diagnostic Codes (continued)

Byte 2	Byte 3	
COMMRC (Function Number)	**COMMRS (Function Description)**	**Description**
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'02'	X'30'	XXYYZ_RTA: 'Z' value is greater than (2 * TRKS/CYL) - 2.
X'02'	X'34'	XXYYZ_RTA: 'XX' value is greater than the total number of RTAs on the volume.
X'02'	X'38'	XXYYZ_RTA: RTA+1 value is greater than the total number of RTAs on the volume.
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.

Table 68. DADSM Common Diagnostic Codes (continued)

Byte 2	Byte 3	
COMMRC (Function Number)	**COMMRS (Function Description)**	**Description**
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.
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'0F'	X'01'	INITEDT: unexpected condition code from MVCL.
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.
X'11'	X'04'	CCCCcccH_RTA: input to function was all zeros.
X'11'	X'18'	CCCCcccH_RTA: end CCCCcccH is less than the start CCCCcccH.
X'11'	X'1C'	CCCCcccH_RTA: start CCCCccc value is not less than MAX CYL.
X'11'	X'20'	CCCCcccH_RTA: end CCCCccc value is not less than MAX CYL.
X'11'	X'24'	CCCCcccH_RTA: start 'H' value is not less than MAX TRKS.
X'11'	X'28'	CCCCcccH_RTA: end 'H' value is not less than MAX TRKS.
X'12'	X'04'	RTA_CCCCcccH: input to function was all zeros.
X'12'	X'10'	RTA_CCCCcccH: end RTA is not higher than the start RTA.
X'12'	X'14'	RTA_CCCCcccH: input RTA value cannot be converted (RTA is too large).
X'12'	X'14'	RTA_CCCCcccH: input RTA value cannot be converted (RTA is too large).
X'12'	X'1C'	RTA_CCCCcccH: start RTA value is not less than MAX CYL.
X'12'	X'20'	RTA_CCCCcccH: end RTA value is not less than MAX CYL.
X'13'	X'04'	RTA_XXXXYYYYZZZZ: Zero value input.
X'13'	X'08'	RTA_XXXXYYYYZZZZ: Invalid start RTA.

Table 68. DADSM Common Diagnostic Codes (continued)

Byte 2	**Byte 3**	
COMMRC (Function Number)	**COMMRS (Function Description)**	**Description**
X'13'	X'0C'	RTA_XXXXYYYYZZZZ: Invalid end RTA.
X'13'	X'10'	RTA_XXXXYYYYZZZZ: End RTA is not higher than start RTA.
X'13'	X'14'	RTA_XXXXYYYYZZZZ: The input RTA value cannot be converted into the 4–byte field (RTA is too large).
X'13'	X'2C'	RTA_XXXXYYYYZZZZ: 'YYYY' value is not less than MAX CYL.

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 20 on page 185 for a sample title line.

```
JOBNAME   DADSM  funct  ERROR    TCB=tcbaddr  W/A=workarea
```

Descriptions of the variables in this format follow:

- *funct* is the failure-related DADSM function.
- *tcbaddr* is the TCB address.
- *workarea* is the DADSM work-area address.

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

Figure 20. Sample DADSM Dump Title Line

Examine the work area pointed to by *workarea* 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 Table 11 on page 49. For a description of the CVSTAT codes, see “CVSTAT field codes” on page 118.

Chapter 8. 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 28 on page 227 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. Table 69 on page 187 provides a detailed description of all PDA information.

Table 69. 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

Table 69. PDA Information (continued)

Offset Dec(Hex)	Bytes and Bit Pattern	Field Name	Description
29 (X'1D')	3	DISPDID3	ID of third most recent calling function

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 70 on page 189 describes nonzero return codes and their reason and response codes.

Table 70. 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 by an internal system call.

Table 71 on page 190 describes nonzero return codes and their reason and *status codes* if applicable.

Table 71. 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 71. 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 by an internal system call.

Table 72 on page 192 describes nonzero return codes and their reason codes and *response codes* if applicable.

Table 72. 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.

Table 72. Nonzero Return Codes and Applicable Reason and Response Codes (continued)

Return Code	Description
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.
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.

Table 72. Nonzero Return Codes and Applicable Reason and Response Codes (continued)

Return Code	Description
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 9. DESERV macro diagnostic aids

This section lists all the reason codes returned by the DESERV macro.

DESERV reason codes 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.Table 73 on page 195 lists all DESERV reason codes.

Table 73. DESERV reason codes

Reason Code	Meaning
0000 (X'0000')	Successful processing.
1001 (X'03E9')	The module is buffered by LLA. No connection was established.
1002 (X'03EA')	Some members not found.
1003 (X'03EB')	The SMDE parameter is mutually exclusive with C370LIB(YES).
1004 (X'03EC')	For GETTYPE=SMDE, the input SMDE pointer was zero.
1005 (X'03ED')	AREA and AREAPTR are mutually exclusive.
1006 (X'03EE')	The DCB was not opened with DSORG=PO. This Applies only to the get function when C370LIB=YES.
1007 (X'03EF')	The C370LIB directory indicates that a symbol is associated with a member name but that name does not exist in the PDS directory.
1008 (X'03F0')	While processing a C370LIB directory, inconsistencies were found in the text records.
1009 (X'03F1')	DCBBLKSI is too small.
1010 (X'03F2')	C370LIB(YES) and PDSDE are mutually exclusive.
1011 (X'03F3')	A DESERV GET_ALL call was made where the CONNECT=HOLD parameter was specified, but the caller was not in a system key nor in supervisor state.
1012 (X'03F4')	No members in directory.
1013 (X'03F5')	Caller specified an authorized subpool, but is not authorized.
1014 (X'03F6')	More than one error has occurred. Check codes in DESL.
1015 (X'03F7')	A call to IEWBXILO returned an unexpected return code.
1016 (X'03F8')	For the DESERV function UPDATE, the caller updated a field in the PMAR that is not supported for update.
1017 (X'03F9')	The release function was expecting a subfunction code of connid or smde list. Neither was specified.
1018 (X'03FA')	The SMDE for the release function had a null pointer or eyecatcher was invalid.
1019 (X'03FB')	The connection specified in the SMDE could not be found. Possible user error.
1020 (X'03FC')	File lock unavailable, possible sharing problem.
1021 (X'03FD')	Alias name is same name as primary name for this member.
1022 (X'03FE')	The smde input defined a token section that was deemed to be bad. The item number may have been inconsistent with the alias flag.
1023 (X'03FF')	For an UPDATE function, the caller has changed the length of the PMAR.
1024 (X'0400')	Name to be replaced did not previously exist.
1025 (X'0401')	IEWLCNVT macro call error.
1026 (X'0402')	PMARA to PMAR merge failed.

Table 73. DESERV reason codes (continued)

Reason Code	Meaning
1027 (X'0403')	No space to store connections.
1028 (X'0404')	Number of CONNIDs exceeded.
1029 (X'0405')	IGWFTOKM SET failed.
1030 (X'0406')	IGWFTOKM EXTRACT failed.
1031 (X'0407')	Bad return code from SETLOCK.
1032 (X'0408')	Error converting PDSDE to PMAR.
1033 (X'0409')	JCDM returned an unexpected return and reason code combination.
1034 (X'040A')	Error converting TTR to CCHHR.
1035 (X'040B')	FREEMAIN failure.
1036 (X'040C')	Function supports PDSE program objects only.
1037 (X'040D')	LLA does not support long names.
1038 (X'040E')	The PUT failed because of a name conflict.
1039 (X'040F')	The MEM_DATA must have one member designated as primary.
1040 (X'0410')	The first 8 bytes of a name were all 1001 (X'FF'.
1041 (X'0411')	The id, length, or level of the DESP is invalid.
1042 (X'0412')	The MEM_DATA must have only one member designated as primary.
1043 (X'0413')	MLT is not valid (PUT func). Caller appears to have tried to update the mlt field (UPDATE function).
1044 (X'0414')	Connect token invalid.
1045 (X'0415')	This function requires a PDSE data set.
1046 (X'0416')	Area provided is too small.
1047 (X'0417')	A call to disconnect files was rejected.
1048 (X'0418')	An internal function call to IGWCDFNX was invalid because the function code was invalid.
1049 (X'0419')	** INTERNAL CODE **.
1050 (X'041A')	File token is invalid.
1051 (X'041B')	Address of the PDSDE is 0.
1052 (X'041C')	The concatenation number specified is greater Than the concatenation number of last data set in the concatenation.
1053 (X'041D')	The DEBCHK macro failed. The DCB or DEB was invalid.
1054 (X'041E')	Address of the DEB is 0 or DCB was input But the DCB pointed to by the DEB did not point back to the DEB.
1055 (X'041F')	With function PUT the DCB must be opened for output.
1056 (X'0420')	A compressed name is inconsistent for some reason (such as the index value is greater than the number of names defined for the program object).
1057 (X'0421')	The passed DCB is not opened.
1058 (X'0422')	The address of the DCB is 0.
1059 (X'0423')	To pass the DEB the caller must be in supervisor state or a privileged key.
1060 (X'0424')	The FUNC value is incorrect.
1061 (X'0425')	The length of a name was either 0 or greater than 8.

Table 73. DESERV reason codes (continued)

Reason Code	Meaning
1062 (X'0426')	The address of the NAME parameter is 0.
1063 (X'0427')	The count of entries in MEM_DATA block is 0.
1064 (X'0428')	The address of the MEM_DATA block is 0.
1065 (X'0429')	PUT function requires OPTION field be specified.
1066 (X'042A')	The RELEASE function must be specified with The CONN_ID parameter or the DE_LIST parameter.
1067 (X'042B')	The address of the CONN_ID parameter was 0.
1068 (X'042C')	The number of entries in the DE_LIST was 0.
1069 (X'042D')	The address of the DE_LIST parameter was 0.
1070 (X'042E')	Gap specified was too large. This gap must not be larger than DESP_ENTRY_GAP_MAX.
1071 (X'042F')	The length of the area provided was insufficient. For the GET function, area length must be larger than the fixed portion of the DESB.
1072 (X'0430')	The address of the AREAPTR is 0.
1073 (X'0431')	The address of a DESB provided is 0.
1074 (X'0432')	The GET function accepts only a NAME_LIST or a PDSDE. Neither was provided.
1075 (X'0433')	The connect identifier in SMDE provided is not valid.
1076 (X'0434')	The count of entries in the NAME_LIST was 0.
1077 (X'0435')	The address of the NAME_LIST structure is 0.
1078 (X'0436')	The connect intent specified is not valid.
1079 (X'0437')	Non-zero return code from an IGWFESTK request.
1080 (X'0438')	A purge for a DCLA was attempted but the anchor was not in use.
1081 (X'0439')	An obtain for a DCLA was attempted but the anchor was not free.
1082 (X'043A')	Desd data length is invalid, input length was 0.
1083 (X'043B')	An STCK instruction failed.
1084 (X'043C')	Alias name specified is a primary name and the options did not allow for deleting primary names.
1085 (X'043D')	The maximum allowable connection identifiers (64k) was allocated and a request to add one more failed.
1086 (X'043E')	An I/O error occurred, the post code in the ECB was unexpected.
1087 (X'043F')	Issued by the DESERV recovery routine when entered for an unknown reason (for example, a program check) while the exit routine was in control. Most likely an exit error.
1088 (X'0440')	PDSDE merge into PMAR failed.
1089 (X'0441')	Two MLT names were found for a single member.
1090 (X'0442')	An unexpected Standard Attribute Record (SAR) code was found when reading attribute records.
1091 (X'0443')	A name record was obtained from jcdm but no associated PNAR was found.
1092 (X'0444')	A call to IGWCDFNC failed.
1093 (X'0445')	A call to IGWFECHG failed.
1094 (X'0446')	A call to GETMAIN failed.

Table 73. DESERV reason codes (continued)

Reason Code	Meaning
1095 (X'0447')	Issued by DESERV recovery routine when entered for unknown reason while the exit routines were not in control.
1096 (X'0448')	A call to the pacb_locate routine failed.
1097 (X'0449')	Caller not supervisor state, or system key.
1098 (X'044A')	Invalid return code set by DESERV EXIT.
1099 (X'044B')	An EXIT exists and DESERV FUNC EXIT with NOREPLACE specified was issued, current exit is not replaced.
1100 (X'044C'	A ZERO DESP_EXIT_DST_PTR was passed in.
1101 (X'044D')	The DST header is not correct.
1102 (X'044E')	The EXIT_PREV_DSTPTR parm was not set.
1103 (X'044F')	The EXIT_OPTION specified is not supported.
1104 (X'0450')	The EXIT_SCOPE specified is not supported.
1105 (X'0451')	AN EXIT_OPTION=DELETE specified a DST address that was not current, the compare and swap failed.
1106 (X'0452')	The pointer to the previous DST is zero. This is checked for EXIT_OPTION=DELETE.
1107 (X'0453')	The DST header is invalid for the DST pointed to by DESP_PREV_DSTPTR_PTR. This is checked for EXIT_OPTION=DELETE.
1108 (X'0454')	A FUNC=RENAME request specified a new name and a old name that were the same.
1109 (X'0455')	A FUNC=RENAME request specified a new name and a old name that were both greater than 63 bytes.
1110 (X'0456')	A FUNC=RENAME request specified a new name that already exists in the pdse.
1111 (X'0457')	A FUNC=RENAME request specified a new name that is greater than 63 bytes long and was not an original name for the program object.
1112 (X'0458')	A FUNC=PUT request specified an alias name for a primary that was deleted by the time the alias name was to be added to the directory, therefore no directory update was performed for the name.
1113 (X'0459')	An error occurred processing a defer element purge request.
1114 (X'045A')	An error occurred processing a defer anchor purger request.
1115 (X'045B')	An error occurred processing a defer anchor release request
1116 (X'045C')	An error occurred processing a defer anchor obtain request.
1117 (X'045D')	A call to nullify the DCII latch failed
1118 (X'045E')	A call to initialize the DCII latch failed.
1119 (X'045F')	A call to OBTAIN the DCII latch failed.
1120 (X'0460')	A call to RELEASE the DCII latch failed.
1121 (X'0461')	A C370LIB directory is bad. The first record is not an ESD for either @@DC370$ OR @@DC390$.
1122 (X'0462')	A C370LIB directory is bad. EOF encountered before reading END record.
1123 (X'0463')	An igwftokm extract returned a pacb address of zero
1124 (X'0464')	MEMBER_COUNTS and CONCAT(ALL) are mutually exclusive
1125 (X'0465')	A call to IGWLOCK to get a PDSE directory lock failed
1126 (X'0466')	DCII EYE CATCHER BAD

Table 73. DESERV reason codes (continued)

Reason Code	Meaning
1127 (X'0467')	UNIX directories not supported.
1129 (X'0469')	Global exit address is not in common storage.

Chapter 10. 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.

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 215
- "Distributed FileManager/MVS communications error codes" on page 217
- "Distributed FileManager/MVS agent error codes" on page 217
- "Distributed FileManager/MVS file/directory error codes" on page 218
- "Distributed FileManager/MVS locking error codes" on page 218
- "Distributed FileManager/MVS access method error codes" on page 218

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 74 on page 204, Table 75 on page 204, and Table 76 on page 204 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 204.

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 211, "GTF Tracing of Input/Output Buffers," and “Formatting LOGREC records” on page 210. 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 202 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 215) 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” on page 204 for more information. Additionally, an AGNPRMRM is returned. For more information, see Table 74 on page 204.

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 74 on page 204, Table 75 on page 204, and Table 76 on page 204 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 77 on page 205 for more information.

Distributed FileManager/MVS system messages

The messages issued by Distributed FileManager/MVS are described in the topic about GDE messages in *z/OS MVS System Messages, Vol 5 (EDG-GFS)*.

DDM reply messages returned by Distributed FileManager/MVS

Tables Table 74 on page 204, Table 75 on page 204, Table 76 on page 204, and Table 77 on page 205 list DDM reply messages returned by Distributed FileManager/MVS.

Table 74. Message Returned during Conversation Initialization

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

Table 75. 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 76. 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'

Table 76. Reply Messages Returned for Control DSSs by Communications Manager (continued)

General Condition	DDM Reply Message	Possible DDM Error Codes
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'

Table 77. 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

Table 77. DDM Reply Messages Returned by Distributed FileManager/MVS (continued)

DDM Reply Message	Codepoint	Description
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
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:

Command	Keywords
TRACE	,WTRSTART=*membername* [WRAP|NOWRAP]]]
	CT [,WTRSTOP=*jobname*]]
	[CT [,ON] { {COMP=SYSSMS } [,SUB=(*sub[sub],...*)]
	[,PARM=CTISMS*xx*] }]
	[CT [,*nnn*K] { }]
	[CT [,*nnnn*M] { }]
	[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:

Command	Keywords
REPLY	id
	[,JOBNAME=(jobnamelist)]

Command	Keywords
	ASID[,=(asidlist.)] [,OPTIONS=(namename...)] [,WTR=procname|DISCONNECT] [,END]

Note:

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*.

DFM trace options

The valid options for use with this trace facility are:

Table 78. 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

Note:

1. You can also specify ALL rather than a list for both the COMP and SUBCOMP keywords.
2. 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:

```
TRACE CT,OFF,COMP=SYSSMS
```

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.

Type	Exit	Keyword
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 components and formats only those control blocks that are detected to be invalid for the selected components. 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 components.

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 components.

JOB

jobname represents the jobname and associated ASID to select for dump access of control blocks for the selected components.

Table 79. 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.

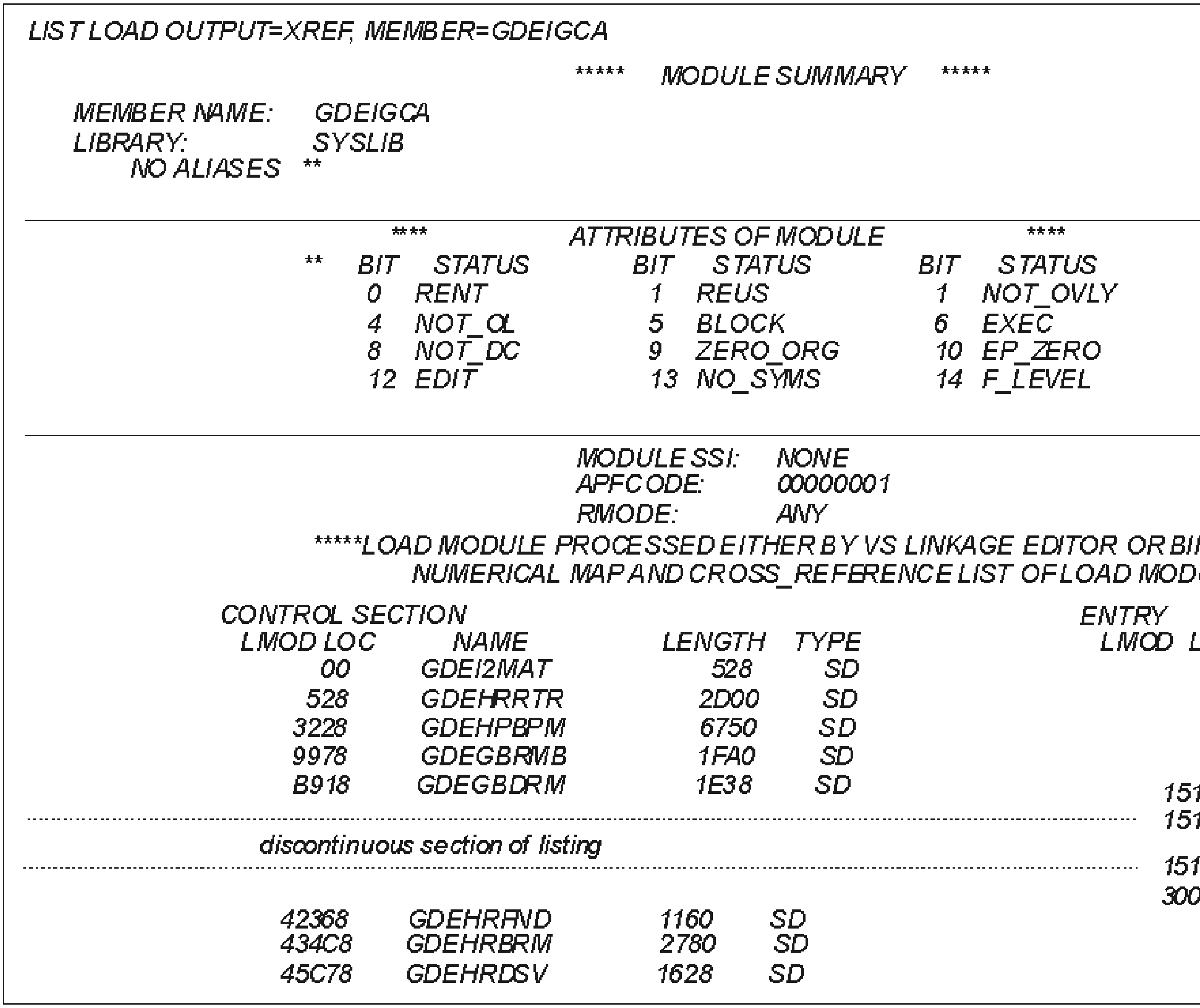
Because only one PER SLIP trap can be set at a time, you need to determine which Reply Message Builder is invoked (GDEHRBRM or GDEGBRMB). In the case of SYNTAXRM and PRCCNVRM, the system LOGREC record tells you the name of the invoked Reply Message Builder. Otherwise, examine the original reply message 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 213. 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 21 on page 212. 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 BI
                               NUMERICAL MAP AND CROSS_REFERENCE LIST OF LOAD MOD

                    CONTROL SECTION                                          ENTRY
                     LMOD LOC      NAME           LENGTH   TYPE               LMOD  L
                          00      GDEI2MAT           528     SD
                         528      GDEHRRTR          2D00     SD
                        3228      GDEHPBPM          6750     SD
                        9978      GDEGBRMB          1FA0     SD
                        B918      GDEGBDRM          1E38     SD                    151
..............................................................................   151
                      discontinuous section of listing
..............................................................................   151
                                                                                 300
                       42368      GDEHRFND          1160     SD
                       434C8      GDEHRBRM          2780     SD
                       45C78      GDEHRDSV          1628     SD
```

Figure 21. 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 204.

Figure 22 on page 214 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 215 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 80 on page 214.

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

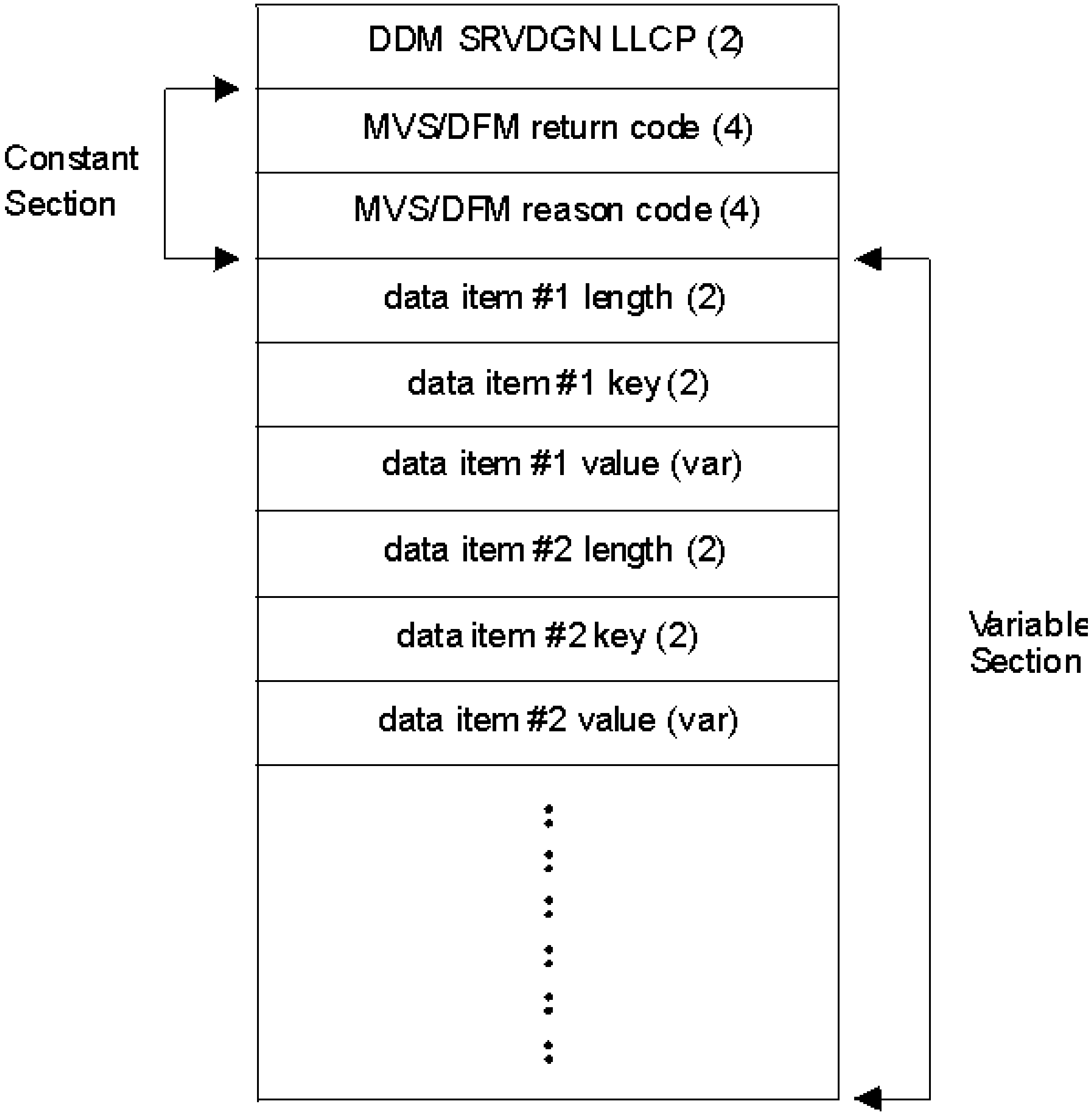


Figure 22. Server Diagnostic Area (SRVDGN) Format

Table 80. 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

Table 80. Distributed FileManager/MVS SRVDGN Data Items (continued)

Data Item Key	Data Item Length	Data Item Description
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 Table 81 on page 215, Table 82 on page 217, Table 83 on page 217, Table 84 on page 218, Table 85 on page 218, and Table 86 on page 218 list Distributed FileManager error codes.

Table 81. 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.

Table 81. Distributed FileManager/MVS General Error Codes (continued)

Error Code	Description
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.
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.

Table 81. Distributed FileManager/MVS General Error Codes (continued)

Error Code	Description
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 82. 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 83. 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.

Table 83. Distributed FileManager/MVS Agent Error Codes (continued)

Error Code	Description
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 84. 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 85. 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 one or more locks that are not owned by the requester.
5002	Requester attempts to release locks 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 86. Distributed FileManager/MVS Access Method Error Codes

Error Code	Description
6001	Maximum record length supported by the MVS access method has exceeded.

Table 86. Distributed FileManager/MVS Access Method Error Codes (continued)

Error Code	Description
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 that 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.
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 (versus 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 11. 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:

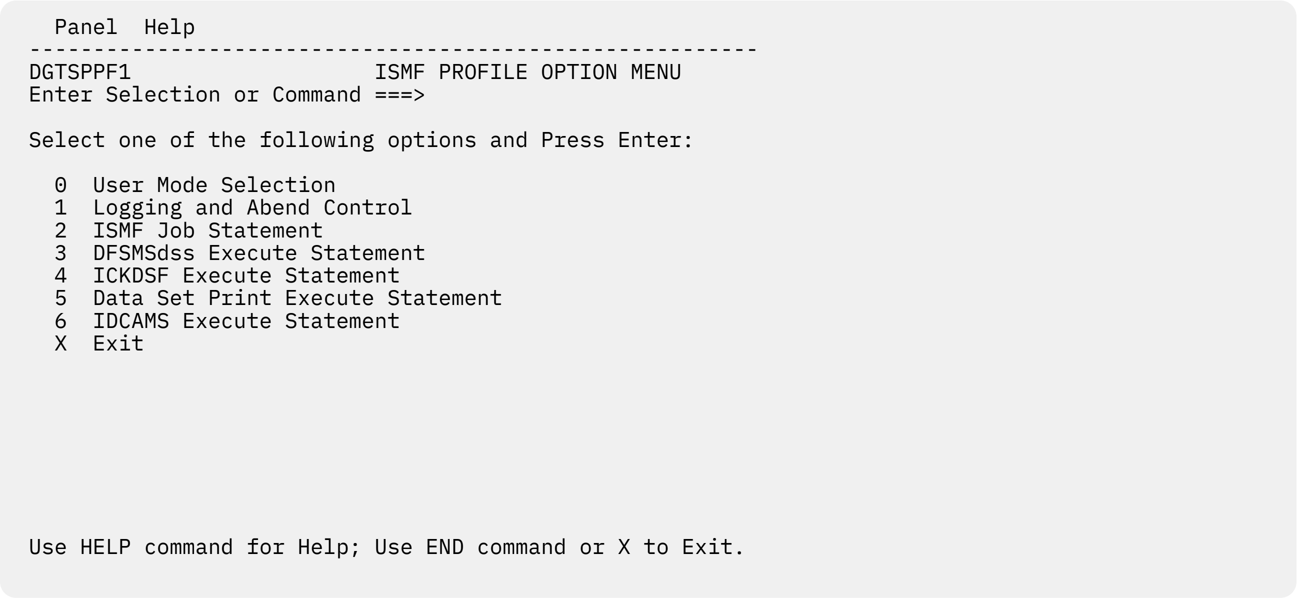
1. Run with **WTP message** turned on in your TSO/E profile. This gives you the message identifier and an online symptom dump for error messages issued by components other than ISMF. *z/OS TSO/E Command Reference* explains how to set up your TSO/E profile.
2. Allocate a SYSUDUMP, SYSABEND, or SYSMDUMP 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" on page 221 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 23 on page 222. 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.



```
  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 23. ISMF—Profile Option Menu

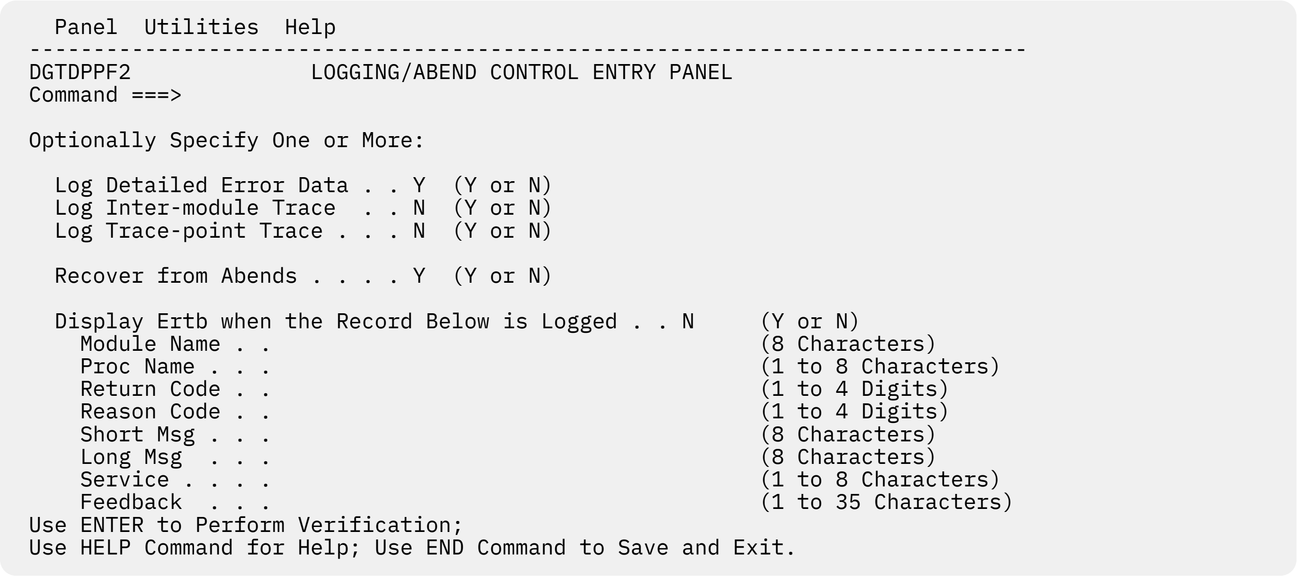
Using the logging/abend control entry panel

Selecting option 1, **LOGGING and ABEND CONTROL**, on the ISMF profile option menu, displays the panel shown in Figure 24 on page 223, which allows you to:

- Specify the amount of information ISMF records (the first three options shown on the panel in Figure 24 on page 223). This is discussed in “ISMF's use of the ISPF log” on page 226.
- View or change the way ISMF handles abends (the fourth option, "RECOVER FROM ABENDS," shown on the panel in Figure 24 on page 223). This is discussed in “Recovering from abends” on page 225.
- Cause ISMF to automatically present the ERTB display panel (see Figure 25 on page 224) 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 24 on page 223.) 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 24. ISMF—Entry Panel for Logging and Abend Parameters

Note: The contents of the ERTB field are defined following Figure 25 on page 224.

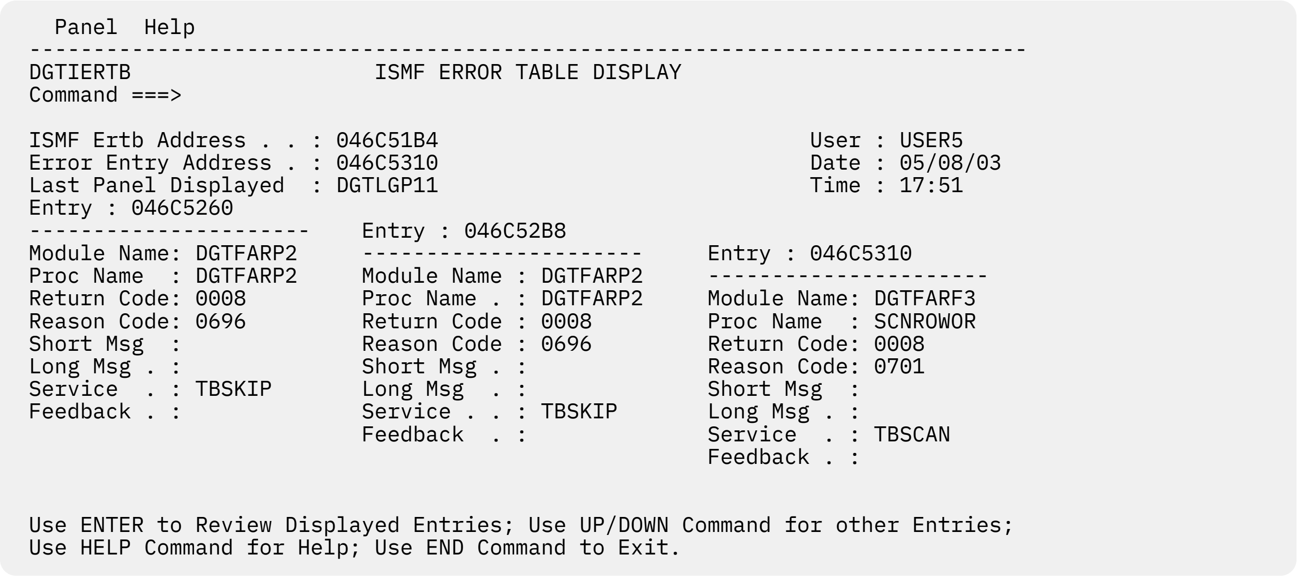
Using the ERTB display panel

You display the ISMF error table (ERTB) panel (shown in Figure 25 on page 224) 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 222.)
- 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 "Keyword worksheet" on page 4, 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 25 on page 224 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 25 on page 224 is the one that matches the values selected in Figure 24 on page 223.



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

ISMF Ertb Address . . : 046C51B4                             User : USER5
Error Entry Address . : 046C5310                             Date : 05/08/03
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 25. 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 that 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 26 on page 225 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 26. 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 27 on page 226 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 27. 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 "TSO/E messages for ISMF abends" on page 12. 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 Vol I*. 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 28 on page 227 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 28. 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 29 on page 228 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 29. 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 30 on page 228 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 30. 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 13, “OAM diagnostic aids,” on page 281.

For DB2 errors, refer to IMS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPH2).

ISMF return codes

When one ISMF function calls another, the return code from the called function is placed in register 15. Table 87 on page 229 lists the standard return codes that are used by ISMF. Some of the individual interfaces may define return codes differently.

Table 87. 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
Errors 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 with 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 or SRVTARN1 (or both) in SRVT control block are 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 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 with 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 by using 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 that 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.

1316
Convert utility: Invalid packed decimal field found while doing a date conversion.

1317
Largest extent in the track-managed space cannot be calculated because some needed data was not acquired.

1318
Error acquiring cylinders in the largest free extent in the track-managed space.

1319
Error acquiring additional tracks in the largest free extent in the track-managed space.

1320
Error acquiring free cylinders in the track-managed space.

1321
Error acquiring additional free tracks in the track-managed space.

1322
Allocated space in the track-managed space cannot be calculated because some needed data was not acquired.

1323
Free space in the track-managed space cannot be calculated because some needed data was not acquired.

1324
Volume capacity in the track-managed space cannot be calculated because some needed data was not acquired.

1325
VTOCLIST line operator not allowed for migrated data set (DS in the MIGRAT VOLSER).

1326
SMS GETMAIN failed.

1327
Storage group volumes copy - duplicate volume.

1328
VTOCLIST line operator is not allowed for the data set.

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 configuration database initialization failed.

3385
OAM configuration database volume table setup failed.

3386
OAM configuration database volume row retrieval failed.

3387
OAM configuration database volume table termination failed.

3388
OAM 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.

3428
Uplevel CDS conversion aborted.

3429
Storage group wasn't found in the "To" SCDS.

6000
Unsupported DFSMS/dfp version number was found.

6001
Unable to locate associations or volume serial numbers for VSAM cluster.

Chapter 12. 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.DUMP*xx* data set using the SDUMP macro instruction. If an indeterminate error occurs, an error record may be written to the logrec data set.

The media manager uses SDUMP to get a dump only if the operating environment was able to establish a functional recovery routine (FRR). An FRR is not established for functions invoked by the MMCNVT macros. Functions invoked by the MMCALL procedure are protected by an FRR. MMSRV is protected by an ESTAE to provide SDUMP information in the case of an indeterminate error.

Figure 31 on page 276 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.

```
----- SUMLIST RANGE -------- ASID FFFF.


00FD6920                                         045820C9 C3E8D4D4 E5E340F0 F961F0F3  *                ...ICYMMVT 09.03*
00FD6940   61F8F7C8 C4D7F3F3 F1F040D5 D6D5C540   404040F5 F6F6F560 E7C1F340 4DC35D40  *.05HDP3310 NONE    5665.XA3 .C. *
00FD6960   C3D6D7E8 D9C9C7C8 E340C9C2 D440C3D6   D9D74B40 F1F9F8F7 6B40F1F9 F8F840D3  *COPYRIGHT IBM CORP. 1987. 2005 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 09/14/05  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   00040200 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*
```

```
COMPON=MEDIA MANAGER, COMPID=28415, ISSUER=ICYFRR            MODULE IEAVTSDT DATE 09/14/05  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                                                          *........                        *
```

```
 -------------- SUMMARY DUMP OUTPUT FOR HASID = 000F PASID = 000F SASID = 000F PSW = 070C0000 810070A4

----- SDWA ----------------- ASID FFFF.
```

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:

```
ccccffss

 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.
	0008	—	Two reads to the same buffer.
	0008	19	PURGE Failure.

ss	*cccc*	*ff*	**Error Description**
	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.
	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.

ss	*cccc*	*ff*	Error Description
	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 *cccc* field X'00xx' indicates MMCALL or MMCNVT requests.

The *ff* field of the return code identifies the module that detected the error.

ff field	Module
—	ICYMMSRV (Media Manager Services)
—	ICYMSTAE (Media Manager Services)
00	ICYFRR
01	ICYRDWR
02	ICYFW
03	ICYPFMT
04	ICYRBA
05	ICYCCHHR
06	ICYSTOR
07	ICYSORT
08	ICYBLIST
09	ICYELE (Event Listen Exit)
0A	ICYBLD31
0B	ICYPF31
0C	ICYDIE, second entry from IOS
0D	ICYPFDIE
0E	ICYNRM
0F	ICYABN
10	ICYPFAPP
11	ICYPGAD
12	ICYPURG
13	ICYSCHED
14	ICYVCACH

ff field	Module
15	ICYWRITE
16	ICYBMFI
17	ICYPIO31
18	ICYREP31
19	ICYRCD0
1A	ICYEXIT
1B	ICYTRACE
1C	ICYBLD64
1D	ICYPF64
1E	ICYPIO64
1F	ICYREP64
20	ICYBLDRE
3A	ICYDIE, first entry from IOS
40	ICYINIT
41	ICYIEDB
42	ICYILPMB

Chapter 13. 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 object 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

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;
```

- Tape Volume Table

```
SELECT * FROM XXXXXXXX.TAPEVOL;
```

where:

- xx - Storage group number from 00-99
- yy - Object name.
- *xxxxxxxx* is the qualifier of the ID that created the table (this information can be obtained from SYSIBM.SYSTABLES).

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 that are shown in Table 88 on page 282.

Note: The F OAM,START,DIAGMSGS,OSREQFS command can be used to specify that OAM messages are to display reason codes that are related to errors in the file system processing for requests that are initiated from the OSREQ macro. See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* for information on how to use this command.

Table 88. 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 89 on page 283 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 that is 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 89. 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 that is completed with a warning condition. Correct program, if necessary. **x** Internal function code **y** Error indication **z** Reserved
4 (X'04')	X'04'	x	X'01'	z	QEL buffer segments are too short to accommodate all available entries; as many entries that might fit in the segments 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 databases. Activate databases, if necessary.
4 (X'04')	X'04'	x	X'03'	z	UNACCESS has completed; the token has been cleared; there are 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.
4 (X'04')	X'04'	x	X'04'	z	STORE, STOREBEG, 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: **1xxx xxxx** Db2 Collection Name Table entry was created for the collection (STORE and STOREBEG only) **x1xx xxxx** ODRETDT overrode RETPD, EVENTEXP, or Management Class expiration date **xx1x xxxx** Storage class specified for the collection was overridden (STORE and STOREBEG only) **xxx1 xxxx** Management class specified for the collection was overridden (STORE and STOREBEG only) **xxxx 1xxx** Retention period specified for the object via RETPD or EVENTEXP was overridden **xxxx x1xx** Reserved **xxxx xx1x** Storage class specified for the object was overridden **xxxx xxx1** 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.

Table 89. OSREQ Return and Reason Codes (continued)

Return Code	Reason Code (Bytes)				Description
	0	1	2	3	
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.
	X'24'	x	X'02'	X'03'	The size specified on an OSREQ STOREBEG is not greater than 256M.
	X'24'	x	X'02'	X'04'	The size specified on an OSREQ STOREPRT is not less than or equal to the total object size specified on the OSREQ STOREBEG for this store sequence.
	X'24'	x	X'02'	X'05'	The size specified on an OSREQ STOREPRT when added to all of the previous OSREQ STOREPRT requests exceeds the total object size specified on the OSREQ STOREBEG for this store sequence.
	X'24'	x	X'02'	X'06'	The size specified on an OSREQ STOREEND is not equal to the size specified on the OSREQ STOREBEG for this store sequence.
	X'24'	x	X'02'	X'07'	The size specified on an OSREQ STOREEND is not equal to the total of the object sizes provided with previous OSREQ STOREPRT requests for this store sequence. Check the previous STOREPRT requests to ensure that they provided all of the parts of the object data and that these previous STOREPRT requests were all successful. Correct calling program.
	X'24'	x	X'02'	X'08'	The size specified on an OSREQ STOREPRT is less than the minimum part size allowed. Only the last STOREPRT in the store sequence can be less than the minimum.
	X'24'	x	X'02'	X'09'	The SIZE value is required, but was not specified
					Correct calling program.
8 (X'08')	X'24'	x	X'03'	X'01'	RETPD area in unusable storage.
		x	X'03'	X'02'	RETPD invalid value, must be -2 thru 93000 or X'7FFFFFFF'
					Correct calling program.
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.

Table 89. OSREQ Return and Reason Codes (continued)

	Reason Code (Bytes)				
Return Code	0	1	2	3	Description
8 (X'08')	X'24'	x X'06'	X'06'	X'01'	QEL buffer list in unusable storage.
			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'06'	X'04'	QEL buffer in unusable storage.
			X'06'	X'05'	The QEL value is required, but was not specified
					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'03'	OBL BUFLIST contains incorrect ID value for a Non-Query.
	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'07'	The QEL or OBL BUFLIST contains incorrect header values.
	X'24'	x	X'08'	X'08'	Amount of buffer data space provided on the retrieve request is insufficient for object.
					Correct calling program.
	X'24'	x	X'08'	X'09'	The QEL or OBL BUFLIST contains incorrect buffer descriptor fields.
	X'24'	x	X'08'	X'0A'	When storing an object greater than 50 MB, multiple data buffers were supplied that are not in contiguous storage.
	X'24'	x	X'08'	X'0B'	When retrieving an object greater than 50 MB and less than or equal to 256 MB, the first data buffer supplied is not large enough to contain the object.
	X24'	x	X'08'	X'0C'	The BUFLIST value is required, but was not specified
					Correct calling program.
8 (X'08')	X'24'	x	X'09'	X'01'	Token area in unusable storage.
				X'02'	Token contains invalid contents.
				X'03'	The TOKEN value is required, but was not specified
					Correct calling program.
8 (X'08')	X'24'	x	X'0A'	X'01'	Name in unusable storage.
	X'24'	x	X'0A'	X'02'	The OBJECT NAME passed to OAM on the OSREQ macro is not fully qualified. The OBJECT NAME contains one or more wildcard characters (‘*’,’%’,’_’) but the function is not QUERY.
	X'24'	x	X'0A'	X'03'	Qualifier longer than 8 characters.
	X'24'	x	X'0A'	X'04'	Name contains an invalid character.
	X'24'	x	X'0A'	X'05'	Null qualifier.
	X'24'	x	X'0A'	X'06'	The OBJECT NAME passed to OAM on the OSREQ macro contains more than one asterisk (‘*’) wildcard and/or an invalid mix of asterisks with percent and/or underscore (‘%’ or ‘_’) characters.

Table 89. OSREQ Return and Reason Codes (continued)

Return Code	Reason Code (Bytes)				Description
	0	1	2	3	
	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.
	X24'	x	X'0A'	X'0A'	The OBJECT NAME value is required, but was not specified
					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.
	X'24'	x	X'0C'	X'04'	The offset specified on an OSREQ STOREPRT is not immediately following the last part of the object stored on the previous OSREQ STOREPRT for this store sequence or is not zero for the first OSREQ STOREPRT for this store sequence.
	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.
				X'04'	The length specified on an OSREQ RETRIEVE is greater than 256M.
					Correct calling program.
8 (X'08')	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 (@, #, $).
	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.

Table 89. OSREQ Return and Reason Codes (continued)

Return Code	Reason Code (Bytes)				Description
	0	1	2	3	
	X24'	x	X'0F'	X'0A'	The COLLECTION NAME value is required, but was not specified 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'24'	x	X'12'	X'01'	RECALL parameter in unusable storage. Correct calling program.
	X'24'	x	X'12'	X'02'	RECALL parameter is larger than maximum allowed. Correct calling program.
	X'24'	x	X'12'	X'03'	RECALL parameter is negative number. Correct calling program.
8 (X'08')	X'24'	x	X'13'	X'01'	RETCODE2 parameter in unusable storage. Correct calling program.
8 (X'08')	X'24'	x	X'14'	X'01'	The STOKEN parameter is in unusable storage for which the application program does not have both fetch and store authorization. This is an indication of a programming logic error in the application program that is issuing the OSREQ macro invocation. Correct calling program.
	X24'		X'14'	X'02'	The STOKEN value provided does not represent a store sequence currently in progress.
	X24'	x	X'14'	X'03'	The STOKEN value is required, but was not specified Correct calling program.
8 (X'08')	X'24'	x	X'15'	z	STIMEOUT
	X'24'	x	X'15'	X'01'	The STIMEOUT area (fullword) passed to OAM on the OSREQ macro is in unusable storage. This means that OAM encountered a virtual storage translation exception (for example, an 0C4 ABEND) when it attempted to reference the area of storage containing the STIMEOUT (fullword).
	X'24'	x	X'15'	X'02'	The value specified for STIMEOUT is invalid.
8 (X'08')	X'24'	x	X'16'	X'01'	EVENTEXP area in unusable storage.
	X'24'	x	X'16'	X'02'	EVENTEXP invalid value, must be 0-93000.
	X'24'	x	X'16'	X'03'	EVENTEXP and RETPD both supplied, only one allowed.
8 (X'08')	X24'	x	X'17'	z	BUFFER64
	X24'	x	X'17'	X'01'	The BUFFER64 area (double word) passed to OAM on the OSREQ macro is in unusable storage. This means that OAM encountered a virtual storage translation exception (for example, 0C4 ABEND) when it attempted to reference the area of storage containing the BUFFER64 value (double word).
8 (X'08')	X24'	x	X'18'	z	SIZE64
	X24'	x	X'18'	X'01'	The SIZE64 area (double word) passed to OAM on the OSREQ macro is in unusable storage. This means that OAM encountered a virtual storage translation exception (for example, 0C4 ABEND) when it attempted to reference the area of storage containing the SIZE64 value (double word).

Table 89. OSREQ Return and Reason Codes (continued)

Return Code	Reason Code (Bytes)				Description
	0	1	2	3	
	X24'	x	X'18'	X'02'	The size specified in the SIZE64 area for an OSREQ STORE request has an invalid value. The value is negative or zero.
	X24'	x	X'18'	X'03'	The size specified in the SIZE64 area for an OSREQ STORE request has an invalid value. The value exceeds the maximum configured object size (which may be up to 2000M).
	X24'	x	X'18'	X'04'	The size specified in the SIZE64 area for an OSREQ STORE request has an invalid value. The value exceeds 256M which is the maximum for an object to be stored to the Optical level of the OAM storage hierarchy.
8 (X'08')	X24'	x	X'19'	z	LENGTH64
	X24'	x	X'19'	X'01'	The LENGTH64 area (double word) passed to OAM on the OSREQ macro is in unusable storage. This means that OAM encountered a virtual storage translation exception (for example, 0C4 ABEND) when it attempted to reference the area of storage containing the LENGTH64 value (double word).
	X24'	x	X'19'	X'02'	The length specified in the LENGTH64 area for an OSREQ RETRIEVE request has an invalid value. The value is negative or zero.
	X24'	x	X'19'	X'03'	The length specified in the LENGTH64 area for an OSREQ RETRIEVE request has an invalid value. For a retrieve of a full object, the value is larger than the object size. For a partial retrieve, the length, when added to the offset specified in the OFFSET64 area, is beyond the end of the object.
8 (X'08')	X24'	x	X'1A'	z	OFFSET64
	X24'	x	X'1A'	X'01'	The OFFSET64 area (double word) passed to OAM on the OSREQ macro is in unusable storage. This means that OAM encountered a virtual storage translation exception (for example, 0C4 ABEND) when it attempted to reference the area of storage containing the OFFSET64 value (double word).
	X24'	x	X'1A'	X'02'	The offset specified in the OFFSET64 area for an OSREQ RETRIEVE request has an invalid value. The value is negative.
	X24'	x	X'1A'	X'03'	The offset specified in the OFFSET64 area for an OSREQ RETRIEVE request has an invalid value. The offset value specified is beyond the last byte of the object.
8 (X'08')	X24'	x	X'1B'	z	DB2ID
	X24'	x	X'1B'	X'01'	The DB2ID area passed to OAM on the OSREQ macro is in unusable storage. This means that OAM encountered a virtual storage translation exception (for example, 0C4 ABEND) when it attempted to reference the area of storage containing the DB2ID value.
	X24'	x	X'1B'	X'02'	The DB2ID passed to OAM on the OSREQ macro contains an invalid length value. It is negative, greater than 4, or (in a multiple OAM configuration only) zero.
	X24'	x	X'1B'	X'03'	The DB2ID passed to OAM on the OSREQ macro does not match any of the Db2 subsystems that have been configured for OAM or OTIS has not yet connected to the designated Db2 subsystem.
	X24'	x	X'1B'	X'04'	The application is running in the CICS or DSN environment or specified the IADDRESS keyword indicating that a connection to a Db2 subsystem has already been made, but the DB2ID passed to OAM on the OSREQ macro is not consistent with the connected Db2 subsystem.
	X24'	x	X'1B'	X'06'	In a multiple OAM configuration, when not in a DSN environment and IADDRESS is not specified, the DB2ID keyword is required but was not specified.
	X'24'	x	X'1B'	X'07'	The application is either running in the CICS or DSN environment, or it has specified the IADDRESS keyword indicating that a connection to a Db2 subsystem has already been made. Either the connected Db2 subsystem does not match any of the subsystems defined in a multiple OAM configuration, or OTIS has not yet connected to the Db2 subsystem.

Table 89. OSREQ Return and Reason Codes (continued)

Return Code	Reason Code (Bytes)				Description
	0	1	2	3	
8 (X'08')	X'28'	x	y	z	IADDRESS routine error. xyz System/User completion code. Correct calling program.
	X'2C'	x	y	z	No valid object found.
8 (X'08')	X'2C'	x	X'01'	z	Directory entry not found.
8 (X'08')	X'2C'	x	X'02'	z	Object segments not found.
8 (X'08')	X'2C'	x	X'03'	z	Backup copy was not found.
8 (X'08')	X'2C'	x	X'04'	z	Second backup copy was not found.
8 (X'08')	X'2C'	x	X'05'	z	The specified object's size is larger than the maximum object size supported by the OSREQ function at the current system level. Retry the requested OSREQ function on a system that supports objects of such size. Correct calling program.
8 (X'08')	X'30'	x	y	z	Object already exists.
8 (X'08')	X'30'	x	X'01'	z	Directory entry already exists.
8 (X'08')	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, 32K, or LOB table for this object. Correct calling program.
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.
8 (X'08')	X'38'	x	y	z	A store sequence function (STOREBEG, STOREPRT, STOREEND) was issued while a STOREBEG is in progress. **z** Reserved and undefined.
8 (X'08')	X'38'	x	X'01'	z	A store sequence function (STOREBEG, STOREPRT, STOREEND) was issued while a STOREBEG is in progress.
8 (X'08')	X'38'	x	X'02'	z	A store sequence function (STOREBEG, STOREPRT, STOREEND) was issued while a STOREPRT is in progress.
8 (X'08')	X'38'	x	X'03'	z	A store sequence function (STOREBEG, STOREPRT, STOREEND) was issued while a STOREEND is in progress.
8 (X'08')	X'38'	x	X'04'	z	A store sequence could not be begun (STOREBEG) because the object location of Optical is not supported for a store sequence.
8 (X'08')	X'38'	x	X'06'	z	On a STOREPRT or STOREEND request for an object to be stored to DASD an attempt to access the Db2 buffer resulted in a -423 Db2 SQL code. It was determined that the Db2 buffer can no longer be accessed. Ensure that the application program did not issue a COMMIT or ROLLBACK during a store sequence which can be a cause of the -423 Db2 SQL code. Once a STOREBEG has been issued, the application cannot perform a COMMIT or ROLLBACK until after the corresponding STOREEND for the store sequence.
8 (X'08')	X'38'	x	X'07'	z	A STOREBEG or STOREPRT request has completed after an UNACCESS has been issued. In this case, the UNACCESS will be deferred and will fail because of the pending store sequence. Ensure that the application program does a STOREEND to finish the store sequence or a STOREEND with CANCEL=YES to cancel the store sequence, then issue UNACCESS again.

Table 89. OSREQ Return and Reason Codes (continued)

Return Code	Reason Code (Bytes)				Description
	0	1	2	3	
8 (X'08')	X'38'	x	X'08'	z	The expected length of the object to be retrieved is greater than the maximum retrieval buffer size of 256M. If LENGTH has a value of 0 or is not specified on an OSREQ RETRIEVE request, then by default the length will be set to the length from either the offset (if OFFSET specified) or beginning (if OFFSET not specified) to the end of the object.
8 (X'08')	X'38'	x	X'09'	z	UNACCESS request can not be processed because of a pending store sequence. Ensure that the application program did a STOREEND to finish the store sequence or a STOREEND with CANCEL=YES to cancel the store sequence before issuing UNACCESS.
8 (X'08')	X'38'	x	X'0A'	z	On a STOREPRT or STOREEND request for an object to be stored to Tape, it was determined that the buffer can no longer be accessed in the OAM address space. Possible cause could be too low STIMEOUT value specified on the OSREQ STOREBEG.
8 (X'08')	X'3C'	X'05'	X'01'	X'01'	OSREQ CHANGE: The EVENTEXP parameter is not allowed because the object is not waiting for an event-based-retention event.
8 (X'08')	X'3C'	X'05'	X'02'	X'02'	OSREQ CHANGE: The RETPD parameter not allowed for an event-based-retention object.
8 (X'08')	X'40'	X'06'	X'01'	X'00'	OSREQ DELETE: Deletion is not allowed because the object is in DELHOLD=HOLD state.
8 (X'08')	X'40'	X'06'	X'02'	X'01'	OSREQ DELETE: Deletion is not allowed because the object is under deletion-protection and is still in event-based-retention state.
	X'40'	X'06'	X'02'	X'02'	OSREQ DELETE: Deletion is not allowed because the object is under deletion-protection and the object’s expiration date has not yet been reached.
8 (X'08')	X'40'	X'06'	X'03'	X'01'	OSREQ DELETE: Deletion is not allowed because the object is under retention-protection and is still in event-based-retention state.
	X'40'	X'06'	X'03'	X'03'	OSREQ DELETE: Deletion is not allowed because the object is under retention-protection and the object’s retention date has not yet been reached.
8 (X'08')	X'44'	x	y	z	64-bit OSREQ request error
8 (X'08')	X'44'	x	X'01'	z	A 64-bit OSREQ STORE request requires that both BUFFER64 and SIZE64 values are specified
8 (X'08')	X'44'	x	X'02'	z	A 64-bit OSREQ RETRIEVE request requires that both BUFFER64 and LENGTH64 values are specified and, optionally, an OFFSET64 can be specified when both BUFFER64 and LENGTH64 have been specified.
8 (X'08')	X'44'	x	X'03'	z	For a 64-bit OSREQ STORE request, BUFFER64 and SIZE64 are mutually exclusive with BUFLIST, RELBUF, and SIZE
8 (X'08')	X'44'	x	X'04'	z	For a 64-bit OSREQ RETRIEVE request, BUFFER64, LENGTH64, and OFFSET64 are mutually exclusive with BUFLIST, LENGTH, and OFFSET

Table 89. OSREQ Return and Reason Codes (continued)

Return Code	Reason Code (Bytes)				Description
	0	1	2	3	
12 (X'0C')	X'50'	x	y	z	Error occurred in file system sublevel configuration **x** Internal function code **y** Error Indication **z** Reserved
	X'50'	x	X'01'	z	File system sublevel work initiated, but no SETDISK statements have been configured.
	X'50'	x	X'02'	z	Directory specified on SETDISK statement for storage group resolves to a path that exceeds the maximum allowable OAM path length
	X'50'	x	X'03'	z	A component of the OAM constructed portion of the directory or file path now includes a symbolic link, however symbolic links are not allowed
	X'50'	x	X'04'	z	The file system type specified on the SETDISK statement for the storage group does not match the mounted file system type at the resolved directory location
	X'50'	x	X'05'	z	The z/OS UNIX file system maximum file size that has been configured has prevented OAM file system activity
	X'50'	x	X'06'	z	The metafile is invalid.
12 (X'0C')	X'51'	x	y	z	Unsuccessful Unix System Services file system access. Look up the *errno*, if provided, in *z/OS UNIX System Services Messages and Codes*. **x** Internal function code **yz** Unix System Services *errno*
12 (X'0C')	X'52'	x	y	z	Unable to access required file system sentinel. Look up the *errno*, if provided, in *z/OS UNIX System Services Messages and Codes*. **x** Internal function code **yz** Unix System Services *errno*
12 (X'0C')	X'53'	x	y	z	Unable to successfully create directory in file system. Look up the *errno*, if provided, in *z/OS UNIX System Services Messages and Codes*. **x** Internal function code **yz** Unix System Services *errno*
12 (X'0C')	X'54'	x	y	z	Unable to open file in file system. Look up the *errno*, if provided, in *z/OS UNIX System Services Messages and Codes*. **x** Internal function code **yz** Unix System Services *errno*

Table 89. OSREQ Return and Reason Codes (continued)

Return Code	Reason Code (Bytes)				Description
	0	1	2	3	
12 (X'0C')	X'55'	x	y	z	Unable to write file in file system. Look up the *errno*, if provided, in *z/OS UNIX System Services Messages and Codes*. **x** Internal function code **yz** Unix System Services *errno*
12 (X'0C')	X'56'	x	y	z	Unable to close file in file system. Look up the *errno*, if provided, in *z/OS UNIX System Services Messages and Codes*. **x** Internal function code **yz** Unix System Services *errno*
12 (X'0C')	X'57'	x	y	z	Unix System Services abnormally ended processing of the file system request with a EC6 abend code. Look up the reason code, if provided, in *z/OS MVS System Codes* under the EC6 abend code. **x** Internal function code **yz** Abend reason code
12 (X'0C')	X'58'	x	y	z	Unix System Services abnormally ended processing of the file system request with a 422 abend code. Look up the reason code, if provided, in *z/OS MVS System Codes* under the 422 abend code. **x** Internal function code **yz** Abend reason code
12 (X'0C')	X'59'	x	y	z	Unix System Services abnormally ended processing of the file system request with an unexpected abend code. Look for messages in the log to identify the abend code then look up the associated reason code, if provided, in *z/OS MVS System Codes* under the abend code. **x** Internal function code **yz** Abend reason code
12 (X'0C')	X'5A'	x	y	z	Unsuccessful Unix System Services file system access in meta file system. Look up the errno, if provided, in z/OS Unix System Services Messages and Codes, YYZZ is Unix System Services errno.
12 (X'0C')	X'5B'	x	y	z	Unable to access required file system sentinel in meta file system. Look up the errno, if provided, in z/OS Unix System Services Messages and Codes, YYZZ is Unix System Services errno.
12 (X'0C')	X'5C'	x	y	z	Unable to open file in meta file system. Look up the errno, if provided, in z/OS Unix System Services Messages and Codes, YYZZ is Unix System Services errno.
12 (X'0C')	X'5D'	x	y	z	Unable to write file in meta file system. Look up the errno, if provided, in z/OS Unix System Services Messages and Codes, YYZZ is Unix System Services errno.
12 (X'0C')	X'5E'	x	y	z	Unable to close file in meta file system. Look up the errno, if provided, in z/OS Unix System Services Messages and Codes, YYZZ is Unix System Services errno.
12 (X'0C')	X'60'	x	y	z	OTIS interface error detected.

Table 89. OSREQ Return and Reason Codes (continued)

Return Code	Reason Code (Bytes)				Description
	0	1	2	3	
12 (X'0C')	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.
12 (X'0C')	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.
	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.
	X'60'	x	X'02'	X'08'	Getmain failed.
	X'60	x	X'02'	X'09'	Unsuccessful attempt to put element on work queue.
	X'60'	x	X'02'	X'12'	Invalid number of parms.
	X'60'	x	X'02'	X'13'	Parameter with invalid size.
	X'60'	x	X'02'	X'14'	Invalid parameter list.
	X'60'	x	X'02	X'0A'	Unsuccessful attempt to take element off completion queue.
12 (X'0C')	X'60'	x	X'03'	X'0B'	Error during collection table update. Db2 not connected.
12 (X'0C')	X'60'	x	X'03'	X'0C'	Error during collection table update. Db2 thread not OPEN. Correct application program.
	X'60'	x	X'03'	X'0D'	The Db2 subtask is terminating.
	X'60'	x	X'03'	X'0E'	The DB2ID was not specified.
	X'60'	x	X'03'	X'0F'	The DB2ID specified is not valid. An OAM instance has not been configured for the Db2 subsystem.
	X'60'	x	X'03'	X'10'	The DB2ID specified is not valid. A non-blank value was specified in a classic OAM configuration.
	X'60'	x	X'03'	X'11'	The OTIS address space is terminating.
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'68'	x	X'03'	z	The OAM configuration contains no OAM subsystems because all previously defined OAM subsystems have been removed.
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 299 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 89. OSREQ Return and Reason Codes (continued)

Return Code	Reason Code (Bytes)				Description
	0	1	2	3	
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 299 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	Db2 detected failure (media). yz represents the Db2 SQL error. Convert the yz to decimal, and look up the resulting SQL code. **yz** Db2 SQL code
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. **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. **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'7C'	x	X'01'	z	Insufficient virtual storage for re-blocking.
12 (X'0C')	X'7C'	x	X'02'	z	Insufficient virtual storage for OAM buffer.
12 (X'0C')	X'7C'	x	X'03'	z	Insufficient virtual storage for work area.
12 (X'0C')	X'7C'	x	X'04'	z	Insufficient 31-bit virtual storage available in the application address space needed to transfer data to or from a 64-bit addressable storage area
12 (X'0C')	X'7C'	x	X'05'	z	Insufficient virtual storage for parameter list.
12 (X'0C')	X'80'	x	y	z	Database contents inconsistent.
12 (X'0C')	X'80'	x	X'01'	z	Directory entry location invalid.
12 (X'0C')	X'80'	x	X'02'	z	Directory entry object size not equal to sum of segment sizes.
12 (X'0C')	X'80'	x	X'03'	z	The object size returned by OAM interface not equal to size requested.
12 (X'0C')	X'80'	x	X'04'	X'00'	ODSTATF_EBR inconsistent with ODEXPDT. ODEXPDT must be '0002-02-02' for event-based-retention object.

Table 89. OSREQ Return and Reason Codes (continued)

Return Code	Reason Code (Bytes)				Description
	0	1	2	3	
12 (X'0C')	X'80'	x	X'05'	X'00'	ODSTATF_RETPROT inconsistent with ODRETDT. For retention-protected object, the ODRETDT can only be ‘0001-01-01’ if the ODSTATF_EBR flag is ON.
12 (X'0C')	X'80'	x	X'06'	X'00'	ODEXPDT inconsistent with ODRETDT for retention-protected object
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. **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. **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. **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. **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. **yz** Db2 SQL code
12 (X'0C')	X'98'	x	y	z	Error attempting to retrieve variable value from Db2. yz represents the Db2 SQL error. Convert the yz to decimal, and look up the resulting SQL code. 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 525.
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. **yz** Db2 SQL code
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.
12 (X'0C')	X'A8'	x	X'02'	z	Parameter list error.
12 (X'0C')	X'A8'	x	X'03'	z	SMS subsystem not active.
12 (X'0C')	X'A8'	x	X'04'	x	SMS subsystem not installed.
12 (X'0C')	X'A8'	x	X'05'	z	Error invoking SMS subsystem.
12 (X'0C')	X'A8'	x	X'06'	z	SMS is restarting.

Table 89. OSREQ Return and Reason Codes (continued)

Return Code	Reason Code (Bytes)				Description
	0	1	2	3	
12 (X'0C')	X'A8'	x	X'08'	z	Storage group not connected.
12 (X'0C')	X'A8'	x	X'09'	z	Constructs not assigned by SMS.
12 (X'0C')	X'A8'	x	X'0A'	z	No expiration criteria in management class
12 (X'0C')	X'A8'	x	X'0C'	z	Storage group has the wrong associated Db2 identifier in a multiple OAM address space configuration.
12 (X'0C')	X'A8'	x	X'0D'	z	Storage group has the wrong associated Db2 identifier in a multiple OAM address space configuration. A new collection name has been inserted into the OAM Collection table for this Storage Group. You might need to manually remove this new collection or update the DB2ID in the storage group definition.
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'B2'	x	y	z	Request denied due to Db2 environment
12 (X'0C')	X'B2'	x	X'01'	z	For a 64-bit OSREQ STORE request, the required LOB environment does not exist for storage of an object greater than 256M to the Db2 sublevel
12 (X'0C')	X'B4'	x	X'01'	z	Collection definition errors.
12 (X'0C')	X'B4'	x	X'01'	z	Collection is not defined (on any request other than a store).
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.
12 (X'0C')	X'BC'	x	y	z	Installation exit CBRUXSAE has resulted in an abend. This OSREQ request is failed and subsequent requests will fail until the error in CBRUXSAE is fixed and the exit is reset using the LIBRARY RESET,CBRUXSAE command. OSREQ functions that are in BYPASSED mode prior to the abend are not affected, and will not be failed. The D SMS,OAM command can be used to display the CBRUXSAE status for each OSREQ function type. **x** represents the OSREQ function code **y** Reserved for IBM use. **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'CC'	x	X'01'	z	Invalid Db2 data base name.
16 (X'10')	X'CC'	x	X'02	z	Invalid OSMI data base function code.

Table 89. OSREQ Return and Reason Codes (continued)

Return Code	Reason Code (Bytes)				Description
	0	1	2	3	
16 (X'10')	X'CC'	x	X'03'	z	Insufficient storage for OSMI data base services.
16 (X'10')	X'CC'	x	X'04'	z	Invalid OSMI work area chain.
16 (X'10')	X'CC'	x	X'05'	z	An OSMI work area has been damaged.
16 (X'10')	X'CC'	x	X'06'	z	ASXB ptr to OMCB is invalid.
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 299. and look for the associated OAM macro reason code. The return code 16 is an OSREQ return code, and may not match the return code for the OAM macro.
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. **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. **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'DD'	x	y	z	OAM File System Delete Table Db2 error. *yz* represents the Db2 SQL error. Convert the *yz* to decimal, and look up the resulting SQL code. **x** Internal function code **yz** Db2 SQL code
16 (X'10')	X'DE'	x	y	z	OAM File System Delete Table general error.
16 (X'10')	X'E0'	x	y	z	SMSI detected error. **y** error indicator
16 (X'10')	X'E0'	x	X'06'	z	Error issuing messages from SMS.
16 (X'10')	X'E0'	x	X'07'	z	Reserved for IBM use.
16 (X'10')	X'E0'	x	X'08'	z	Reserved for IBM use.
16 (X'10')	X'E0'	x	X'0A'	z	Error on ATTACH macro.
16 (X'10')	X'E0'	x	X'0C'	z	Reserved for IBM use.
16 (X'10')	X'E0'	x	X'0D'	z	Unexpected error from Db2.
16 (X'10')	X'E4'	x	y	z	Store sequence environmental error: **y** Error indicator.
16 (X'10')	X'E4'	x	X'01'	z	A STOREBEG request was issued for an object to be stored on DASD, but the required Db2/LOB environment does not exist.

Table 89. OSREQ Return and Reason Codes (continued)

Return Code	Reason Code (Bytes)				Description
	0	1	2	3	
16 (X'10')	X'E4'	x	X'02'	z	A STOREBEG request was issued for an object to be stored on File system or Tape, but a buffer could not be acquired in the OAM address space for the object. Ensure system configured to make sufficient 64-bit addressable virtual memory available to the OAM address space.

RETCODE2

RETCODE2 is an optional parameter on the OSREQ API. It can be used to determine if OAM scheduled additional processing for this OSREQ request. The information returned in RETCODE2 depends on the OSREQ function (RETRIEVE or STORE) requested. Refer to *z/OS DFSMS OAM Application Programmer's Reference* for more information on the OSREQ API and its keywords.

For an OSREQ RETRIEVE request, RETCODE2 specifies whether this RETRIEVE request resulted in scheduling a RECALL of the object to Db2 DASD. The value returned in this keyword is valid only when the RETRIEVE is successful, in which case it provides the following information:

Table 90. RETCODE2 values for OSREQ RETRIEVE

RETCODE2	Description
0 (X'00')	Either RECALL not specified on OSREQ RETRIEVE request so no attempt was made to schedule recall, or RECALL specified on OSREQ RETRIEVE request and recall successfully scheduled
4 (X'04')	RECALL not specified on OSREQ RETRIEVE request, however, recall successfully scheduled due to CBROAMxx PARMLIB member specifications.
8 (X'08')	An attempt to schedule a recall for this object was not successful because OSMC=No specified on OAM started procedure.
10(X'0A')	An attempt to schedule a recall for this object was not successful because MAXRECALLTASKS(0) specified in CBROAMxx member of PARMLIB
12 (X'0C')	An attempt to schedule a recall for this object was not successful because RECALLOFF(ON) specified in CBROAMxx member of PARMLIB
14 (X'0D')	An attempt to schedule a recall for this object was not successful due to unexpected scheduling error.
16 (X'10')	An attempt to schedule a recall for this object was not successful because the retrieve was performed on a down-level OAMplex member that does not support recall processing.

For an OSREQ STORE request, RETCODE2 specifies whether this STORE request resulted in scheduling an Immediate Backup copy be written for this object. The value returned is only valid when the STORE is successful, in which case it provides the following information:

Table 91. RETCODE2 values for OSREQ STORE

RETCODE2	Description
0 (X'00')	Immediate backup copy request successfully scheduled.
4 (X'04')	Immediate backup copy request not required.
8 (X'08')	An attempt to schedule an immediate backup for this object was not successful because OSMC is not up and running.

Table 91. RETCODE2 values for OSREQ STORE (continued)

RETCODE2	Description
14 (X'0D')	An attempt to schedule an immediate backup for this object was not successful due to unexpected scheduling error.
16 (X'0F')	Immediate backup to optical not supported for STOREEND.

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 Table 92 on page 299.

Table 92. 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 93 on page 299 presents the reason codes that are associated with each of the return codes.

Table 93. 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.

Table 93. OAM Macro Reason Codes (continued)

Return Code	Error Code	Reason Code	Description
		X'0040'	Part of volume list operation was successfully scheduled.
		X'0041'	Part of a library audit was successfully scheduled.
		X'0042'	File system request processing cancelled
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.
		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 the current OAM configuration.
		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.

Table 93. OAM Macro Reason Codes (continued)

Return Code	Error Code	Reason Code	Description
		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'044D'	OAM already has a defrag 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 OAM 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 the current OAM configuration.
		X'0461'	Library name not defined in OAM configuration database.
		X'0462'	NONGROUP volume set not defined in OAM 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.
		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'0469'	A write request for tape sublevel2 was received, but the storage group associated with the write has a blank L2TAPEUNITNAME.
		X'0470'	Invalid SUBLEVEL option specified.
		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'0483'	Requested Object was detected to have a possible corrupted length value written in the prefix of the tape record or the Db2 directory table.
		X'0490'	Buffer address missing.
		X'0491'	Buffer length missing.

Table 93. OAM Macro Reason Codes (continued)

Return Code	Error Code	Reason Code	Description
		X'0492'	One buffer parameter supplied and one missing for TYPE(READ).
		X'0493'	Multiple buffer specifications (both a buffer address and a buffer manager token specified, or both a 31 bit and a 64 bit buffer address specified).
		X'0494'	A buffer manager token was provided on a request (READ) that did not originate from within the OAM address space.
		X'0495'	A 64-bit buffer address was specified on a request that originated from within the OAM address space
		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'04B0'	File system driver parameter list missing
		X'04B1'	File system driver parameter list identifier invalid
		X'04B2'	File system driver request type invalid
		X'04B3'	File system driver file system type invalid
		X'04B4'	File system driver file system directory invalid
		X'04B5'	File system driver system name invalid
		X'04B6'	File system driver storage group name invalid
		X'04B7'	File system driver collection name invalid
		X'04B8'	File system driver object name invalid
		X'04B9'	File system driver request origination type invalid
		X'04BA'	File system driver conflicting buffer specifications
		X'04BB'	File system driver missing buffer specification
		X'04BC'	File system driver buffer length invalid
		X'04BD'	File system driver buffer length conflicts with request size
		X'04BE'	File system driver instance id missing on a read
		X'04BF'	File system driver instance id invalid on a write
		X'04C0'	File system driver instance id missing on a delete
		X'04C1'	File system driver actual file system length less than expected
		X'04C2'	File system driver work-to-do data missing on a read
		X'04C3'	File system driver work-to-do data missing on a write
		X'04C4'	File system driver work-to-do data invalid on a delete
		X'04C5'	File system driver object size less than buffer length
		X'04C6'	File system driver object size greater than buffer length
		X'04C7'	File system driver read position exceeds actual file size
		X'04C8'	File system driver read offset exceeds actual file size
		X'04C9'	Missing file system driver connection token
		X'04CA'	Missing task name
		X'04CB'	Compliance type invalid or missing
		X'0500'	Invalid type specified on a QUERY request.

Table 93. OAM Macro Reason Codes (continued)

Return Code	Error Code	Reason Code	Description
		X'0520'	Volume that was a subject of a QUERY (for optical or tape) was not found in the OAM configuration.
		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 writes 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'0700'	Invalid 64-bit addressable storage buffer token.
		X'0701'	Invalid length specified for data in 64-bit addressable buffer.
		X'0702'	Invalid offset specified for data in 64-bit addressable buffer.
		X'0710'	No SETDISK statements have been specified
		X'0711'	Missing storage group name on a file system request
		X'0712'	Missing file system instance id on a file system read or delete request
X'0C'		X'0800'	OAM address space not available.
		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.
		X'0813'	Resource required to complete this request owned by other OAM in OAMplex, but request exceeds 50MB limit for sending across OAMplex.

Table 93. OAM Macro Reason Codes (continued)

Return Code	Error Code	Reason Code	Description
		X'0820'	File system driver error performing a "stat" function request
		X'0821'	File system driver error performing a "close" function request
		X'0822'	File system driver error performing a "remove" function request
		X'0823'	File system driver detected a file name within a path
		X'0824'	File system driver detected error when attempting to determine the file system type
		X'0825'	File system driver received an invalid request type
		X'0826'	File system driver detected unexpected "mkdir" error
		X'0827'	File system driver received an invalid open request
		X'0828'	File system driver detected that the file system type of the mounted file system is not the expected type
		X'0829'	File system driver error performing a "mkdir" function request
		X'082A'	File system driver detected that the path length exceeds the maximum length
		X'082B'	File system driver error performing a "realpath" function request
		X'082C'	File system driver detected that the sentinel file is missing
		X'082D'	File system driver detected that the amount of data written was less than amount of data provided
		X'082E'	File system driver detected a symbolic link in the OAM portion of the directory path
		X'082F'	File system driver detected a directory that exceeds the maximum length
		X'0830'	File system driver detected less data read than expected
		X'0831'	File system driver detected invalid offset
		X'0832'	File system driver error performing a "lstat" function request
		X'0833'	File system driver error performing a "open" function request
		X'0834'	File system driver error performing a "read" function request
		X'0835'	File system driver error performing a "seek" function request
		X'0836'	File system driver error performing a "sync" function request
		X'0837'	File system driver error performing a "write" function request
X'0C'	Hardware resource unavailable	X'0900'	No eligible optical drive is capable of executing this request. In a multiple OAM configuration, this could mean there is no SETOAM statement for the active Backup Groups specified on the SETOSMC statement.
		X'0901'	OAM received a request to eject a volume, but that volume is mounted on a drive that 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.

Table 93. OAM Macro Reason Codes (continued)

Return Code	Error Code	Reason Code	Description
		X'0905'	Volume not 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 volume on the target OAM is no longer mounted when the request gets to the target OAM.
		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. This is a result of an operator canceling a previous mount request for this volume (CBR2003I message would have been issued), or an operator command has been entered to mark the volume lost.
		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'0924'	End Of Data has been reached reading or writing to a tape volume.
		X'0925'	A specific write for backup was received, but volume sublevel is not blank.
		X'0926'	A specific write for sublevel 1 was received, but volume sublevel is not '1'.
		X'0927'	A specific write for sublevel 2 was received, but volume sublevel is not '2'.
		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.
		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.

Table 93. OAM Macro Reason Codes (continued)

Return Code	Error Code	Reason Code	Description
		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'0B01'	Write request resulted in MVS Scratch allocation, but the capacity of the tape volume was less than the object size.
		X'0B02'	Write request required an MVS Scratch allocation, but the storage groups MVS Scratch threshold is less than the current object size.
		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.

Table 93. OAM Macro Reason Codes (continued)

Return Code	Error Code	Reason Code	Description
		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 that is offline or pending offline.
		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 that 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'0B73'	Volume was not demounted
		X'0B74'	Specific request for a LOST volume
		X'0B75'	Specific request for an undefined volume
		X'0B76'	Read of an object from the file system failed. Access Backup is active for file system errors; for OSREQ requests a read may be attempted for the backup copy of the object.
		X'0B80'	LCS file system tasks not operational
		X'0B81'	Missing file system directory for storage group
		X'0B82'	Object to be deleted from file system could not be added to file system delete table
X'10'	OAM ABEND	X'0C00'	OAM ABEND during request processing.
		X'0C01'	z/OS Unix ended a LCS file system task with a EC6 abend code. **Note:** The OAM address space must be stopped before shutting down z/OS Unix. Shutting down z/OS Unix when the OAM address space has been started can result in an EC6 abend.
		X'0C02'	z/OS Unix ended a LCS file system task with a 422 abend code.
		X'0C03'	A LCS file system task ended with an unexpected abend code.
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.

Table 93. OAM Macro Reason Codes (continued)

Return Code	Error Code	Reason Code	Description
		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 OAM configuration database.
		X'0D47'	OAM failed in its attempt to update a row into TAPEVOL (tape volume) table in the OAM 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'0D70'	Unable to obtain storage for a 64-bit addressable buffer.
		X'0D71'	Unable to obtain TCB token (which is required to attempt to obtain a 64-bit addressable buffer.
		X'0D80'	The specified object size is greater than the maximum size supported by the optical level of the storage hierarchy.
		X'0D81'	Request requires OSMC to be up yet OSMC is currently down.
		X'0D90'	File system driver unable to MVS LOAD LE CELQPIPI
		X'0D91'	File system driver unable to start LE CELQPIPI
		X'0D92'	File system driver unable to end LE CELQPIPI
		X'0D93'	File system driver unable to invoke LE CELQPIPI
		X'0D94'	File system driver internal logic error
		X'0D95'	File system driver internal logic error
		X'0D96'	File system driver internal logic error
		X'0D97'	File system driver internal logic error
		X'0D98'	File system driver internal logic error
		X'0D99'	File system driver internal logic error
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.
		X'0E0C'	I/O attempted to an offline CTC.
		X'0E0D'	Error attempting to write a full volume.

Table 93. OAM Macro Reason Codes (continued)

Return Code	Error Code	Reason Code	Description
		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.
		X'0E10'	Unspecified internal zero pointer error during OAM processing.

Additional OAM macro reason code information for OAM file system sublevel

Some OAM macro reason codes associated with the OAM file system sublevel can be supplemented with additional diagnostic information in, for example, messages issued by OAM. Table 94 on page 309 indicates the values that could be present in an additional return code or additional reason code.

Table 94. Additional diagnostic information for OAM macro reason codes related to OAM file system sublevel

OAM Macro Reason Code	Additional Return Code	Additional Reason Code	Action
X'0820'		*errno*	Look up the *errno* in *z/OS UNIX System Services Messages and Codes*.
X'0821'		*errno*	Look up the *errno* in *z/OS UNIX System Services Messages and Codes*.
X'0824'		*errno*	Look up the *errno* in *z/OS UNIX System Services Messages and Codes*.
X'0829'		*errno*	Look up the *errno* in *z/OS UNIX System Services Messages and Codes*.
X'082B'		*errno*	Look up the *errno* in *z/OS UNIX System Services Messages and Codes*.
X'082C'		*errno*	Look up the *errno* in *z/OS UNIX System Services Messages and Codes*.
X'0832'		*errno*	Look up the *errno* in *z/OS UNIX System Services Messages and Codes*.
X'0833'		*errno*	Look up the *errno* in *z/OS UNIX System Services Messages and Codes*.
X'0834'		*errno*	Look up the *errno* in *z/OS UNIX System Services Messages and Codes*.
X'0835'		*errno*	Look up the *errno* in *z/OS UNIX System Services Messages and Codes*.
X'0836'		*errno*	Look up the *errno* in *z/OS UNIX System Services Messages and Codes*.
X'0837'		*errno*	Look up the *errno* in *z/OS UNIX System Services Messages and Codes*.
X'0C01'		abend reason code	Look up the abend reason code associated with the SEC6 abend in *z/OS MVS System Codes*.
X'0C02'		abend reason code	Look up the abend reason code associated with the S422 abend in *z/OS MVS System Codes*.

Table 94. Additional diagnostic information for OAM macro reason codes related to OAM file system sublevel (continued)

OAM Macro Reason Code	Additional Return Code	Additional Reason Code	Action
X'0C03'		abend reason code	Look up the abend reason code associated with the abend found in the OAM log or the system log in *z/OS MVS System Codes*.
X'0D90'	R1	R15	Look up the R1 and R15 register values for MVS LOAD in *z/OS MVS Programming: Authorized Assembler Services Reference LLA-SDU*.
X'0D91'	CELQPIPI return code		Look up the CELQPIPI return code under CELQPIPI init_sub in *z/OS Language Environment Programming Guide for 64-bit Virtual Addressing Mode*.
X'0D92'	CELQPIPI return code		Look up the CELQPIPI return code under CELQPIPI term in *z/OS Language Environment Programming Guide for 64-bit Virtual Addressing Mode*.
X'0D93'	Byte 1: CELQPIPI return code Bytes 2-4: bytes 2-4 of the first word of the condition token	Second word of the condition token	Look up the CELQPIPI return code under CELQPIPI call_sub and look up the format of the condition token under "Using Condition Tokens" in *z/OS Language Environment Programming Guide for 64-bit Virtual Addressing Mode*.

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 95 on page 310

CBRXLCS return and reason codes

Note:

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 95. 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.

Table 95. CBRXLCS Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Meaning
		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 is incompatible with the volume media type or the specified drive type.
		131	Scratch volume threshold processing not performed because library was not operational.
		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.
		322	Command rejected by the library.
		323	Unable to retrieve policy names from the library
8	Invalid request	9	Required type parameter not specified.
		10	Mutually exclusive required parameters 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.

Table 95. CBRXLCS Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Meaning
		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.
		24	Invalid volume list, mixed media, rewritable and worm volumes.
		25	Invalid expiration date specified.
		26	Library is not defined to a storage group in the active configuration.
		27	Invalid library name specified.
		28	Invalid write protection date specified.
		29	Invalid address specified for parameter list or mapping macro.
		30	Specified storage not aligned on fullword boundary.
		31	Required TDSI parameter not specified.
		32	Required library ID or library name 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.
		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.
		130	Specified volume already resides in this tape library.
		201	Volume already ejected (not in library).
		202	Invalid value specified for eject option.
		203	Invalid value specified for bulk eject.

Table 95. CBRXLCS Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Meaning
		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.
		217	At least one optional keyword must be specified with the function.
		218	Required datatype not specified.
		219	Invalid policy name specified.
		220	Mutually exclusive optional keywords specified.
		221	Invalid storage header and/or length specified.
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 groups or the library is unknown.
		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	Specified library is not defined to the active SMS configuration.
		63	Volume record not found for requested volume.
		64	Library record not found.
		65	Library logical type not defined.
		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 update volume record.
		81	LCS external services unable to obtain storage.
		82	Installation exit abended.
		83	Installation exit returned invalid return code or data.

Table 95. CBRXLCS Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Meaning
		84 *	Abend occurred during LCS external services processing.
		93	Library is offline, pending offline, or not operational.
		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.
		312	Requested function is incompatible 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.
		322	Command rejected by the library.
		323	Unable to retrieve policy names from the library.
		324	Volume expiration time has not elapsed.
		325	I/O terminated due to timeout detection.
		326	Selective Device Access Control Group denied request.
		327	Selective Device Access Control Group is not valid.
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.

Table 95. CBRXLCS Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Meaning
		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 96 on page 315 describes the return codes.

Table 96. 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.
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 97 on page 315 describes the return codes.

Table 97. CBRXVOL Return Codes

Return Code	Error Type	Description
0	Successful	Successful execution.

Table 97. CBRXVOL Return Codes (continued)

Return Code	Error Type	Description
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 98 on page 316 describes the return codes.

Table 98. 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.

Table 98. CBRXTCF Return Codes (continued)

Return Code	Error Type	Description
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 Table 99 on page 317, Table 100 on page 321, Table 101 on page 323, Table 102 on page 325, Table 103 on page 327, Table 104 on page 329, Table 105 on page 330, and Table 106 on page 331 describe the CBRXLACS return and reason codes. AL pertains to automated library. ML pertains to manual library.

Table 99. 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 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

Table 99. CBRXLACS FUNC(MOUNT) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
		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
		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 that signaled the completion of the mount order contained an undefined mount completion code.	Yes	No	CBR4102I

Table 99. 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 that 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
		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

Table 99. CBRXLACS FUNC(MOUNT) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
		X'0179'	The requested cartridge is damaged in such a way that it cannot be loaded into a tape drive. 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'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'0193'	The previous (clone) volser of the multivolume dataset does not reside in the library.	Yes	No	CBR4170I
		X'0194'	The library encountered a failure when attempting to mount the logical volume. This code is only set after a synchronous request.	Yes	No	CBR4171I
		X'0195'	The mount request was cancelled at the library and could not be completed prior to the library being varied offline by an operator at the library manager. This code is only set after a synchronous request.	Yes	No	CBR4172I
		X'0196'	The specified I/O VTS in the PtP VTS subsystem is not available.	Yes	No	CBR4173I
		X'0197'	The library could not obtain volume ownership information.	Yes	No	CBR4174I
		X'0198'	Selective Device Access Control Group denied request.	Yes	No	CBR4175I
		X'0199'	Selective Device Access Control Group is not valid.	Yes	No	CBR4176I
		X'019A'	WORM volume metadata is inconsistent between the library and the volume. This code is set only after a synchronous request.	Yes	No	CBR4177I
		X'019B'	Library is not WORM enabled.	Yes	No	CBR4178I

Table 99. CBRXLACS FUNC(MOUNT) 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'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
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. AL pertains to automated library. ML pertains to manual library.

Table 100. 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

Table 100. CBRXLACS FUNC(DEMOUNT) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
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
		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

Table 100. CBRXLACS FUNC(DEMOUNT) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
		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
		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. AL pertains to automated library. ML pertains to manual library.

Table 101. 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

Table 101. CBRXLACS FUNC(WAIT) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
		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 scratch mount can be retried.	Yes	No	CBR4012I
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 that 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 that 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.	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
		X'0394'	The library encountered a failure when attempting to mount the logical volume.	Yes	No	CBR4171I

Table 101. CBRXLACS FUNC(WAIT) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
		X'0395'	The mount request was cancelled at the library and could not be completed prior to the library being varied offline by an operator at the library manager.	Yes	No	CBR4172I
		X'039A'	WORM volume metadata is inconsistent between the library and the volume.	Yes	No	CBR4177I
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. AL pertains to automated library. ML pertains to manual library.

Table 102. 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 102. 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 102. 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. AL pertains to automated library. ML pertains to manual library.

Table 103. 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
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

Table 103. CBRXLACS FUNC(CANCEL) 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(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
		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

Table 103. CBRXLACS FUNC(CANCEL) 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'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. AL pertains to automated library. ML pertains to manual library.

Table 104. 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
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

Table 104. CBRXLACS FUNC(WTO) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
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. AL pertains to automated library. ML pertains to manual library.

Table 105. 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
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

Table 105. CBRXLACS FUNC(DOM) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
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. AL pertains to automated library. ML pertains to manual library.

Table 106. 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
		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

Table 106. CBRXLACS FUNC(ERRTEXT) Return and Reason Codes (continued)

Return Code	Error Type	Reason Code	Description	AL	ML	Message
		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 Table 107 on page 332, Table 108 on page 333, and Table 109 on page 335 provide mapping of the OBR.

The mapping of the 3995 library long OBR follows.

Table 107. 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.

Table 107. 3995 Library Long OBR Format (continued)

Offset Dec(Hex)	Name	Length	Description
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 108. 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.

Table 108. 3995 Optical Disk Drive Long OBR (continued)

Offset Dec(Hex)	Name	Length	Description
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 109. 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 that 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.
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.

Table 109. 3995 Library MDR (continued)

Offset Dec(Hex)	Name	Length	Description
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.
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 that 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

ASID[(*address*)]
Format the control block chain (012F) at the specified address.

BFCB[(*address*)]
Format the buffer manager control block (BFCB) at the specified address.

CBDUMP
Format a subset of the OAM control blocks (all of the following).

CBRMXCF[(*address*)]
Format the control block chain during a multiple OAM configuration at the specified address. An individual chain can also be formatted explicitly by using CBRMXCF(0B39E4A0).

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.

DB2[(*address*)]
Format the control block chain (DB2C) at the specified address.

DBCB[(*address*)]
Format the control block chain during a multiple OAM configuration at the specified address. An individual chain can also be formatted explicitly by using DBCB(0B39FA30).

DTCB[(*address*)]
Format the drive task control block (DTCB) at the specified address.

FTCB[(*address*)]
Format the file system task control block (FTCB) 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.

JOBN[(*address*)]
Format the control block chain (OAMC) 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.

OSMAX[(*address*)]
Format the control block chain during a multiple OAM configuration at the specified address. An individual chain can also be formatted explicitly by using OSMAX(0B39F770).

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.

RCB[(*address*)]
Format the recycle master control block (RCB) at the specified address.

RVCB[(*address*)]
Format the recycle volume control block (RVCB) 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.

TCA[(*address*)]
Format the component trace control block (TCA) 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.

XMCB[(*address*)]
Format the XCF member control block (XMCB) at the specified address.

XTCB[(*address*)]
Format the XCF task control block (XTCB) at the specified address.

Note: When specifying control blocks 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.

In a multiple OAM configuration, the default OAM ASID will be the Tape Library ASID if it exists. To view OAM control blocks (MCB, WTD, VCB, etc) for other Object OAM address spaces that might exist in the dump, use of an ASID, JOBN, or DB2 parameter on the VERBX OAMDATA command is required. Existing OVT and OSMAX chains will always be displayed, however, one of the ASID, JOBN, or DB2 parameters is required to display control blocks for a specific Object OAM address space. ASID, JOBN, and DB2 are mutually exclusive. For OSREQ related control blocks, the VERBX OAMDATA command will continue to search every address space in the dump for OMCBs and if found, try to display them and other associated OSREQ control blocks such as MWAs and RWAs. For OSREQ, the ASID, JOBN, or DB2 parameter can be used to show informational and error messages for the requested address space.

Example:

```
VERBX OAMDATA 'ASID(012F)'
```

or

```
VERBX OAMDATA 'JOBN(OAMC)'
```

or

```
VERBX OAMDATA 'DB2(DB2C)'
```

The ASID parameter can also be used to format OPL or other OAM control blocks that may exist in a non-OAM user address space.

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 OAM 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 CBR0*xxx*I is -904, and the type and reason codes in DSNT501I are 200 and 00C90081 respectively, the table space corresponding to the table in CBR0*xxx*I 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 that 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 OAM 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.

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, refer to DB2 for z/OS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPEK/db2z_prodhome.html).

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, refer to DB2 for z/OS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPEK/db2z_prodhome.html).

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 110 on page 341.

Table 110. 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 111 on page 342.

Table 111. 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 112 on page 342.

Table 112. 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.

Table 112. SCSI Adapter Completion Code (continued)

Completion Code Contents	Description
82	Adapter detected bad SCSI terminator power.
83	Adapter detected SCSI bus timeout.
84	Adapter detected timeout.

Chapter 14. 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.

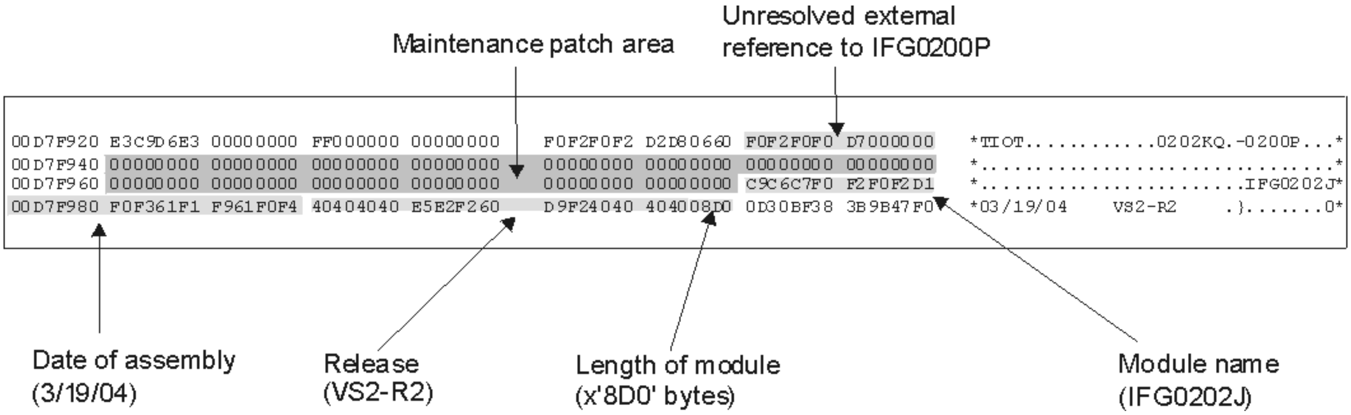
Note: This chapter discusses only selected control blocks. For more control blocks, with programming interface information, refer to *z/OS DFSMS Macro Instructions for Data Sets*.

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 32 on page 345 shows the module name of module IFG0202J. The object code **precedes** this module name.



```
00D7F920 E3C9D6E3 00000000 FF000000 00000000   F0F2F0F2 D2D80660 F0F2F0F0 D7000000   *TIOT...........0202KQ.-0200P...*
00D7F940 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *...............................*
00D7F960 00000000 00000000 00000000 00000000   00000000 00000000 C9C6C7F0 F2F0F2D1   *.......................IFG0202J*
00D7F980 F0F361F1 F961F0F4 40404040 E5E2F260   D9F24040 404008D0 0D30BF38 3B9B47F0   *03/19/04    VS2-R2    .}.......0*
```

Maintenance patch area
Unresolved external reference to IFG0200P
Date of assembly (3/19/04)
Release (VS2-R2)
Length of module (x'8D0' bytes)
Module name (IFG0202J)

Figure 32. Storage Dump Showing the Transfer Control Table for Module IFG0202J

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.

Note:

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	Type	Description
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 33 on page 347 and Figure 36 on page 356.

```
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
                         00000000    00000000   00000000   00800000   00000000
                         DCBUFND     DCBBUFCR   DCBBUFCR   DCBDSORG   DCB1OBAD    DCHLARC    DCBEDDAD    DCBRECFM
DCBLXLST    DCBT1OT
                         00          0000001    0000       4000       00000001    00         004859C     80
000000      0054
                         DCBMACP6    DCBIFIGS   DCBDEBAD   DC80FLGS
                         5000        00         063DE0     C3
                         ACCESS METHOD INTERFACE
                         0050000     00000001   00000050   00000000   00000000    00000001   00000050    00000001
00000050   00000001
                         000000      00000001
                   TIME 80738.51586C
  USRFF FFF WKAREA 1     VOL LBL. FILE LRD. DSCBS. CR MSG AREA        +10         +14        +18         +1C
+2C        +24
                         000000010   00063DD8   8CCAB73C   4000A3AA   00063DE0    8004B514   00063FC8    00048518
00063FF0   0001DF0C
                         +28         +2C        +30        +34        +38         +3C        +40         +44
+48        +4C
                         0001EC8     0000BSC8   0000006    0000010    0000008     0000000    0000000     0000000
0000000    0000000
                         +50         +54        +58        +5C        +60
                         0000A4EE    00063680   00C63DEQ   20FA0000   0000000

---------------------------------------------------------------------------------------------------------------
            TIME 80738.702779
  *** DATE DAY 274 YEAR 1973  TIME
01.22.18                                                                                           ***

USRFF FFF JFCBDSNM
                    D86CC9D3 40404040  C5E74040  40404040  40404040  40404040  40404040  40404040  40404040  40404040
40404040
                    40404040
            TIME   80739.393241

---------------------------------------------------------------------------------------------------------------
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 33. 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

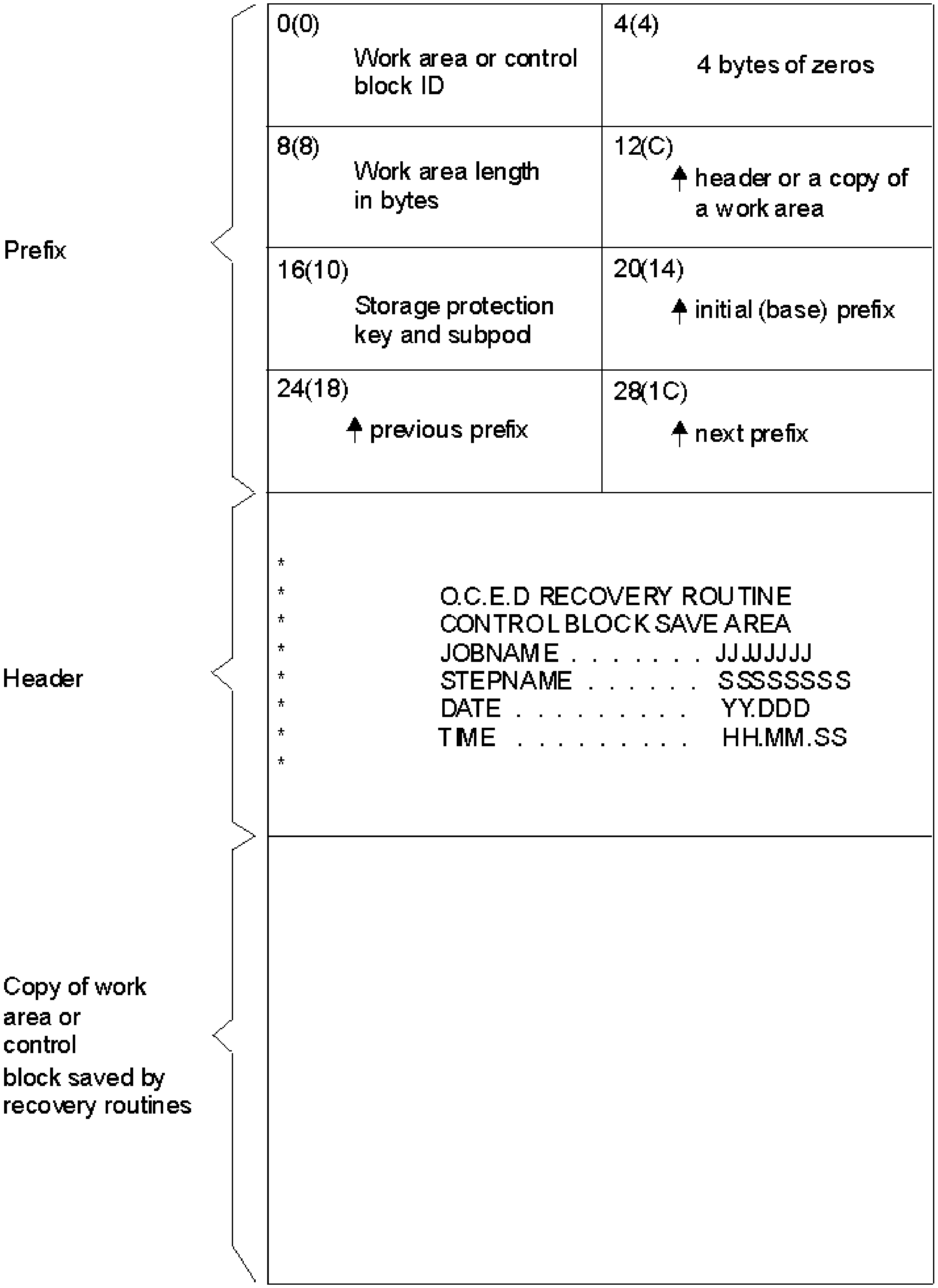
The O/C/EOV system recovery environment is established through the ESTAE macro, issued when storage was obtained during initialization of the O/C/EOV function. The O/C/EOV system recovery routines receive control from R/TM when any abend condition (determinate or indeterminate) is encountered during 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.

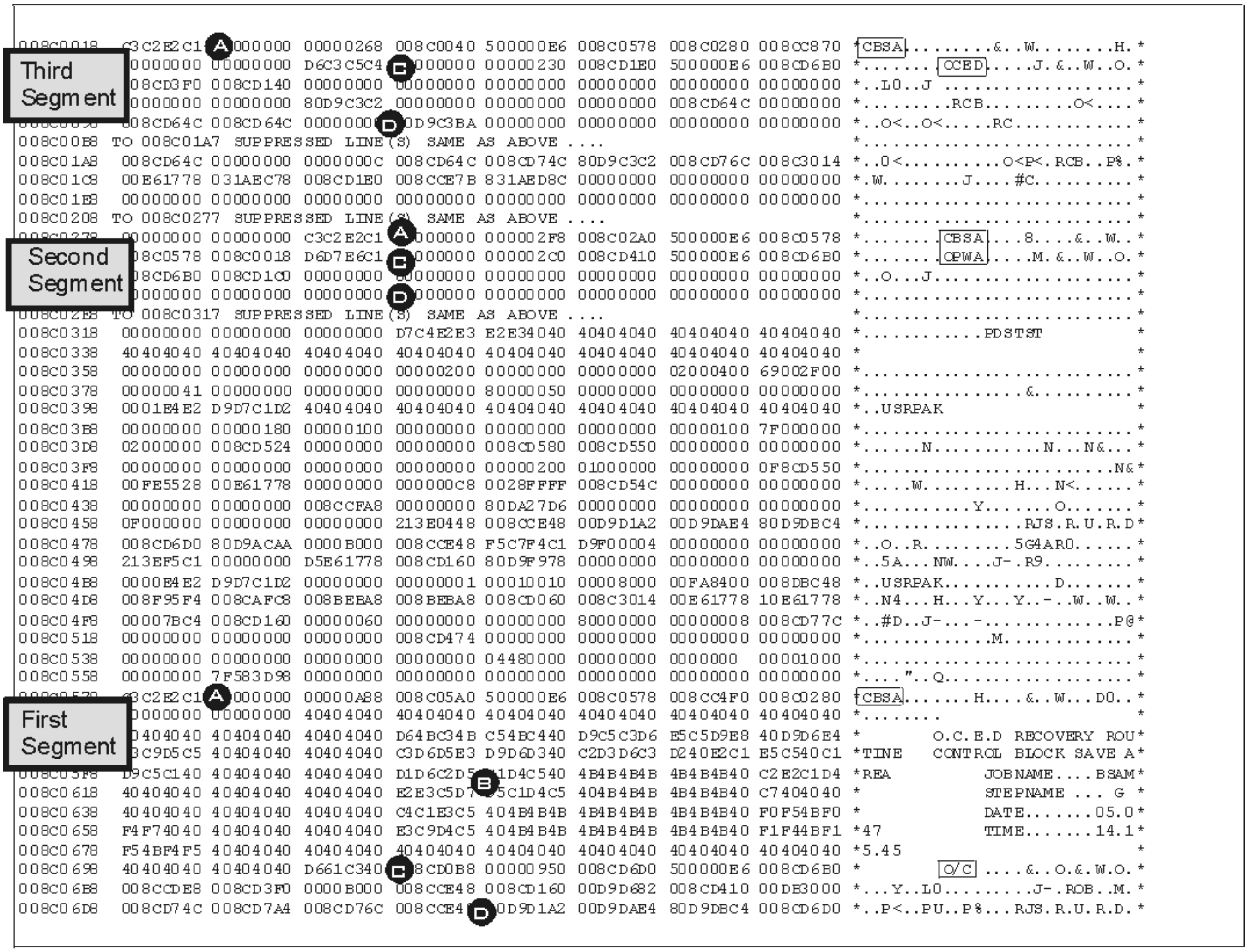
When O/C/EOV recovery processing has been completed, the RRCBSA is used to refresh the O/C/EOV chained work areas, which may have been modified during recovery processing. The RRCBSA is then freed. Note that an RRCBSA will not appear in a dump of the original error, but it will appear in a dump of an error that occurs during recovery processing.

The contents of the RRCBSA are defined in Figure 34 on page 349. An example of a dump of this save area is shown in Figure 35 on page 350.


0(0)
Work area or control block ID
4(4)
4 bytes of zeros
8(8)
Work area length in bytes
12(C)
↑ header or a copy of a work area
16(10)
Storage protection key and subpod
20(14)
↑ initial (base) prefix
24(18)
↑ previous prefix
28(1C)
↑ next prefix
Prefix
Header
Copy of work area or control block saved by recovery routines

```
*
*          O.C.E.D RECOVERY ROUTINE
*          CONTROL BLOCK SAVE AREA
*          JOBNAME . . . . . . . JJJJJJJJ
*          STEPNAME . . . . . .  SSSSSSSS
*          DATE . . . . . . . . .  YY.DDD
*          TIME  . . . . . . . . .  HH.MM.SS
*
```

Figure 34. Contents of Recovery Routine Control Block Save Area (RRCBSA)



```
008C0018  C3C2E2C1 00000000 00000268 008C0040 500000E6 008C0578 008C0280 008CC870 *CBSA.........&..W........H.*
          00000000 00000000 D6C3C5C4 00000000 00000230 008CD1E0 500000E6 008CD6B0 *........CCED.....J.&..W..O.*
          008CD3F0 008CD140 00000000 00000000 00000000 00000000 00000000 00000000 *..L0..J ...................*
          00000000 00000000 80D9C3C2 00000000 00000000 00000000 008CD64C 00000000 *.........RCB.........O<....*
          008CD64C 008CD64C 00000000 00D9C3BA 00000000 00000000 00000000 00000000 *..O<..O<.....RC............*
008C00B8  TO 008C01A7  SUPPRESSED LINE(S)  SAME AS ABOVE ....                   *...........................*
008C01A8  008CD64C 00000000 0000000C 008CD64C 008CD74C 80D9C3C2 008CD76C 008C3014 *..0<..........O<P<.RCB..P%.*
008C01C8  00E61778 031AEC78 008CD1E0 008CCE7B 831AED8C 00000000 00000000 00000000 *.W........J....#C..........*
008C01E8  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *...........................*
008C0208  TO 008C0277  SUPPRESSED LINE(S)  SAME AS ABOVE ....                   *...........................*
          00000000 00000000 C3C2E2C1 00000000 000002F8 008C02A0 500000E6 008C0578 *........CBSA....8....&..W..*
          008C0578 008C0018 D6D7E6C1 00000000 000002C0 008CD410 500000E6 008CD6B0 *........OPWA.....M.&..W..O.*
          008CD6B0 008CD1C0 00000000 00000000 00000000 00000000 00000000 00000000 *..O...J....................*
          00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *...........................*
          TO 008C0317  SUPPRESSED LINE(S)  SAME AS ABOVE ....                   *...........................*
008C0318  00000000 00000000 00000000 D7C4E2E3 E2E34040 40404040 40404040 40404040 *............PDSTST         *
008C0338  40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *                           *
008C0358  00000000 00000000 00000000 00000200 00000000 00000000 02000400 69002F00 *...........................*
008C0378  00000041 00000000 00000000 00000000 80000050 00000000 00000000 00000000 *................&..........*
008C0398  0001E4E2 D9D7C1D2 40404040 40404040 40404040 40404040 40404040 40404040 *..USRPAK                   *
008C03B8  00000000 00000180 00000100 00000000 00000000 00000000 00000100 7F000000 *...........................*
008C03D8  02000000 008CD524 00000000 00000000 008CD580 008CD550 00000000 00000000 *......N...........N...N&...*
008C03F8  00000000 00000000 00000000 00000000 00000200 01000000 00000000 0F8CD550 *.........................N&*
008C0418  00FE5528 00E61778 00000000 000000C8 0028FFFF 008CD54C 00000000 00000000 *.....W.........H...N<......*
008C0438  00000000 00000000 008CCFA8 00000000 80DA27D6 00000000 00000000 00000000 *...........Y.......O.......*
008C0458  0F000000 00000000 00000000 213E0448 008CCE48 00D9D1A2 00D9DAE4 80D9DBC4 *................RJS.R.U.R.D*
008C0478  008CD6D0 80D9ACAA 0000B000 008CCE48 F5C7F4C1 D9F00004 00000000 00000000 *..O..R.........5G4AR0......*
008C0498  213EF5C1 00000000 D5E61778 008CD160 80D9F978 00000000 00000000 00000000 *..5A...NW....J-.R9.........*
008C04B8  0000E4E2 D9D7C1D2 00000000 00000001 00010010 00008000 00FA8400 008DBC48 *..USRPAK...........D.......*
008C04D8  008F95F4 008CAFC8 008BEBA8 008BEBA8 008CD060 008C3014 00E61778 10E61778 *..N4...H...Y...Y..-..W..W..*
008C04F8  00007BC4 008CD160 00000060 00000000 00000000 80000000 00000008 008CD77C *..#D..J-....-.............P@*
008C0518  00000000 00000000 00000000 008CD474 00000000 00000000 00000000 00000000 *.............M.............*
008C0538  00000000 00000000 00000000 00000000 04480000 00000000 0000000  00001000 *...........................*
008C0558  00000000 7F583D98 00000000 00000000 00000000 00000000 00000000 00000000 *...."..Q...................*
          C3C2E2C1 00000000 00000A88 008C05A0 500000E6 008C0578 008CC4F0 008C0280 *CBSA.......H....&..W...D0..*
          00000000 00000000 40404040 40404040 40404040 40404040 40404040 40404040 *........                   *
          40404040 40404040 40404040 D64BC34B C54BC440 D9C5C3D6 E5C5D9E8 40D9D6E4 *        O.C.E.D RECOVERY ROU*
          E3C9D5C5 40404040 40404040 C3D6D5E3 D9D6D340 C2D3D6C3 D240E2C1 E5C540C1 *TINE    CONTROL BLOCK SAVE A*
          D9C5C140 40404040 40404040 D1D6C2D5 C1D4C540 4B4B4B4B 4B4B4B40 C2E2C1D4 *REA            JOBNAME....BSAM*
008C0618  40404040 40404040 40404040 E2E3C5D7 D5C1D4C5 404B4B4B 4B4B4B40 C7404040 *               STEPNAME ... G *
008C0638  40404040 40404040 40404040 C4C1E3C5 404B4B4B 4B4B4B4B 4B4B4B40 F0F54BF0 *               DATE.......05.0*
008C0658  F4F74040 40404040 40404040 E3C9D4C5 404B4B4B 4B4B4B4B 4B4B4B40 F1F44BF1 *47             TIME.......14.1*
008C0678  F54BF4F5 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *5.45                       *
008C0698  40404040 40404040 D661C340 008CD0B8 00000950 008CD6D0 500000E6 008CD6B0 *        O/C ....&..O.&.W.O.*
008C06B8  008CCDE8 008CD3F0 0000B000 008CCE48 008CD160 00D9D682 008CD410 00DB3000 *...Y..L0.........J-.ROB..M.*
008C06D8  008CD74C 008CD7A4 008CD76C 008CCE48 00D9D1A2 00D9DAE4 80D9DBC4 008CD6D0 *..P<..PU..P%...RJS.R.U.R.D.*
```

A Prefixes to the RRCBSA segments

B RRCBSA header

C Prefix included in the copy of the work area or control block.

D Copy of 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).

Figure 35. 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 35 on page 350, 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”. 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 34 on page 349.

Abend error recording in the logrec data set

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 113 on page 351.

Table 113. 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.
	1.11 .111	RRFEXTND	Extend.
	1..1 .1..	RRFPRLSE	Partial release.
	11.. 111.	RRFLSPAC	LSPACE.
	1..1 1.11	RRFOBTN	Obtain.

Table 113. O/C/EOV Recovery Routine Parameter List (continued)

Offset Dec(Hex)	Length	Name	Description
	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.
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

If you suspect an error in the OPEN/CLOSE/EOV component and you can recreate the problem, you can create an optional work area trace to help with diagnosis. The work area trace traces the DCB and the O/C/EOV work area and extended work area associated with the DCB. It 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.

To request the optional work area trace, use one of these methods:

- Specify DCB=DIAGNS=TRACE on the DD statement of a data set for which you wish to trace
- Specify the DALDIAGN text unit with dynamic allocation for a data set for which you wish to trace
- Use the problem determination optional work area trace started task, IFGOCETR. Refer to "Non-VSAM problem determination work area trace started task" on page 353 for information on the started task. Each time that IFG019RA is entered, it writes the work area and control blocks to the GTF data set. No tracing is done when control is passed to the O/C/EOV problem determination and recovery routines or the initial and final modules of each function of O/C/EOV. Tracing also occurs during access method executor processing for the SAM, BDAM create and BPAM executors.

  If you start GTF, use the TRACE=USR option. Refer to "Abend error recording initiated by O/C/EOV" on page 346 for more information.

Any of these methods of requesting an optional work area trace causes the BAM component to create an internal BAM trace table. This tracing does not require GTF. Tracing is in a wrap-around trace table. It does not greatly increase execution time and does not produce a great deal of output. Refer to "BAM internal trace facility" on page 90 for more information.

Figure 36 on page 356 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 36 on page 356 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 sets for which tracing is desired. Any combination of these specifications is allowed, but the more specific the information provided, the more selective O/C/EOV will be in determining which data sets 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" on page 354 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.

Note: If the data sets being traced are being opened by JES2 or JES3 as part of the start of these tasks, you must start IFGOCETR with SUB=MSTR,TIME=60 in order for the started task to be started.

Work area trace started task messages

IEC980A SPECIFY TRACE KEYS: DSN= ,DDN= ,JN= ,DISPLAY ,DELETE, ALL|NUM END

The keys are:

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 that 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 that 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. The value can include a wildcard character, *, which stands for any character or string of characters. So, for example, JN=abc* specifies any job name that begins with abc.

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

where:

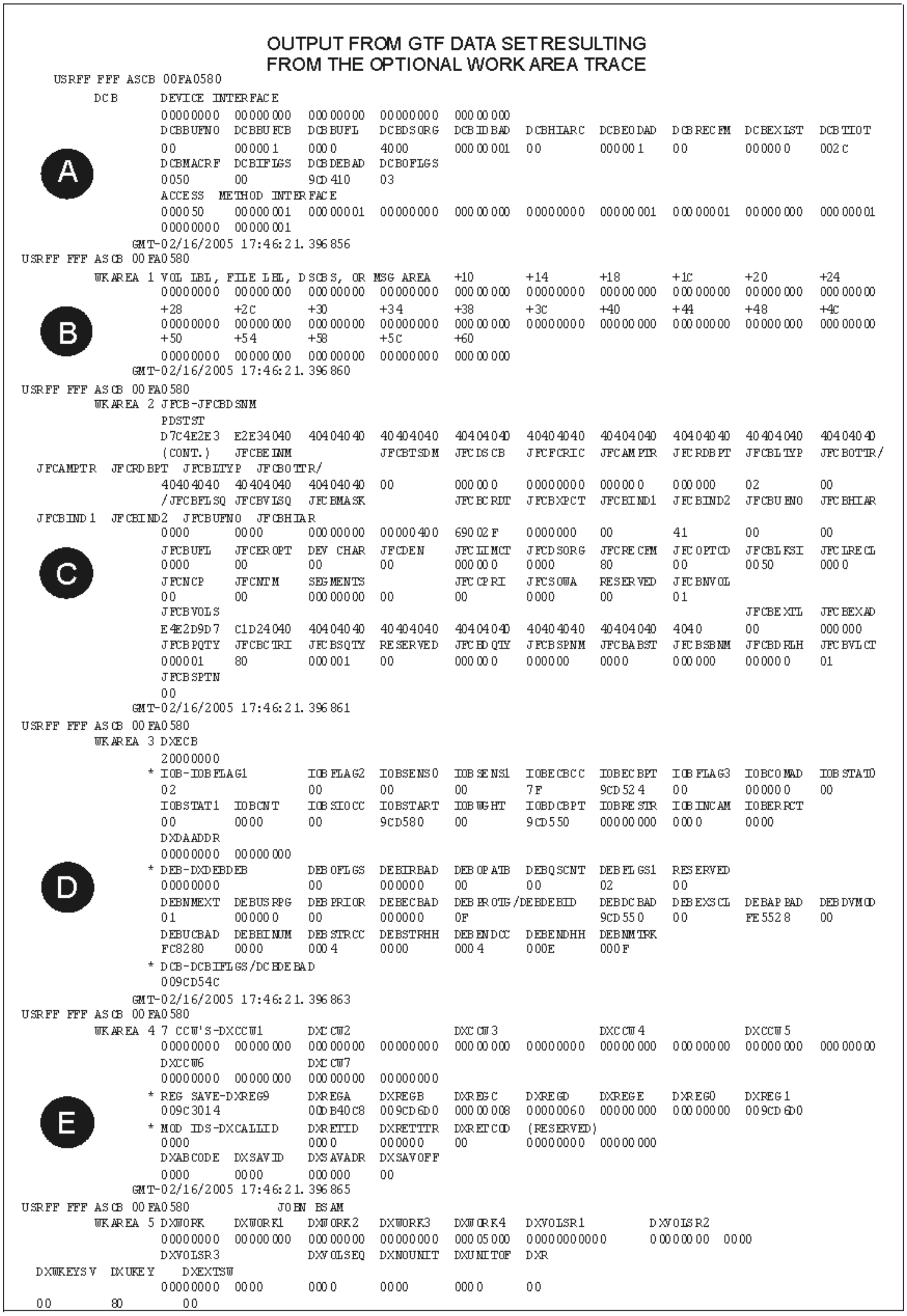
INVALID REPLY
A key specified in the response to message IEC980A is invalid. Message IEC980A will be reissued so the correct key can be specified.

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. If you are interested in starting the BAM internal trace and not the OPEN/CLOSE/EOV work area trace, then you do not have to start GTF.



```
                                    OUTPUT FROM GTF DATA SET RESULTING
                                    FROM THE OPTIONAL WORK AREA TRACE
    USRFF FFF ASCB 00FA0580
          DCB      DEVICE INTERFACE
                   00000000  00000000  00000000  00000000  00000000
                   DCBBUFNO  DCBBUFCB  DCBBUFL   DCBDSORG  DCBIOBAD  DCBHIARC  DCBEODAD  DCBRECFM  DCBEXIST  DCBTIOT
                   00        000001    0000      4000      00000001  00        000001    00        000000    002C
                   DCBMACRF  DCBIFLGS  DCBDEBAD  DCBOFLGS
                   0050      00        9CD410    03
                   ACCESS  METHOD INTERFACE
                   00050     00000001  00000001  00000000  00000000  00000000  00000001  00000001  00000000  00000001
                   00000000  00000001
               GMT-02/16/2005 17:46:21.396856
USRFF FFF ASCB 00FA0580
          WKAREA 1 VOL LBL, FILE LBL, DSCBS, OR MSG AREA   +10       +14       +18       +1C       +20       +24
                   00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000
                   +28       +2C       +30       +34       +38       +3C       +40       +44       +48       +4C
                   00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000
                   +50       +54       +58       +5C       +60
                   00000000  00000000  00000000  00000000  00000000
               GMT-02/16/2005 17:46:21.396860
USRFF FFF ASCB 00FA0580
          WKAREA 2 JFCB-JFCBDSNM
                   PDSTST
                   D7C4E2E3  E2E34040  40404040  40404040  40404040  40404040  40404040  40404040  40404040  40404040
                   (CONT.)   JFCBEINM            JFCBTSDM  JFCDSCB   JFCFCRIC  JFCAMPTR  JFCRDBPT  JFCBLTYP  JFCBOTTR/
  JFCAMPTR  JFCRDBPT  JFCBLTYP  JFCBOTTR/
                   40404040  40404040  40404040  00        00000     00000000  000000    000000    02        00
                   /JFCBFLSQ JFCBVISQ  JFCBMASK            JFCBCRDT  JFCBXPCT  JFCBIND1  JFCBIND2  JFCBUFNO  JFCBHIAR
  JFCBIND1  JFCBIND2  JFCBUFNO  JFCBHIAR
                   0000      0000      00000000  00000400  69002F    0000000   00        41        00        00
                   JFCBUFL   JFCEROPT  DEV CHAR  JFCDEN    JFCLIMCT  JFCDSORG  JFCRECFM  JFCOPTCD  JFCBLKSI  JFCLRECL
                   0000      00        00        00        00000     0000      80        00        0050      0000
                   JFCNCP    JFCNTM    SEGMENTS            JFCCPRI   JFCSOWA   RESERVED  JFCBNVOL
                   00        00        00000000  00        00        0000      00        01
                   JFCBVOLS                                                                        JFCBEXTL  JFCBEXAD
                   E4E2D9D7  C1D24040  40404040  40404040  40404040  40404040  40404040  4040      00        000000
                   JFCBPQTY  JFCBCTRI  JFCBSQTY  RESERVED  JFCBDQTY  JFCBSPNM  JFCBABST  JFCBSBNM  JFCBDRLH  JFCBVLCT
                   000001    80        000001    00        00000     000000    0000      000000    000000    01
                   JFCBSPTN
                   00
               GMT-02/16/2005 17:46:21.396861
USRFF FFF ASCB 00FA0580
          WKAREA 3 DXECB
                   20000000
                 * IOB-IOBFLAG1        IOBFLAG2  IOBSENS0  IOBSENS1  IOBECBCC  IOBECBPT  IOBFLAG3  IOBCOMAD  IOBSTAT0
                   02                  00        00        00        7F        9CD524    00        000000    00
                   IOBSTAT1  IOBCNT    IOBSIOCC  IOBSTART  IOBWGHT   IOBDCBPT  IOBRESTR  IOBINCAM  IOBERRCT
                   00        0000      00        9CD580    00        9CD550    00000000  0000      0000
                   DXDAADDR
                   00000000  00000000
                 * DEB-DXDEBDEB        DEBOFLGS  DEBIRBAD  DEBOPATB  DEBQSCNT  DEBFLGS1  RESERVED
                   00000000            00        000000    00        00        02        00
                   DEBNMEXT  DEBUSRPG  DEBPRIOR  DEBECBAD  DEBPROTG/DEBDEBID   DEBDCBAD  DEBEXSCL  DEBAPPAD  DEBDVMOD
                   01        000000    00        000000    0F                  9CD550    00        FE5528    00
                   DEBUCBAD  DEBBINUM  DEBSTRCC  DEBSTRHH  DEBENDCC  DEBENDHH  DEBNMTRK
                   FC8280    0000      0004      0000      0004      000E      000F
                 * DCB-DCBIFLGS/DCBDEBAD
                   009CD54C
               GMT-02/16/2005 17:46:21.396863
USRFF FFF ASCB 00FA0580
          WKAREA 4 7 CCW'S-DXCCW1      DXCCW2              DXCCW3              DXCCW4              DXCCW5
                   00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000
                   DXCCW6              DXCCW7
                   00000000  00000000  00000000  00000000
                 * REG SAVE-DXREG9     DXREGA    DXREGB    DXREGC    DXREGD    DXREGE    DXREG0    DXREG1
                   009C3014            00DB40C8  009CD6D0  00000008  00000060  00000000  00000000  009CD6D0
                 * MOD IDS-DXCALLID    DXRETID   DXRETTTR  DXRETCOD  (RESERVED)
                   0000                0000      000000    00        00000000  00000000
                   DXABCODE  DXSAVID   DXSAVADR  DXSAVOFF
                   0000      0000      000000    00
               GMT-02/16/2005 17:46:21.396865
USRFF FFF ASCB 00FA0580            JOBN BSAM
          WKAREA 5 DXWORK    DXWORK1   DXWORK2   DXWORK3   DXWORK4   DXVOLSR1        DXVOLSR2
                   00000000  00000000  00000000  00000000  00005000  000000000000    00000000  0000
                   DXVOLSR3            DXVOLSEQ  DXNOUNIT  DXUNITOF  DXR
  DXWKEYSV  DXUKEY    DXEXTSW
                   00000000  0000      0000      0000      0000      00
  00        80        00
```

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 36. 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"

Offset	Type/Value	Len	Name (Dim)	Description
	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)
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				
	0010 0001		DCBDV311	"X'21'"– 2311 DISK STORAGE
	0010 0010		DCBDV301	"X'22'"– 2301 PARALLEL DRUM
	0010 0011		DCBDV303	"X'23'"– 2303 SERIAL DRUM
	0010 0100		DCBDV345	"X'24'"– 9345 DISK STORAGE FACILITY
	0010 0101		DCBDV321	"X'25'"– 2321 DATA CELL STORAGE
	0010 0110		DCBD1305	"X'26'"– 2305 DRUM MODEL-1
	0010 0111		DCBDV305	"X'27'"– 2305 DRUM MODEL-2
	0010 1000		DCBDV314	"X'28'"– 2314/2319 DISK STORAGE FACILITY
	0010 1001		DCBDV330	"X'29'"– 3330 DISK STORAGE FACILITY 3330 MODEL-1 3330 MODEL-2 3333 MODEL-1
	0010 1010		DCBDV340	"X'2A'"– 3340/3344 DISK STORAGE FACILITY
	0010 1011		DCBDV350	"X'2B'"– 3350 DISK STORAGE FACILITY MODELS A2, B2, AND C2
	0010 1100		DCBDV375	"X'2C'"– 3375 DISK STORAGE FACILITY
	0010 1101		DCBDV331	"X'2D'"– 3330 MODEL–11 OR 3333 MODEL–11 DISK STORAGE FACILITY
	0010 1110		DCBDV380	"X'2E'"– 3380 DISK STORAGE FACILITY
	0010 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				

Offset	Type/Value	Len	Name (Dim)	Description
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
	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 or 3590 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

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . .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
	.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

Offset	Type/Value	Len	Name (Dim)	Description
	.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.
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

Offset	Type/Value	Len	Name (Dim)	Description
	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
	. . . . . .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

Offset	Type/Value	Len	Name (Dim)	Description
	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
	. . . . 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

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	RESERVED
	. .1.		DCBDSGTQ	RESERVED
	. . . . 1. . .		DCBACBM	"DCBBIT4"– ACCESS METHOD CONTROL BLOCK
	. . . . .1. .		DCBDSGTR	RESERVED
28 (1C)	ADDRESS	4	DCBIOBAD(0)	ADDRESS OF IOB, SET BY ACCESS METHOD FOR CHAINED SCHEDULING OR OPTIONALLY BY USER WHEN EXCP IS USED
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
	1.		DCBH1	"DCBBIT0"– HIERARCHY 1 MAIN STORAGE IF BIT 5 IS ZERO. DCBE EXISTS IF BIT 5 IS ON.

Offset	Type/Value	Len	Name (Dim)	Description
				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
	.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

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . 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
	. . . . 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

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . 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)
	.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)
	. . .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) –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)

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . 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	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

Offset	Type/Value	Len	Name (Dim)	Description
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
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)	RESERVED
60 (3C)	BITSTRING	1	DCBCIND1	– CONDITION INDICATORS

Offset	Type/Value	Len	Name (Dim)	Description
	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)
	1.		DCBSTQCK	RESERVED
	.1.		DCBSTFLS	RESERVED
	.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.
68 (44)	BITSTRING	1		– DCBNCP (BSAM, BPAM)
69 (45)	ADDRESS	3	DCBIOBB	– SAME AS DCBIOBA ABOVE
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

Offset	Type/Value	Len	Name (Dim)	Description
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 READ OR WRITE REQUESTS THAT MAY BE ISSUED PRIOR TO ISSUING A CHECK MACRO
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
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

Offset	Type/Value	Len	Name (Dim)	Description
	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	.	– 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

Name	Hex Offset	Hex Value	Level
DCBBLKSI	3E		2
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

Name	Hex Offset	Hex Value	Level
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
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

Name	Hex Offset	Hex Value	Level
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
DCBEOEA	3C		2
DCBEOPFG	6	80	2

Name	Hex Offset	Hex Value	Level
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

Name	Hex Offset	Hex Value	Level
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
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

Name	Hex Offset	Hex Value	Level
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
DCBMRGET	32	40	2
DCBMRGTQ	33	40	2
DCBMRIDW	33	8	2
DCBMRIND	C		2
DCBMRLCG	32	8	2

Name	Hex Offset	Hex Value	Level
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

Name	Hex Offset	Hex Value	Level
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
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

Name	Hex Offset	Hex Value	Level
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
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

Name	Hex Offset	Hex Value	Level
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

Name	Hex Offset	Hex Value	Level
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
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

Name	Hex Offset	Hex Value	Level
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:
The system no longer supports the indexed sequential access method (ISAM). Instead, VSAM emulates ISAM. This data control block (DCB) is used by the VSAM routines that emulate ISAM and holds data pertinent to the use of a data set that is maintained by these routines. VSAM does not maintain all of the fields in this DCB. 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"

Offset	Type/Value	Len	Name (Dim)	Description
	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
	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	RESERVED
	. .1.		DCBDSGTQ	RESERVED
	. . . . 1. . .		DCBACBM	"DCBBIT4"– ACCESS METHOD CONTROL BLOCK
	. . . . .1. .		DCBDSGTR	RESERVED
28 (1C)	ADDRESS	4	DCBIOBAD(0)	ADDRRESS OF IOB.
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

Offset	Type/Value	Len	Name (Dim)	Description
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
	. .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

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . 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 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

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . 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)
	.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) – 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)

Offset	Type/Value	Len	Name (Dim)	Description
	. . .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
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				

Offset	Type/Value	Len	Name (Dim)	Description
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
	. .1.		DCBEXNSP	"DCBBIT2"– SPACE NOT FOUND
	. . .1		DCBEXINV	"DCBBIT3"– INVALID REQUEST
	. . . . 1. . .		DCBEXIER	"DCBBIT4"– UNCORRECTABLE INPUT ERROR
	. . . . .1. .		DCBEXOER	"DCBBIT5"– UNCORRECTABLE OUTPUT ERROR

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . . .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'.
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

Offset	Type/Value	Len	Name (Dim)	Description
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
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

Offset	Type/Value	Len	Name (Dim)	Description
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
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

Name	Hex Offset	Hex Value	Level
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
DCBESETL	54		2
DCBEX	31	40	2
DCBEXBLI	50	2	2
DCBEXBLU	50	1	2
DCBEXCD1	50		2

Name	Hex Offset	Hex Value	Level
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

Name	Hex Offset	Hex Value	Level
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
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

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

Name	Hex Offset	Hex Value	Level
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
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

Name	Hex Offset	Hex Value	Level
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
DCBWKPT1	E4		2
DCBWKPT2	E8		2
DCBWKPT3	EC		2
DCBWKPT4	F0		2

Name	Hex Offset	Hex Value	Level
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"

Offset	Type/Value	Len	Name (Dim)	Description
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	–BYTE OF DCBDSORG
	1.		DCBDSGGS	"DCBBIT0"– GS – GRAPHICS ORGANIZATION
	.1.		DCBDSGTX	RESERVED
	. .1.		DCBDSGTQ	RESERVED
	. . . . 1. . .		DCBACBM	"DCBBIT4"– ACCESS METHOD CONTROL BLOCK
	. . . . .1. .		DCBDSGTR	RESERVED
28 (1C)	ADDRESS	4	DCBIOBAD (0)	ADDRRESS OF IOB. SET WITH CHAINED SCHEDULING OR OPTIONALLY WHEN EXCP IS USED.
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

Offset	Type/Value	Len	Name (Dim)	Description
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
	. .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

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . 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
	. . . . 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

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . 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)
	.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) – 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)

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . 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"– 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)

Offset	Type/Value	Len	Name (Dim)	Description
	.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 IOB 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
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

Name	Hex Offset	Hex Value	Level
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
DCBDSGPS	1A	40	2

Name	Hex Offset	Hex Value	Level
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

Name	Hex Offset	Hex Value	Level
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
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

Name	Hex Offset	Hex Value	Level
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
DCBOFOPN	30	10	2
DCBOFPPC	30	8	2
DCBOFTM	30	4	2
DCBOFUEX	30	2	2

Name	Hex Offset	Hex Value	Level
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

Name	Hex Offset	Hex Value	Level
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"

Offset	Type/Value	Len	Name (Dim)	Description
	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
	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	RESERVED
	. .1.		DCBDSGTQ	RESERVED
	. . . . 1. . .		DCBACBM	"DCBBIT4"– ACCESS METHOD CONTROL BLOCK
	. . . . .1. .		DCBDSGTR	RESERVED
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

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
	.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)	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

Offset	Type/Value	Len	Name (Dim)	Description
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.
	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

Offset	Type/Value	Len	Name (Dim)	Description
	. . .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)
	.1.		DCBMRFE	"DCBBIT1"– FOUNDATION EXTENSION IS PRESENT (EXCP)
	.1.		DCBMRGET	"DCBBIT1"– GET (QSAM, QISAM)
	.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) - 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"– 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

Offset	Type/Value	Len	Name (Dim)	Description
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
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

Offset	Type/Value	Len	Name (Dim)	Description
	. .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.
	. . . . 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

Name	Hex Offset	Hex Value	Level
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
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

Name	Hex Offset	Hex Value	Level
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
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

Name	Hex Offset	Hex Value	Level
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

Name	Hex Offset	Hex Value	Level
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
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

Name	Hex Offset	Hex Value	Level
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
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

Name	Hex Offset	Hex Value	Level
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 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

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

Offset	Type/Value	Len	Name (Dim)	Description
	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	RESERVED
	. .1.		DCBDSGTQ	RESERVED
	. . . . 1. . .		DCBACBM	"DCBBIT4"– ACCESS METHOD CONTROL BLOCK
	. . . . .1. .		DCBDSGTR	RESERVED
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

Offset	Type/Value	Len	Name (Dim)	Description
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
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

Name	Hex Offset	Hex Value	Level
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
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

Name	Hex Offset	Hex Value	Level
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 or by application program

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.

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

Offset	Type/Value	Len	Name (Dim)	Description
	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.
	. . .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.

Offset	Type/Value	Len	Name (Dim)	Description
−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)	
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

Offset	Type/Value	Len	Name (Dim)	Description
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
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)

Offset	Type/Value	Len	Name (Dim)	Description
	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)
	. . . . . .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	IOBRESCT	RESIDUAL COUNT
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

Offset	Type/Value	Len	Name (Dim)	Description
	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.
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)

Offset	Type/Value	Len	Name (Dim)	Description
	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
	. . . . . .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

Offset	Type/Value	Len	Name (Dim)	Description
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.
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

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . 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
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

Offset	Type/Value	Len	Name (Dim)	Description
	.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
	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

Name	Hex Offset	Hex Value	Level
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
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

Name	Hex Offset	Hex Value	Level
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
IOBDSTAT	32		2
IOBDTYPE	30		2
IOBDTYP2	31		2
IOBDYNBF	30	4	2
IOBECBCC	4		2

Name	Hex Offset	Hex Value	Level
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

Name	Hex Offset	Hex Value	Level
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
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

Name	Hex Offset	Hex Value	Level
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
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

Name	Hex Offset	Hex Value	Level
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

Name	Hex Offset	Hex Value	Level
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
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

Name	Hex Offset	Hex Value	Level
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*.

Table 114. DEB fields

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	RESERVED
	 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
	.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

Table 114. DEB fields (continued)

Offset	Type/Value	Len	Name (Dim)	Description
	 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	RESERVED
	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
	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	RESERVED
–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 DEB ID FLAG
	. . . .		DEBTAMID	RESERVED
	 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

Table 114. DEB fields (continued)

Offset	Type/Value	Len	Name (Dim)	Description
	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
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

Table 114. DEB fields (continued)

Offset	Type/Value	Len	Name (Dim)	Description
	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.		DEBVCR	"X'40'" RLS=CR JCL OPTION
	. .1.		DEBVNRI	"X'20'" RLS=NRI JCL OPTION
	. . .1		DEBVCRE	"X'10'" RLS=NRI JCL OPTION
	. . . . 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.		DEB31UCB	ALL UCB ADDRESSES ARE FOUR BYTES AND THE DEVICE MODIFIER BYTE FOLLOWS EACH UCB ADDRESS.
	1		DEBTVS	"X'01'" TRANSACTIONAL VSAM
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

Table 114. DEB fields (continued)

Offset	Type/Value	Len	Name (Dim)	Description
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				
	Note: Address of a UCB associated with this extent. The following applies to DEBSUCBA, DEBSUCBB, DEBUCBAD and DEBUCBA: If the actual UCB is above the 16 MB line and the dynamic allocation nocapture option is not in effect, allocation normally captures the UCB to create a 24-bit address. When using EXCP and you specify the nocapture option of dynamic allocation but not the LOC=ANY option on the DCBE, then OPEN or EOV captures the UCB until a later EOV or close. In these cases the high order byte of this word contains the device modifier byte. If you specify nocapture on the dynamic allocation and the actual DASD or tape UCB address is above the line and the DCB is for BSAM, BPAM, QSAM or EXCP and the DCBE has specified LOC=ANY, then the system does not capture the UCB. In that case OPEN turns on the DEB31UCB bit to signify the 31-bit UCB address field is valid and that the device modifier byte is in DEBSDVMX or DEBDVMOD31. It may remain on for subsequent volumes even though they have actual 24-bit addresses in a four-byte field.			
32 (20)	ADDRESS	4	DEBSUCBA (0)	– ADDRESS OF A UCB ASSOCIATED WITH A GIVEN DATA SET. See Note.
32 (20)	BITSTRING	1	DEBSDVM	– DEVICE MODIFIER. FOR MAGNETIC TAPE, MODESET OPERATION CODE. FOR UNIT RECORD, NOT DEFINED. VALID ONLY IF DEB31UCB IS OFF.
	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

Table 114. DEB fields (continued)

Offset	Type/Value	Len	Name (Dim)	Description
	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. See Note.
36 (24)		0	DEBDEVED(0)	END OF COMMON UNIT RECORD AND UNIT RECORD FIELDS IF DEB31UCB IS OFF.
NEXT FOUR BYTES PRESENT ONLY IF DEB31UCB IS ON				
36 (24)	BITS	1	DEBSDVMX	DEVICE MODIFIER. FOR MAGNETIC TAPE, MODESET OPERATION CODE OR MODESET FUNCTION BYTE. FOR UNIT RECORD, RESERVED. PRESENT ONLY IF DEB31UCB IS ON.
37 (25)	CHARACTERS	3		RESERVED
40 (28)		0	DEBDVEDX	END OF SECTION IF DEB31UCB IS ON
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
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)

Table 114. DEB fields (continued)

Offset	Type/Value	Len	Name (Dim)	Description
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
	. .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	– ADDRESS OF UCB ASSOCIATED WITH THIS DATA EXTENT. SeeNote.

Table 114. DEB fields (continued)

Offset	Type/Value	Len	Name (Dim)	Description
0 (0)	BITSTRING	1	DEBDVMOD	– DEVICE MODIFIER – FILE MASK. VALID ONLY IF DEB31UCB IS OFF.
1 (1)	ADDRESS	3	DEBUCBA	– ADDRESS OF UCB ASSOCIATED WITH THIS DATA EXTENT. VALID ONLY IF DEB31UCB IS OFF.. SeeNote.
4 (4)	BINARY	1	DEBDVMOD31	RESERVED IF DEB31UCB IS OFF. FILE MASK (DEVICE MODIFIER) IF DEB31UCB IS ON.
5 (5)	BINARY	1	DEBNMTRKHI	High order byte of a three-byte number of tracks in the extent. Low order two bytes are in DEBNMTRK.
6 (6)	CHARACTER	2	DEBSTRCC	– CYLINDER ADDRESS FOR THE START OF AN EXTENT.
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 THIS 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
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)

Table 114. DEB fields (continued)

Offset	Type/Value	Len	Name (Dim)	Description
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
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

Table 114. DEB fields (continued)

Offset	Type/Value	Len	Name (Dim)	Description
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				
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

Table 114. DEB fields (continued)

Offset	Type/Value	Len	Name (Dim)	Description
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
	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		DEBXMVLF	"X'01'"TAPE MULTIVOL & 1ST VOL READ
4 (4)	ADDRESS	4	DEBXDSAB	– POINTER TO DSAB MDC004

Table 114. DEB fields (continued)

Offset	Type/Value	Len	Name (Dim)	Description
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.

Offset	Type/Value	Len	Name (Dim)	Description
40 (28)	BITSTRING	8	DEBXDEF (0)	DEFINE EXTENT DATA AREA
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

Table 114. DEB fields (continued)

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . .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		RESERVED
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		DEBXDUDA	"X'10'" IEC708I DUP VOL ISSUED FOR DASD
	. . . . 1 . . .		DEBXCAP	"X'08'"UCB ADDRESS IN DEB CAPTURED BY OCE

Table 114. DEB fields (continued)

Offset	Type/Value	Len	Name (Dim)	Description
	. . . . .1 . .		DEBXVTOC	"X'04'"NOT AUTHORIZED TO READ VTOC
	. . . . . . 1 .		DEBXENQE	"X'02'"ENQUEUED ON PS, DISP=SHARE, OUTPUT
	 1		DEBXUCNT	"X'01'"DECREMENT EDI USE COUNT IN ESTAE
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
	. . . 1		DEBXAIB	DEBXAMB DEFINES AN INTERMEDIATE BUFFER USED FOR READ PREVIOUS TAPE PROCESSING
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)
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

Table 114. DEB fields (continued)

Offset	Type/Value	Len	Name (Dim)	Description
	.1.....		DEB2XNLM	NL TAPE DATASET OPENED FOR MOD
	..1.....		DEB2XTCO	TCLOSE TAPE OUTPUT
	...1....		DEB2XIBC	INACCURATE BLOCK COUNT
	....1...		DEB2XNL1	NL TAPE FILE SEQ 1
	.....1..		DEB2XVLL	DEB2X INCLUDES DATASET VOLUME LIST
	......1.		DEB2XABC	ACCURATE TOTAL BLOCK COUNT
	1		DEB2XECU	3490EMULATION EXT CAPACITY WRITTEN TO
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)	BITSTRING	1	DEB2XFG2	SECOND FLAG BYTE
	1.......		DEB2X32M	OCE ISSUED 32BIT MODESET
	.1......		DEB2XBTM	BUFFED TM REQUESTED
	..1.....		DEB2XUSS	UNIX DIRECTORY OPENED USING BPAM OR PART OF BPAM CONCATENATION
	...1....		DEB2XEXV	DEBXMXV EXCEEDED
	....1...		DEB2XSCA	TAPE MEDIA SCALED OPT PERFORMANCE
	.....1..		DEB2BTMS	SYSTEM DEFAULTED BUFFERED TAPE MARKS
	......1.		DEB2XEMV	EXTEND MULTIVOL TAPE FILE
	1		DEB2X8WN	RACF RC8 CHANGED TO RC0
144 (90)	ADDRESS	4	DEB2XTIM	POINTER TO TIME STAMP
148 (94)	SIGNED	1	DEB2XEXC	DEB2XMXV EXCEED COUNT
149(95)	BITSTRING	1	DEB2XEDI	EDI SMF INDICATORS
	1.......		DEB2XEXT	DSN FOUND IN EXCLUDE TABLE
	.1......		DEB2XOPO	O/P AND ALREADY OPEN O/P
	..1.....		DEB2XINO	I/P AND ALREADY OPEN O/P
	...1....		DEB2XEPS	DCBE, SCT OR DSAB EXCLUDED
150(96)	BITSTRING	1	DEB2XFG3	THIRD FLAG BYTE
	1.......		DEB2XSCS	SYSZEDI SCOPE SYSTEM(S)
	.1......		DEB2XPSC	PERFORMANCE SEGMENTED
	..1.....		DEB2XEF1	3592-E05 WRITING IN EFMT1 RECORDING TECHNOLOGY
	...1....		DEB2XUPF	USER BSAM PGFIX IN USE
	....1...		DEB2XEXCP	BAM DETECTED ONE OR MORE EXCP OR XDAP ISSUANCES
	1		DEB2XSBS	START BLOCK ZERO STORED IN DEBBLKID
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

Table 114. DEB fields (continued)

Offset	Type/Value	Len	Name (Dim)	Description
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)	SIGNED	4	DEB2XTBL	ACCUMULATIVE BLOCKCNT ACROSS VOLUMES
172(AC)	SIGNED	4	DEB2XTCB	DCB BLOCKCNT AT TCLOSE OUTPUT
176(B0)	SIGNED	1	DEB2XSGT	STORAGE GROUP TYPE (GENERAL OR SPECIFIC)
177(B1)	SIGNED	1	DEB2XPAR	STARTING POSITION OF TAPE FILE
178(B2)	BITSTRING	1	DEB2XFG4	FOURTH FLAG BYTE
	..1.....		DEB2XEEF	3592-E05 WRITING IN EFMT1 RECORDING TECHNOLOGY
	...1....		DEB2XE1M	KEKLS 1 METHOD HASH
	....1...		DEB2XE1I	KEKLS 1 INPUT HASH
	.....1..		DEB2XE2M	KEKLS 2 METHOD HASH
	......1.		DEB2XE2I	KEKLS 2 INPUT HASH
179(B3)		1		RESERVED
180(B4)	UNSIGNED	4	DEB2XSBL	STARTING BLOCK NUMBER AFTER POSITIONING TAPE
184(B8)	ADDRESS	4	DEB2XSTV	ADDRESS OF FIRST VOLSER ENTRY
188(BC)	ADDRESS	4	DEB2XNXV	ADDRESS NEXT VOLSER ENTRY
192(C0)	SIGNED	4	DEB2XVLN	NUMBER OF VOLS IN DATASET VOLUME LIST
196(C4)	SIGNED	4	DEB2XMXV	MAXIMUM NUMBER OF VOLUME ENTRIES
200(C8)	UNSIGNED	8	DEB2XDID	TIMESTAMP ID FOR THE DEB
208(D0)	UNSIGNED	1	DEB2XLK1	LENGTH OF KEYLABEL1
209(D1)	UNSIGNED	1	DEB2XLK2	LENGTH OF KEYLABEL2
210(D2)		2		RESERVED
212(D4)	UNSIGNED	4	DEB2XKET	TAPE ENCRYPTOIN KEY EXCHANGE TIME
216(D8)	ADDRESS	4	DEB2XDXP	POINTER TO DEB EXTENSION
218 (DA)	BITSTRING	1	DEB2XFG8	EIGHTH FLAG BYTE
	1.......		DEB2XDBYP	DCBE_BYPASS_AUTH WAS SET ON
	.1......		DEB2XJBYP	JSCBPASS WAS SET ON
	..1.....		DEB2XAUTH	CALLER OF OPEN WAS AUTHORIZED SUPERVISOR STATE, SYSTEM KEY, OR APF
	...1....		DEB2XBYP	OPEN BYPASSED SAF/RACF CHECKS
	....1...		DEB2XBPNOTPDS	BPAM USED BUT DATA SET IS NOT PDS, PDSE, OR UNIX DIRECTORY
219(DB)		1		RESERVED
220(DC)	CHARACTER	7	DEB2XVEN	VOLUME LIST. (NUMBER OF ENTRIES DEPENDS ON DEB2XMXV — ONE ENTRY PER VOLUME, UP TO THE MAXIMUM NUMBER OF VOLUME ENTRIES.)

Cross-Reference

Name	Hex Offset	Hex Value	Level
DEBABEND	C	80	2
DEBACCS	C	F	2
DEBACHK	34		2

Name	Hex Offset	Hex Value	Level
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
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
DEBBLKID	34		2
DEBBLKSI	C		2
DEBBLKSZ	2A		2
DEBBOE	30		2
DEBBOERV	30		2

Name	Hex Offset	Hex Value	Level
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
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

Name	Hex Offset	Hex Value	Level
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

Name	Hex Offset	Hex Value	Level
DEBEXTOK	2C		2
DEBFGRV1	F	40	2
DEBFGRV2	F	20	2
DEBFGRV3	F	10	2
DEBFGRV6	F	2	2
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

Name	Hex Offset	Hex Value	Level
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
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
DEBNMTRKHI	5		2
DEBNOEE	28		2
DEBNPEE	24		2
DEBNRD	29	2	2

Name	Hex Offset	Hex Value	Level
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

Name	Hex Offset	Hex Value	Level
DEBRSV07	28		2
DEBRSV08	2C		2
DEBRSV09	2C	8	2
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

Name	Hex Offset	Hex Value	Level
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
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

Name	Hex Offset	Hex Value	Level
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

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

Name	Hex Offset	Hex Value	Level
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
DEB2XTNN	0		2
DEB2XUOF	8B		2

Chapter 15. 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
- 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.

Command	Keywords
TRACE	,WTRSTART=*membername* [WRAP|NOWRAP]]]
	CT [,WTRSTOP=*jobname*]]
	[CT [,ON]{ {COMP=SYSSMS } [,SUB=(*sub[sub],...*)]
	[,PARM=CTISMS*xx*] }]
	[CT [,*nnn*K] { }]
	[CT [,*nnnn*M] { }]
	[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.

Command	Parameters
REPLY	id
	[,JOBNAME=(jobnamelist)]

Command	Parameters
	ASID[,=(asidlist.)] [,OPTIONS=(namename...)] [,WTR=procname|DISCONNECT] [,END]

Note:

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 115. 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 116. Functional Areas of HFS Management

COMP	SUBCOMP
PFS	—
SSF	ARM SAC CHF DMP ERS VSM RCS

Note:

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 116 on page 476 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 '*nnn*K' is the trace buffer size.

   Or activate trace and specify options with response:

   `TRACE CT,500K,COMP=PFS`

   Note:

   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.

Type	Exit	Keywords
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 these 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.

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 the logrec data set

This topic shows an excerpt from a formatted logrec data set 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 37 on page 482 shows an excerpt from a formatted logrec data set entry.

```
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 37. Formatted SYS1.LOGREC Entry Example

```
  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

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    |........$..... .|
  +020    7F2CDEC0    7F100018    0000009D    5B200123    |"..{".......$...|
...
```

Figure 38. Formatted SYS1.LOGREC Entry Example (continued)

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 will not function correctly when it is in this state. Some requests will return an error with a return code of ENOSPC (00000085x) and a reason code of 5Bxx0E37x. As soon as possible, the file system should be evaluated and any necessary corrective actions taken. For more information, see message IGW022S in *z/OS MVS System Messages, Vol 9 (IGF-IWM)*. 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 wilI 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 117. 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
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.

Reason Code	Description
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 16. PDSE diagnostic aids

The following diagnostic aids are provided for PDSE:

- A restartable SMSPDSE1 address space to eliminate the need to re-IPL because of PDSE hang conditions.
- Operator commands to help diagnose PDSE errors
- IPCS VERBEXIT SMSXDATA dump formatting
- A sample entry in the logrec data set for a PDSE problem
- Return codes that are common to all PDSE modules
- Reason codes that are common to all PDSE modules
- A component trace facility that can be used to diagnose PDSE problems. This component trace facility should be used only at the request of IBM Support, in which case procedures for using the trace will be provided by IBM Support.

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.

Restartable PDSE address space (SMSPDSE1)

SMSPDSE1 is a restartable address space that provides connections to and processes requests for those PDSE data sets that are not part of the global connections associated with SMSPDSE. Global connections are made to those PDSE data sets that are in the LNKLIST data set concatenation. To create the SMSPDSE1 address space during IPL NIP processing in a sysplex coupled systems environment, set the IGDSMS*xx* as follows:

- PDSESHARING(EXTENDED)
- PDSE_RESTARTABLE_AS(YES)

Related reading: For information about configuring the restartable SMSPDSE1 address space, see *z/OS DFSMS Using Data Sets*.

Considerations for restarting the SMSPDSE1 address space

You may decide to restart the SMSPDSE1 address space to try to correct system problems due to a problem in PDSE processing. Before doing so, however, you should be aware this action could result in failures of currently running jobs and TSO sessions that are accessing PDSEs. In addition, you should note the following restrictions and limitations:

1. The SMSPDSE1 address space restart might not work correctly if more than one SMSPDSE1 address space restart is attempted concurrently because of the xQUIESCE time interval that is initiated by the restarting SMSPDSE1. If a SMSPDSE1 restart is performed on more than one system at the same time, then these restarting systems will not get the benefit of the other systems' xQUIESCE time duration interval.
2. If the SMSPDSE1 address space terminates as a result of a hard failure or because of the FORCE command, then the results of a PDSE address space restart are unpredictable. Some undesirable effects that could result from a FORCE command are:
   - If a member is deleted while SMSPDSE1 is down, then a reconnection to that member will fail.
   - If a user on another system opens for update when the restartable connection had a data set open for read/write, the restart connection will fail.
3. If the user address space fails to reconnect after the SMSPDSE1 address space restarts, then the user address space should be forced off of the system.

4. The restart of the SMSPDSE1 address space could result either in a loss of some SMF I/O counts or in duplicate counts.
5. If you used the VARY SMS,PDSE1,MONITOR command to change the PDSE MONITOR values and the SMSPDSE1 address space is restarted, then the values provided either by system defaults or by the MONITOR values specified in the IGDSMS*xx* member of SYS1.PARMLIB will be used to restart the PDSE1 MONITOR.
6. If callers that do not use the standard PDSE interface are connected, then PDSE will not know that the connection is active (and not behaving as expected). At this time there are no known callers of PDSE that fit this criteria.

Restarting the SMSPDSE1 address space

You can use operator commands to restart or activate the SMSPDSE1 address space. This type of operator intervention occurs only in extreme failure situations; for example, when PDSEs are hung because a canceled task is holding a latch.

For more information on using the VARY SMS commands for the SMSPDSE and SMSPDSE1 address spaces, see *z/OS MVS System Commands*. Before you decide whether to restart the SMSPDSE1 address space, review the "Considerations for restarting the SMSPDSE1 address space" on page 487.

Perform the following steps to restart the SMSPDSE1 address space:

1. Issue the VARY SMS,PDSE1,ANALYSIS operator command to determine which PDSE is causing the hang and which latch is being used. This command detects common problems that cause PDSE breakage.

   ```
   VARY SMS,PDSE1,ANALYSIS
   ```

   This example analyzes the all of the PDSEs in the SMSPDSE1 address space.
2. Cancel the related task to attempt to free the latch that is used.
3. If the system is still hung, issue the VARY SMS,PDSE1,FREELATCH command to force the release of the latch.

   ```
   VARY SMS,PDSE1,FREELATCH(latchaddr,ASID,tcbaddr)
   ```

   This example frees the latch at the specified latch address.
4. Ensure that PDSE_RESTARTABLE_AS(YES) and PDSESHARING(EXTENDED) are specified in the IGDSMS*xx* parmlib member.
5. If the system is still hung, restart the SMSPDSE1 address space. This command is valid if the system was IPLed with the restartable SMSPDSE1 address space.

   ```
   VARY SMS,PDSE1,RESTART
   ```

   The system terminates and restarts the SMSPDSE1 address space, and reconnects previously connected PDSEs.

   ```
   VARY SMS,PDSE1,RESTART,QUIESCE(5),COMMONPOOLS(REUSE)
   ```

   In this example, the system waits five seconds to quiesce current operations before shutting down the current SMSPDSE1 address space. During this quiesce interval, new requests are held until the SMSPDSE1 address space is restarted. The restarted SMSPDSE1 address space reuses the storage cell pools in the extended common storage area (ECSA).

   The output is:

   ```
   IGW0701 SMSPDSE1 WILL RESUME ALL USER TASKS.
   ```

Activating the SMSPDSE1 address space

Before you begin: ACTIVATE is a command of last resort. It should only be used when the restart failed to work and the operator forced the SMSPDSE1 address space down manually. Note: IBM does not recommend forcing down the SMSPDSE1 address space because the results are unpredictable. If the SMSPDSE1 address space is stopped, you can use the VARY SMS,PDSE1,ACTIVATE command to activate the SMSPDSE1 address space.

Perform the following steps to activate the SMSPDSE1 address space after it has been shut down:

1. Issue the VARY SMS,PDSE1,ACTIVATE operator command to activate the SMSPDSE1 address space.

   ```
   VARY SMS,PDSE1,ACTIVATE,COMMONPOOLS(REUSE)
   ```

   In this example, the SMSPDSE1 address space is activated and the common storage cell pools in ECSA are reused. SMSPDSE1 startup processing creates storage cell pools in the ECSA. When SMSPDSE1 restarts, the old storage cell pools are not deleted. Normally, it is preferable to reuse the existing ECSA cell pools to avoid ECSA depletion. If the SMSPDSE1 restart is related to a problem with the existing ECSA cell pools, then you can create new ones by specifying COMMONPOOLS(NEW).

   ```
   VARY SMS,PDSE1,ACTIVATE,COMMONPOOLS(NEW)
   ```

   In this example, the SMSPDSE1 address space is activated and new storage cell pools are created in the ECSA.

Monitoring the SMSPDSE or SMSPDSE1 address space

Use the MONITOR command to modify processing of the PDSE or PDSE1 monitor – to start or stop the monitor, and set an interval and duration for monitoring .

```
VARY SMS,PDSE1,MONITOR,ON,60,15
```

This example turns on the monitor for the SMSPDSE1 address space with an interval of 60 seconds and duration of 15 seconds.

The output is:

```
IGW043I PDSE MONITOR IS ACTIVE
+ INVOCATION INTERVAL: 60 SECONDS
+ SAMPLE DURATION: 15 SECONDS
```

```
VARY SMS,PDSE1,MONITOR,OFF
```

This example turns off the monitor for the SMSPDSE1 address space.

```
VARY SMS,PDSE1,MONITOR,RESTART
```

This example restarts the monitor for the SMSPDSE1 address space.

Operator commands for analyzing and repairing PDSE and PDSE1 problems

Although PDSE and PDSE1 errors are not common, when errors do occur in the code for PDSEs and PDSE1s, the results often extend beyond the job or user who encountered the error. This is a result of the complexity of the PDSE and PDSE1 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 and PDSE1 structures or the selection of which element of the system should be terminated is also determined. The two commands are: the VARY SMS,PDSE | PDSE1, ANALYSIS command (referred to as the ANALYSIS command) and the VARY SMS,PDSE | PDSE1, FREELATCH command (referred to as the FREELATCH command).

These commands are commonly used when the processing for one or more PDSEs or PDSE1s hang.

Definitions

BMIB
Root control block for PDSE member and directory pages.

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.

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.

HASH TABLE
A structure that can be used to allow for the lookup of various PDSE control blocks, including DIBs and HL1Bs.

HL1B
A control block that is used by PDSE processing to represent the serialization for the PDSE.

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|PDSE1`commands, there are no mechanisms that allow you to directly determine or modify the status of the PDSE latches.

LOCK
The internal mechanism that is used to serialize access to the PDSE internal structures.

LSD and LADE
Internal PDSE control block which represent the pages which make up the directory or members of a PDSE.

XCM
The subcomponent of PDSE processing that is responsible for communicating between systems using the XCF facilities.

ACTIVATE command

Issue the ACTIVATE command to bring up the restartable PDSE address space.

Command format

```
V SMS,PDSE|PDSE1,ACTIVATE
 [ , COMMONPOOLS( NEW  | REUSE ) ]
```

Operands

PDSE1
Allows you to select the restartable SMSPDSE1 address space.

COMMONPOOLS(NEW)
SMSPDSE1 abandons the old common storage cell pools and creates a new set of cell pools in ECSA.

COMMONPOOLS(REUSE)
This is the default. SMSPDSE1 reuses the existing common storage cell pools that were created by the previous instance of SMSPDSE1.

Restriction: When you restart SMSPDSE1, the old storage cell pools are not deleted. Reuse the existing ECSA cell pools to avoid ECSA depletion. If you restart SMSPDSE1 because of a problem with the existing ECSA cell pools, create new cell pools by specifying COMMONPOOLS(NEW).

ANALYSIS command

Issue the ANALYSIS command only when you suspect that one or more users or jobs are having problems accessing a PDSE or PDSE1. This command and the FREELATCH command use a sampling algorithm that interrogates the state of the PDSE or PDSE1 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 or PDSE1 does not change.

Command format

```
V SMS,PDSE|PDSE1,ANALYSIS
 [ , DSNAME(dsname)[,VOLSER(volser)] ]
   [,RETRIES(retries|1500)]
```

Operands

DSNAME
When specified, causes the analysis to be performed for a particular PDSE or PDSE1. If VOLSER is omitted, the data set is found using the default system catalog.

VOLSER
Allows you to specify an uncataloged PDSE or PDSE1. If specified, VOLSER must follow DSNAME.

RETRIES
Allows you to control the amount of time for which the particular PDSE or PDSE1 situation must remain static. The PDSE or PDSE1 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. The minimum value for RETRIES is 1, the maximum is 32767.

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 493 for details about the possible outputs and the actions you can choose to take for the various conditions detected.

Function

The ANALYSIS command detects a number of the most common problems that result in PDSE or PDSE1 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:

ASRBASCB Hash table latch held
Prevents the PDSE1 address space from restarting. Also may prevent jobs from using PDSEs in a new initiator or the shutting down of an address space.

ASRBULCH latch held
Serializes recovery processing for a user address space.

Contention message to another system is outstanding
Indicates that another system is failing to perform its required role in the sharing of PDSEs or PDSE1s across the sysplex. Whenever this error occurs, the most common problem source will be the system that has not responded to the message.

BMIBREPAIR
Prevents certain repair functions for the I/O subsystem.

CONTROL BLOCK BROKEN
The system encountered an invalid control block in running this command.

DIB Hash Table latch held
Prevents the opening and closing of a subset of the PDSEs or PDSE1s on this system.

DIB latch held
Prevents the selected PDSE or PDSE1 from being opened or closed on this system.

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.

FIBHashTableLATCH latch held
Serializes member processing for a single PDSE.

FIB_LATCH latch held
Serializes a single member for a PDSE.

H1LB Hash table latch held
Prevents the opening and closing of a subset of the PDSEs or PDSE1s 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.

LATCH BROKEN
The data structures for a latch have been overlaid.

LATCH NULLIFIED
The PDSE or PDSE1 code has made the latch not available. The code usually does this when it closes the PDSE or PDSE1 for the last time on a system. This is an unexpected situation.

LICBASRBCHAINLATCH latch held
Serializes PDSE user address space tracking and recovery.

LSDLDELT and LSD_LATCH
Prevents I/O to PDSE directories or members.

LSDLTDW and LADELTDW
Prevents connecting or disconnection to PDSE directories or members.

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 or PDSE1s that this system has locked exclusive. The latch is only acquired by tasks that run in the SMXC address space.

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.

asrbasrb
Address of the ASRB that represents the users address space in the PDSE address space.

licblicb
Address of the LICB which is the root control block for PDSE locking operations.

fibfibfi
Address of the FIB which is the root control block for a member of 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 exception detected is either temporary in nature or is not one of the class of problems that this tool is designed to detect or fix. If message IGW038A continues to occur, obtain an SVC dump of the SMSPDSE(1) address space 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:dddddddd

LATCH:*llllllll* Holder (*aaaa: tttttttt*)

Holding JOB:*jjjjjjjj* | Holding Started Task (*stststst* |

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 restart the PDSE address space.

++ Unable to latch HL1B:*hhhhhhhh*

LATCH: Holder (aaaa: tttttttt)

Holding JOB:*jjjjjjjj* | Holding Started Task (*stststst* |

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 restart the PDSE address space.

++ Unable to latch HL1BPLCH:*hhhhhhhh*

LATCH:*llllllll* Holder (*aaaa: tttttttt*)

Holding JOB:*jjjjjjjj* | Holding Started Task (*stststst* |

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 restart the PDSE address space.

++ Unable to latch ASRBULCH:*asrbasrb*

LATCH:*llllllll* Holder (*aaaa*: *tttttttt*)

Holding JOB:*jjjjjjjj* | Holding Started Task (*stststst*) |

Unable to determine job or started task name for holder

This latch is only held while the address space and task level lock recovery chains are being updated. This will prevent other PDSE operations in the same address space from proceeding. 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 latch. If that does not work, either re-IPL or restart the PDSE address space.

++ Unable to latch LICBASRBCHAINLATCH:*licblicb*

LATCH:*llllllll* Holder (*aaaa*: *tttttttt*)

Holding JOB:*jjjjjjjj* | Holding Started Task (*stststst*) |

Unable to determine job or started task name for holder

This latch is only held on the first PDSE request for a PDSE in an address space or at EOM for the same address space. It is used to serialize the acquiring and freeing of address space PDSE control blocks. 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 latch. If that does not work, either re-IPL or restart the PDSE address space.

++ Unable to latch ASRBASCB HASH TABLE LATCH:*licblicb*

LATCH:*llllllll* Holder (*aaaa*: *tttttttt*)

Holding JOB:*jjjjjjjj* | Holding Started Task (*stststst*) |

Unable to determine job or started task name for holder

This latch is obtained when an address space first utilizes PDSE resoruces, when an address space shuts down and when the PDSE1 address space restarts to determine which address spaces may have to reconnect to PDSE data sets and members.

++ Unable to latch FIBHashTableLATCH:*dddddddd*

LATCH:*llllllll* Holder (*aaaa*: *tttttttt*)

Holding JOB:*jjjjjjjj* | Holding Started Task (*stststst*) |

Unable to determine job or started task name for holder

This latch serializes the access to members of a PDSE. It can prevent most access of a PDSE from being performed. If will only be identified when a analysis command is directed to a single PDSE. 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 latch. If that does not work, either re-IPL or restart the PDSE address space.

++ Unable to latch FIB_LATCH:*fibfibfi*

LATCH:*llllllll* Holder (*aaaa*: *tttttttt*)

Holding JOB:*jjjjjjjj* | Holding Started Task (*stststst*) |

Unable to determine job or started task name for holder

This latch serializes the access to a single member of a PDSE. If will only be identified when a analysis command is directed to a single PDSE. 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 latch. If that does not work, either re-IPL or restart the PDSE address space.

++ 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:tttttttt*)

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:tttttttt*)

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.

FREELATCH command

This command releases any latch that is reported by the ANALYSIS command to be held for too long.

Command format

```
VARY SMS,PDSE | PDSE1, FREELATCH(latchaddr , ASID ,tcbaddr)
[ , Retries([ , Retries(  ) ]
```

Operands

latchaddr

Address of the latch that is to be released.

ASID

ASID of the holder of the latch. Specifying ASID=0 and tcbaddr=0 will cause the latch to be obtained and released. This will ensure that all waiters have a chance to acquire 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 or PDSE1 situation must remain static. The PDSE or PDSE1 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.

Function

The `VARY SMS,PDSE | PDSE1, FREELATCH` command releases any latch that the ANALYSIS command indicates is held and will not release.

When the asid=0 and the tcbaddr=0, it will release any waiters for the latch.

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 or PDSE1.

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.

REFRESH command

Use this command to discard cached PDSE or PDSE1 directory pages. This ensures that the next access to the specified PDSE will use the data directly from the device.

Command format

```
V SMS,{PDSE|PDSE1},REFRESH,DSN(dsname)[,VOL(volser)]
```

Operands

{PDSE|PDSE1},REFRESH
Use this operand to discard cached directory pages for a data set.

DSN(*dsname*)[,VOL(*volser*)]
Specifies the PDSE for which you want to discard cached directory pages. VOL(*volser*) is required only when the PDSE is not cataloged.

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
```

RESTART command

This command recycles the restartable PDSE address space (SMSPDSE1). This command is valid if the system was IPL'd with a restartable PDSE address space.

Restriction: PDSERESTART(YES) and PDSESHARING(EXTENDED) must be up and running to use this command.

Command format

```
V SMS,PDSE1,RESTART
 [ , QUIESCE ( duration | 3 )
COMMONPOOLS ( NEW | REUSE )]
```

Operands

PDSE1
Allows you to select the restartable SMSPDSE1 address space.

RESTART
Use this operand to terminate the SMSPDSE1 address space and immediately activate a new instance of the SMSPDSE1 address space.

QUIESCE
Specifies the maximum time interval, in seconds, that existing in-flight operations quiesce before the current instance of the SMSPDSE1 address space is terminated.

Restriction: If the interval chosen is too long and SMSPDSE1 address requests do not complete, the address space restart is delayed. This delay affects processing on this system, and can also affect PDSE processing on other systems in the sysplex.

COMMONPOOLS(*NEW*)
SMSPDSE1 abandons the old common storage cell pools and creates a new set of cell pools in ECSA.

COMMONPOOLS(*REUSE*)
This is the default. SMSPDSE1 reuses the existing common storage cell pools that were created by the previous instance of SMSPDSE1.

Restriction: When you restart SMSPDSE1, the old storage cell pools are not deleted. Reuse the existing ECSA cell pools to avoid ECSA depletion. If you restart SMSPDSE1 because of a problem with the existing ECSA cell pools, create new cell pools by specifying COMMONPOOLS(NEW).

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
```

PDSE records written to the logrec data set

This section shows an example of a formatted logrec data set 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 39 on page 499 shows a formatted logrec data set entry example.

```
TYPE:  SYMPTOM RECORD    REPORT: SOFTWARE EDIT REPORT   DAY YEAR
                       REPORT DATE: 266  04
SCP:   VS 2 REL 3           ERROR DATE:    217  04
                 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  04
 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
```

```
 +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 39. Formatted SYS1.LOGREC Entry

PDSE return and reason codes

PDSE return and reason codes are found in:

- ABEND0F4 dump title
  - Example:

```
      COMPID=DF115,CSECT=IGWISRCH+05C8,DATE=04/12/05,MAINTID= UW58377
      ,ABND=0F4,RC=00000024,RSN=145AA033
```

- LOGREC

  - Example:

```
SEARCH ARGUMENT ABSTRACT:
     PIDS/566528464 LVLS/DFSMSV1R5 RIDS/IGWBSDMI PRCS/0000000C
     PRCS/2009001B
```

- IGW*xxx* messages.

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 118. 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.
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.

Reason Code	Description
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.
X'01xx5B64'	The data set is not an HFS data set.
X'04460318'	Another PDSE with same synonym name (VSGT01-*volser-ttttrr*) is open on another CPU.
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.

Reason Code	Description
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.
X'1Dxx4E54'	Unable to free a SOR.

Reason Code	Description
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'12xx0802'	PDSE buffered data have not been purged because one or more PDSEs are OPEN on this system.
X'12xx0803'	One or more PDSEs are OPEN on a system other than this system and thus have not been purged from all system buffers.
X'12xx0804'	PDSE buffered data have not been purged from all system buffers because one or more PDSEs are OPEN on this system and on one or more other systems.
X'12xx2401'	Buffer invalidation failed on this system. PDSEs may still be in use.
X'12xx2402'	Broadcast of buffer invalidation to other systems failed. PDSEs may still be in use.
X'12xx2403'	Buffer invalidate failed on some other system in the sysplex. PDSEs may still be in use.
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.

Reason Code	Description
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.

Reason Code	Description
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.
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.

PDSE validation reason codes

Reason Code	Description
X'8000'	Validate failed dump
X'8001'	Invalid VDF
X'8002'	Unable to access AD page
X'8002'	Invalid AD page
X'8003'	Hierarchy problem
X'8004'	ICPR length inconsistence

Reason Code	Description
X'8005'	Bad threshold key
X'8006'	Validate unable to process ND
X'8007'	Invalid ND page
X'8008'	Unable to access ND page
X'8009'	Btree instantiate failed
X'800A'	Btree disable failed
X'800B'	Btree create failed
X'800C'	Btree add failed
X'800D'	Excess AD name records
X'800E'	Orphan ND name records
X'800F'	AD key order violation
X'8010'	ND key order violation
X'8011'	Doubly allocated page
X'8012'	APMs inconsistent
X'8013'	AD number of records invalid
X'8014'	ND number of records invalid
X'83FF'	Validate last RS
X'8400'	Invalid segment in AD
X'8401'	Invalid segment in ND
X'8402'	Invalid ICTL in AD
X'8403'	Invalid ICTL in ND

Chapter 17. Storage Management Subsystem diagnostic aids

The following major diagnostic aids are provided for the Storage Management Subsystem (SMS):

- The SETSMS VOLSELMSG command, for performing problem diagnosis on volume selection.
- 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 UNIX files and IPCS formatting with SMSXDATA
- 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

SETSMS VOLSELMSG command

The SMS command, SETSMS VOLSELMSG(ON) can be used to request summarized and detailed analysis messages on volume selection, if volume selection is not prematurely terminated by an error. These analysis messages can assist the user to perform problem diagnosis on volume selection. See *z/OS MVS System Commands* for information on using SETSMS VOLSELMSG(ON).

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)*

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 119 on page 510 describes the control blocks in a dump that can be formatted.

Table 119. 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.

Table 119. Control Blocks Formatted by SMSDATA (continued)

Control Block	Macro Description
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.

Command	Parameters
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 | DEBUG 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*)]

<table>
<tr><th>Command</th><th>Parameters</th></tr>
<tr><td></td><td>[TADDR(<i>ERTRE address</i>)]
[STORAGE(<i>IMC address,IMC asid</i>,[DATA | FLO | IMC])]</td></tr>
</table>

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 blocks 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 parameters 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 blocks that are *not related to the configuration* by specifying the applicable parameters 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
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.

DEBUG
Trace temporary trace points.

DISP
SMS disposition processing.

DPN
SMS device pool name selection subsystem interface.

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).

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.

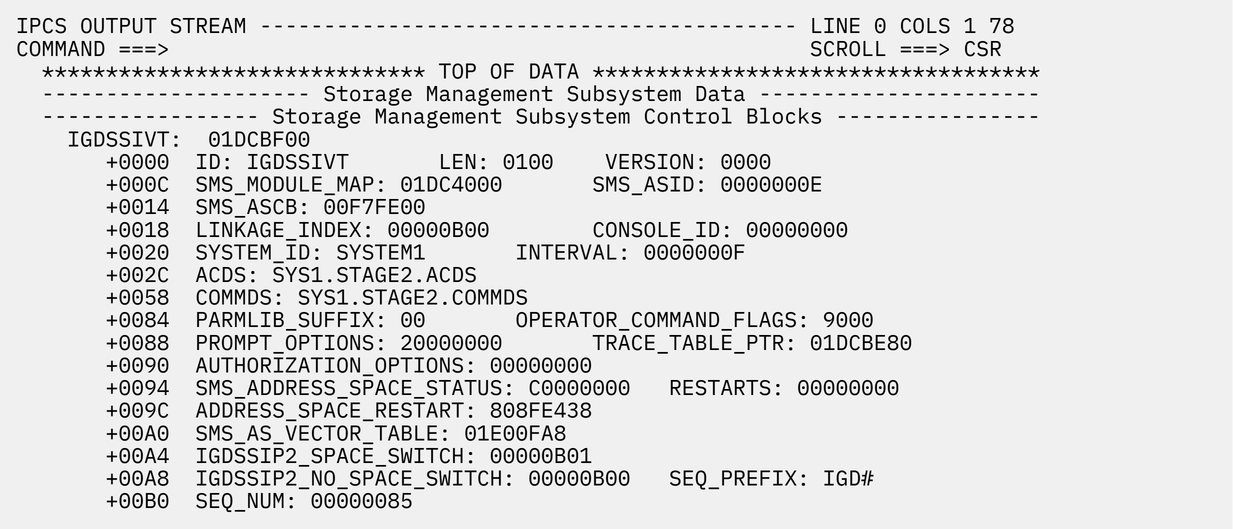
SMSDATA formatting examples

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 40 on page 516.

```
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
```

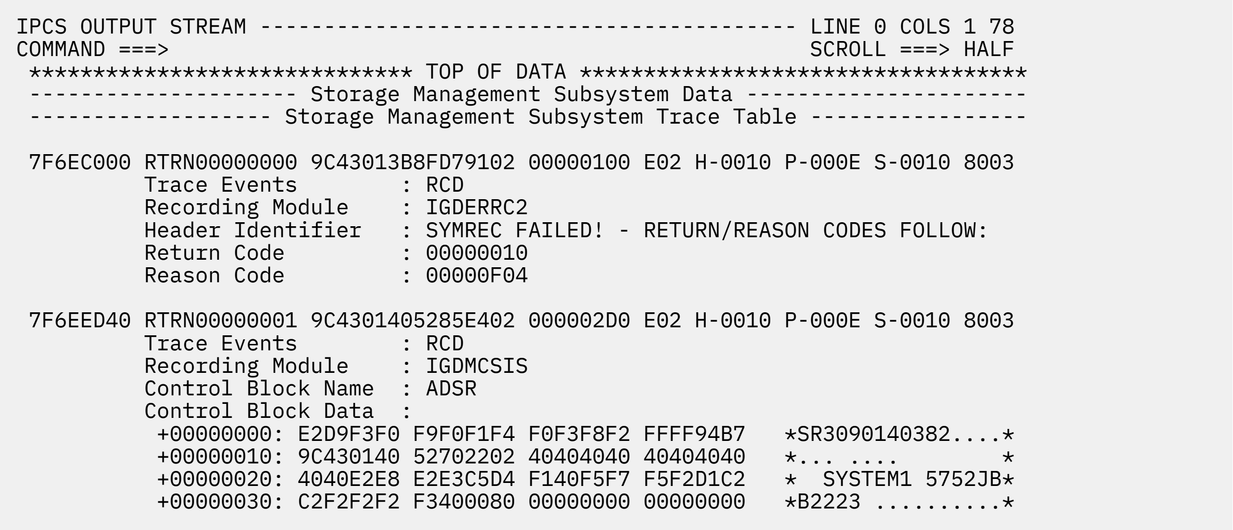
Figure 40. SMSDATA Formatting of the SSIVT Control Block

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 41 on page 516.

```
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 41. SMSDATA Formatting of the SMS Trace Table

SMSDATA trace record format

Each trace record 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 events 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 42 on page 518 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 43 on page 519 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 40 on page 516, 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 42 on page 518:

```
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: 06/14/04  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 42. 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 42 on page 518 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.

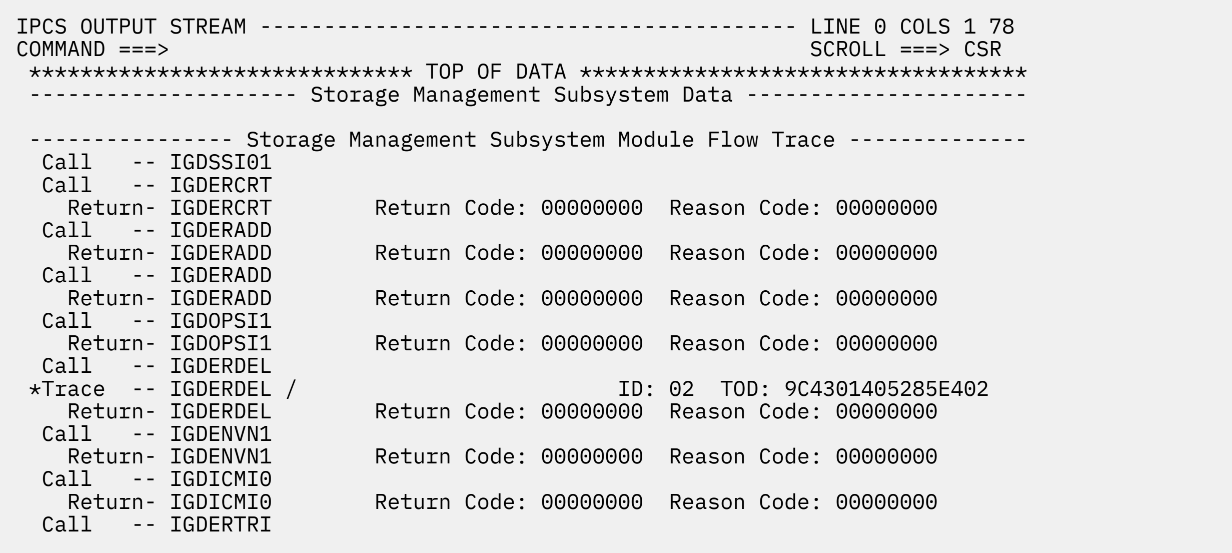
AUTODATA
The automatic data area used by the owning module. The offsets are relative to the SMS base register used and can be used when analyzing the assembler code. For example, if the owning module issues an X'5860B06C' instruction, then, using Figure 42 on page 518, register 6 would be loaded with 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 43 on page 519:

```
    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 43. 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 510 and "Formatting the Storage Management Subsystem trace table" on page 516 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*. Entries in the trace table consist of certain 'events' that occurred during SMS processing.

Areas of SMS processing that can be traced are listed under the description of the SELECT parameter of the SETSMS command.

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.

IGDVSUIB (volume selection user information block)

When you take an SMS trace with the VTOCC or VTOCA option, then (if volume selection was not prematurely terminated by an error), the trace facility will also create a trace entry containing an IGDVSUIB (volume selection user information block) in the private area of the SMS address space for each data set being traced. IGDVSUIB will contain key information used in SMS volume selection, which you can use to perform problem diagnosis on volume selection problems.

The format of IGDVSUIB is:

```
1   VSUIB

      OFFSET   OFFSET
     DECIMAL      HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
    ======== ======== ==================  ============== ===============================
           0      (0) STRUCTURE         *  VSUIB          VOLSEL USER INFO BLOCK
           0      (0) CHARACTER       120   VSUIB_FIXED   START OF FIXED PART
           0      (0) CHARACTER        16    VSUIB_HEADER
           0      (0) CHARACTER         8     ID          EBCDIC EYE CATCHER IGDVSUIB
           8      (8) SIGNED            2     VERNO       VERSION NUMBER OF MACRO
          10      (A) CHARACTER         2     *           RESERVED
          12      (C) CHARACTER         4     LEN         LENGTH OF IGDVSUIB
          16     (10) SIGNED            4    VOL_CTR      NUMBER OF VOLUMES IN TABLE
          20     (14) CHARACTER        44    DSN          DATA SET NAME
          64     (40) CHARACTER         8    SC           STORAGE CLASS NAME
          72     (48) CHARACTER         8    DC           DATA CLASS NAME
          80     (50) SIGNED            8    SPACEKB_REQUESTED
                                                         REQUESTED SP FOR NVSAM
                                                         OR CLUSTER/DATA COMPONENT
          88     (58) SIGNED            8    SPACEKB_FOR_INDEX
                                                         REQUESTED SP FOR INDEX
          96     (60) UNSIGNED          1    DS_TYPE      DS TYPE - VSAM OR NONVSAM
          97     (61) UNSIGNED          1    DS_SUBTYPE   DATA SET SUBTYPE (UNUSED) FOR
                                                         VSAM: 1 - LDS 2 - RRDS 3 -
                                                         ESDS 4 - KSDS 5 - VRRDS
          98     (62) CHARACTER         2    *            RESERVED
         100     (64) BIT(32)           4    DS_FLAGS     FLAGS
         100     (64) BIT(8)            1     DS_FLAG1    FIRST FLAG BYTE
                      1... ....                VSAM_DS    VSAM DATA SET
                      .1.. ....                GUARANTEED_SPACE
                                                         GUARANTEED SPACE REQUEST
                      ..1. ....                BEST_FIT_REQUEST
                                                         BEST FIT REQUEST
                      ...1 ....                STRIPING_REQUEST
                                                         STRIPING REQUEST
                      .... 1111                *          RESERVED
         101     (65) BIT(24)           3     *           RESERVED
         104     (68) CHARACTER        16    *            RESERVED
         120     (78) CHARACTER        64   VOL_VAR(*)    START OF VARIABLE PART
         120     (78) CHARACTER         8    SG           STORAGE GROUP NAME
         128     (80) UNSIGNED          1    SG_TYPE      STORAGE GROUP TYPE: 1-POOL,
                                                         2-OVERFLOW, 3-EXTEND
         129     (81) CHARACTER         3    *            RESERVED
         132     (84) CHARACTER         6    VOLSER       VOLUME SERIAL NUMBER
         138     (8A) CHARACTER         1    *            RESERVED
         139     (8B) CHARACTER         1    VOL_STATE    VOLUME STATE: A - SPACE
                                                         ALLOCATED N - NOT USED R -
                                                         REJECTED
         140     (8C) SIGNED            4    TOTAL_SPACEMB
                                                         TOTAL SPACE IN MB
         144     (90) SIGNED            4    FREE_SPACEMB FREE SPACE IN MB
         148     (94) SIGNED            4    THRESHOLD_SPACEMB
                                                         BELOW THRESHOLD SPACE IN
                                                         MB
         152     (98) CHARACTER         5    VOL_PREFERENCE
                                                         VOLUME PREFERENCE
         157     (9D) CHARACTER         3    *            RESERVED
         160     (A0) BIT(32)           4    FAIL_BYTES   FAILURE REASON FLAGS
         160     (A0) BIT(8)           1     FAIL_BYTE1  FIRST FAILURE REASON BYTE
         161     (A1) BIT(8)           1     FAIL_BYTE2  SECOND FAILURE REASON BYTE
1   VSUIB

      OFFSET   OFFSET
     DECIMAL      HEX  TYPE         LENGTH  NAME (DIM)     DESCRIPTION
    ======== ======== ==================  ============== ===============================
         162     (A2) BIT(8)           1     FAIL_BYTE3  THIRD FAILURE REASON BYTE
         163     (A3) BIT(8)           1     FAIL_BYTE4  FOURTH FAILURE REASON BYTE
         164     (A4) SIGNED           4    DADSM_DIAGDATA
                                                         DADSM DIAGNOSTIC CODE
         168     (A8) CHARACTER       16    *            RESERVED

    CONSTANTS

      LEN  TYPE           VALUE          NAME            DESCRIPTION
    =====  ============================  =============== ===============================
        8  CHARACTER   IGDVSUIB          VSUIB_IDC
        4  DECIMAL            1          VSUIB_VERNOC
        4  DECIMAL          184          VSUIB_LENC

            CONSTANTS
    ========================================================================================

              Identification of level of macro
    ========================================================================================
        1  DECIMAL            1          FVSP_VERSION    Version number.
        8  CHARACTER   IGWFVSP           FVSP_CB_ID      Control block id.

            CONSTANTS
    ========================================================================================

              Identification of level of macro
    ========================================================================================
        1  DECIMAL            1          FVSX_VERSION    Version number.
        8  CHARACTER   IGWFVSX           FVSX_CB_ID      Control block id
           BIT         1                 FVSX_IS_DREFD   Stack is DREFD storage
           BIT         0                 FVSX_IS_NONDREFD
                                                         Stack not DREFD
                                                         storage
        4  DECIMAL            5          FVSX_DATAMGT_KEY
                                                         Data management key
        4  DECIMAL        32768          FVSX_STACK_LEN  stack length
        4  DECIMAL        12288          FVSX_DREF_STK_LEN
                                                         DREFd stack
                                                         len
        4  DECIMAL        16384          FVSX_DREF_ERSTK_LEN
                                                         Drefd ers
                                                         stack len
        4  DECIMAL         8192          FVSX_8K_LEN     8k stack length
        4  DECIMAL        16384          FVSX_16K_LEN    16k stack len
        4  DECIMAL        32768          FVSX_32K_LEN    32k stack len

            CONSTANTS
    ========================================================================================

              Identification of level of macro
    ========================================================================================
        1  DECIMAL            1          FVSS_VERSION    Version number.
        8  CHARACTER   IGWFVSS           FVSS_CB_ID      Control block id.
1   CONSTANTS

      LEN  TYPE           VALUE          NAME            DESCRIPTION
    =====  ============================  =============== ===============================
        4  DECIMAL          128          FVSS_RESERVE_LATCHINGSPACE
                                                         Amount of space to
                                                         reserve in autodata area stack
                                                         for latching calls.
1     CROSS REFERENCE

                                                   HEX         HEX
      NAME                                         OFFSET      VALUE     LEVEL
      ====                                         ======     ========   =====
      BEST_FIT_REQUEST                                 64           20       5
      DADSM_DIAGDATA                                   A4                    3
      DC                                               48                    3
      DS_FLAGS                                         64                    3
      DS_FLAG1                                         64                    4
      DS_SUBTYPE                                       61                    3
      DS_TYPE                                          60                    3
      DSN                                              14                    3
      FAIL_BYTES                                       A0                    3
```

```
       FAIL_BYTE1                                       A0                    4
       FAIL_BYTE2                                       A1                    4
       FAIL_BYTE3                                       A2                    4
       FAIL_BYTE4                                       A3                    4
       FREE_SPACEMB                                     90                    3
       GUARANTEED_SPACE                                 64           40       5
       ID                                                0                    4
       LEN                                               C                    4
       SC                                               40                    3
       SG                                               78                    3
       SG_TYPE                                          80                    3
       SPACEKB_FOR_INDEX                                58                    3
       SPACEKB_REQUESTED                                50                    3
       STRIPING_REQUEST                                 64           10       5
       THRESHOLD_SPACEMB                                94                    3
       TOTAL_SPACEMB                                    8C                    3
       VERNO                                             8                    4
       VOL_CTR                                          10                    3
       VOL_PREFERENCE                                   98                    3
       VOL_STATE                                        8B                    3
       VOL_VAR                                          78                    2
       VOLSER                                           84                    3
       VSAM_DS                                          64           80       5
       VSUIB                                             0                    1
       VSUIB_FIXED                                       0                    2
       VSUIB_HEADER                                      0                    3
```

Storage Management Subsystem records written to logrec data set

If SMS issues message IGD306I to report a failure, it also writes a software failure symptom record to the logrec data set. 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.

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 logrec data set entry.

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 44 on page 524 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 44 on page 524 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  *................................*
```

```
----- 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 44. SMS Sample SUMDUMP—Eye-catchers in EBCDIC Section

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. Table 120 on page 525 describes the conditions that cause the return code.

Table 120. 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 121 on page 525 describes the conditions that cause the return code.

Table 121. 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 122 on page 525 describes the conditions that cause the return code.

Table 122. 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.

Table 122. SMS Initialization Return Codes (continued)

Return Code	Meaning
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 123 on page 526 describes the conditions that cause the return code.

Table 123. 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 (X'0000') **Description**

Reason Code (X'0000')	Description
0000 (X'00')	Successful.
0004 (X'04')	Invalid SSSAID field.
0008 (X'08')	Invalid SSSAVER field.
0012 (X'0C')	Unknown SMS subfunction requested.
0016 (X'10')	SMS does not support the specified SSI function code.
0020 (X'14')	Caller not authorized to perform requested SMS function.
0300 (X'012C')	An SMS abend occurred.
0301 (X'012D')	Data set not eligible for allocation on SMS-managed volume.
0302 (X'012E')	Incompatible storage class and volume.
0303 (X'012F')	MGMTCLAS may not be specified for non-SMS data sets.
0304 (X'0130')	Use of referenced storage group not allowed by ACS routine.
0305 (X'0131')	Storage class assigned to data set which references a non-SMS data set.
0306 (X'0132')	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 (X'0133')	Error in an ACS installation exit.
0308 (X'0134')	Not authorized to the specified SMS class.
0309 (X'0135')	An SMS data set is not allowed within the scope of a JOBCAT/STEPCAT.
0310 (X'0136')	Neither management class nor storage class can be specified for a data set that is not eligible to be managed by SMS.
0313 (X'0139')	DSNTYPE cannot be specified for data set that is not managed.
0314 (X'013A')	DSNTYPE specified is not supported for the specified data class.
0315 (X'013B')	Invalid label specified for new mountable allocation.
0317 (X'013D')	DSNTYPE of PIPE may only be specified with PATH.
0318 (X'013E')	No storage groups of the same type as the referenced data set assigned to the referencing data set.
0320 (X'0140')	z/OS UNIX system services is not installed on this system.
0399 (X'018F')	Unable to add the recovery routine.
0400 (X'0190')	Unable to obtain ASCB pointer.
17246 (X'435E')	JFCBEXAD=0 (JFCB Extension is not provided by the caller) when volume count is greater than 5.).

ACS services reason codes

Reason Code	Description
1001 (X'03E9')	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 (X'03F3')	A nonexistent SMS class was returned.
1012 (X'03F4')	The storage group routine assigned a nontemporary data set to a storage group which is not of type POOL.
1013 (X'03F5')	No storage group was assigned by the storage group routine.

Reason Code	Description
1014 (X'03F6')	The explicitly specified SMS class does not exist.
1015 (X'03F7')	RACINIT failed.
1016 (X'03F8')	An error was detected by the ACS executor.
1017 (X'03F9')	Invalid expiration date.
1018 (X'03FA')	Invalid retention period.
1019 (X'03FB')	The TIME macro failed.
1020 (X'03FC')	The authorization check for management class failed.
1021 (X'03FD')	The authorization check for storage class failed.
1022 (X'03FE')	The authorization check for both management class and storage class failed.
1023 (X'03FF')	Nonzero return code from construct access services.
1024 (X'0400')	Storage group routine assigned an object to a storage group which is not of type object or object backup.
1025 (X'0401')	RACHECK failed; see SYS1.LOGREC record key 45 and key 6 for RACHECK return and reason codes.
1026 (X'0402')	RACDEF failed.
1027 (X'0403')	&SIZE or &MAXSIZE overflow.
1028 (X'0404')	Both pool and tape storage groups derived, only pool assigned.

ACS executor reason codes

Reason Code	Description
2001 (X'07D1')	The ACERO control block ID is incorrect.
2002 (X'07D2')	The ACERO control block version is incorrect.
2003 (X'07D3')	The ACERW control block ID is incorrect.
2004 (X'07D4')	The ACERW control block version is incorrect.
2005 (X'07D5')	The ACOHD control block ID is incorrect.
2006 (X'07D6')	The ACOHD control block version is incorrect.
2007 (X'07D7')	The data set name is blank.
2008 (X'07D8')	Invalid data set name; missing qualifier; two qualifier separators found adjacent (for example, A..BD).
2009 (X'07D9')	Invalid data set name; the qualifier is longer than the maximum allowable.
2010 (X'07DA')	Incorrect ACCT_JOB field; the field length sums are greater than the maximum allowable.
2011 (X'07DB')	Incorrect ACCT_STEP field; the field length sums are greater than the maximum allowable.
2012 (X'07DC')	The data set organization field contains an invalid value.
2013 (X'07DD')	The data set record organization field contains an invalid value.
2014 (X'07DE')	The data set type field contains an invalid value.
2015 (X'07DF')	The execution mode field contains an invalid value.
2016 (X'07E0')	Parameter error in ACERO; the value of the day for the year is greater than 366.
2019 (X'07E3')	LABEL field contains an invalid value.

Reason Code	Description
2020 (X'07E4')	DSNTYPE field is invalid.
2021 (X'07E5')	Member name is blank.
2022 (X'07E6')	Member name is invalid; missing qualifier.
2023 (X'07E7')	Member name is invalid; qualifier is too long.
2024 (X'07E8')	Data set name is invalid, for example, '.ABC.DEF'
2025 (X'07E9')	Space type contains invalid value.
2026 (X'07EA')	FLASHCOPY contains an invalid value.
2027 (X'07EB')	EATTR contains an invalid value.
2028 (X'07EC')	ACS_USERVAR contains an invalid value.
2030 (X'07EE')	Error in an ACS routine; invalid operation code.
2031 (X'07EF')	Error in an ACS routine; invalid type of operand.
2032 (X'07F0')	Error in an ACS routine; invalid subscript.
2033 (X'07F1')	Error in an ACS routine; invalid value for SET.
2034 (X'07F2')	Error in an ACS routine; invalid fragment for write.
2035 (X'07F3')	Error in an ACS routine; invalid read/only variable.
2036 (X'07F4')	Error in an ACS routine; end of data before end of ACS routine.
2037 (X'07F5')	Error in an ACS routine; invalid table in which to find variable or literal.
2038 (X'07F6')	Error in an ACS routine; unknown compare operand.
2040 (X'07F78')	Error in an ACS routine; unknown mask.
2041 (X'07F9')	Error in an ACS routine using the STCK instruction; the epoch bit was invalid.
2042 (X'07FA')	Error in an ACS routine; unexpected clock time.
2043 (X'07FB')	Error in an ACS routine; the work area for the FILTER macro was too small.
2500 (X'09C4')	Error in the ACS executor; execution was unsuccessful because of add error recovery.

ACS translator reason codes

Reason Code	Description
3050 (X'0BEA')	Invalid control block ID.
3051 (X'0BEB')	Invalid control block version.
3052 (X'0BEC')	Invalid listing line length.
3053 (X'0BED')	Invalid listing page length.
3100 (X'0C1C')	Translation unsuccessful because of errors in the source routine.
3101 (X'0C1D')	Error adding local error routine.
3102 (X'0C1E')	Error when opening source data set.
3103 (X'0C1F')	Error when closing source data set.
3104 (X'0C20')	I/O error on source data set.
3105 (X'0C21')	Error when opening listing data set.
3106 (X'0C22')	Error when closing listing data set.
3107 (X'0C23')	I/O error on listing data set.

Reason Code	Description
3108 (X'0C24')	Unable to load I/O routine IGDACTIO.
3109 (X'0C25')	Unable to delete I/O routine IGDACTIO.
3110 (X'0C26')	The source data set DD was not specified.
3111 (X'0C27')	Invalid LRECL for the listing data set.
3112 (X'0C28')	Invalid use of member name with a sequential data set.
3113 (X'0C29')	No member name specified for a PDS.
3114 (X'0C2A')	A nonexistent member name was specified for a PDS.
3115 (X'0C2B')	An open for BLDL failed.
3116 (X'0C2C')	BLDL failed.
3118 (X'0C2E')	An unexpected STACK EMPTY occurred; unable to pop the label.
3119 (X'0C2F')	An unexpected STACK FULL occurred; unable to push the label.
3121 (X'0C31')	The parser recognized a STACK OVERFLOW.
3127 (X'0C37')	Invalid DSORG for the source data set.
3128 (X'0C38')	Invalid DSORG for the listing data set.
3129 (X'0C39')	Error when reading the JFCB for the source data set.
3130 (X'0C3A')	Error when reading the JFCB for the listing data set.
3131 (X'0C3B')	RACF abend; not authorized to the source data set.
3132 (X'0C3C')	RACF abend; not authorized to the listing data set.
3133 (X'0C3D')	Invalid token grammar.
3134 (X'0C3E')	TOKNR recognized a STACK OVERFLOW.
3135 (X'0C3F')	Invalid routine type.
3136 (X'0C40')	Invalid ITEXT operation code.
3137 (X'0C41')	A branch was expected, but was not found.
3138 (X'0C42')	Unresolved label reference; The internal label count limit of 65355 has been exceeded.
3150 (X'0C4E')	X'37' ABEND; out of space.
3201 (X'0C81')	Unknown semantic action was encountered in IGDACTST.
3202 (X'0C82')	Unknown semantic action was encountered in IGDACTFL.
3203 (X'0C83')	Unknown semantic action was encountered in IGDACTIF.
3204 (X'0C84')	Unknown semantic action was encountered in IGDACTPE.
3205 (X'0C85')	Unknown semantic action was encountered in IGDACTSW.
3206 (X'0C86')	Unknown semantic action was encountered in IGDACTEX.
3207 (X'0C87')	Unknown semantic action was encountered in IGDACTWM.
3208 (X'0C88')	Unknown semantic action was encountered in IGDACTCN.
3209 (X'0C89')	Unknown semantic action was encountered in IGDACTCP.
3210 (X'0C8A')	Unknown semantic action was encountered in IGDACTCO.
3211 (X'0C8B')	Unknown semantic action was encountered in IGDACTPA.
3212 (X'0C8C')	Unknown semantic action was encountered in IGDACTCB.
3213 (X'0C8D')	Unknown semantic action was encountered in IGDACTCB.
3301 (X'0CE5')	Unknown literal input from the parser to ACTFL.
3302 (X'0CE6')	Unexpected call to ACTFL during INCLUDE/EXCLUDE processing.

Reason Code	Description
3310 (X'0CEE')	Unexpected input from the parser to ACTPE.
3323 (X'0CFB')	Unexpected TOKEN_CODE from the parser.
3324 (X'0CFC')	Unexpected VALUE_TYPE from the parser.
3325 (X'0CFD')	Unexpected relational operator from the parser.
3326 (X'0CFE')	Unexpected VALUE_TYPE from the parser.

Catalog services reason codes

Reason Code	Description
4000 (X'0FA0')	Locate was not performed; the volume is not SMS-managed.
4001 (X'0FA1')	Catalog locate processing error.
4002 (X'0FA2')	Locate failed for the data set referenced within the scope of JOBCAT/STEPCAT.
4003 (X'0FA3')	Non-SMS-managed data set resides on SMS-managed volumes.
4004 (X'0FA4')	A non-SMS-managed data set was found, but an SMS-managed volume serial was specified.
4006 (X'0FA6')	Volume serials specified are not all in the same tape library.
4007 (X'0FA7')	Volume serials specified are both tape and DASD.
4050 (X'0FD2')	Both the SPAF and SJF tokens were specified; they are mutually exclusive.
4051 (X'0FD3')	The locate was not performed; special, uncataloged system data set.
4052 (X'0FD4')	SJF processing error.
4053 (X'0FD5')	JES3 spool access facility (SPAF) processing error.
4054 (X'0FD6')	Locate failure; the data set was not found.
4055 (X'0FD7')	SMS-managed data set resides on non-SMS-managed volumes.
4056 (X'0FD8')	Error from LCS CBRXVOL.
4100 (X'1004')	The LABEL JCL parameter indicates input processing only and DISP=MOD is specified. This is not valid.
4101 (X'1005')	An LCS error occurred while trying to obtain device pool names.
4107 (X'100B')	The volume list (as specified by the user or obtained from the catalog) contains a combination of DASD and library tape volume serials.
4108 (X'100C')	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 (X'100D')	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 (X'100E')	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 (X'1324')	Error from BPXXSYS(STA) call for z/OS UNIX system services data set.
4901 (X'1325')	Null path name specified.

Construct access services reason codes

Reason Code	Description
5000 (X'1388')	Invalid parameter list.
5001 (X'1389')	Insufficient space was passed by the calling program to contain the requested element.
5002 (X'138A')	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 (X'138B')	No volumes are associated with the storage group.
5004 (X'138C')	Volume verification; all volumes are defined to SMS, but do not belong to the same storage group.
5005 (X'138D')	Invalid user of provided data area.
5006 (X'138E')	The volume list is invalid or inconsistent with the active configuration.
5007 (X'138F')	The volume definition in the active configuration is associated with a nonexistent storage group.
5008 (X'1390')	The count is required, but was not provided.
5009 (X'1391')	Unable to acquire sufficient space in the user's address space.
5010 (X'1392')	Volume verification; all volumes in the list are not defined to SMS.
5011 (X'1393')	The volume verification list contains both volumes defined to SMS and undefined volumes.
5012 (X'1394')	Extended interface; invalid subfunction specified.
5013 (X'1395')	Caller requested specific release, and item does not exist in that release.
5014 (X'1396')	Caller requested specific release, and length specified is invalid.
5015 (X'1397')	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 (X'1771')	Function code out of range.
6002 (X'1772')	Invalid access token.
6003 (X'1773')	Data set not accessed.
6004 (X'1774')	Data set not formatted.
6005 (X'1775')	Data set accessed in read mode.
6006 (X'1776')	Invalid type specified by the caller.
6007 (X'1777')	Error detected by error recovery.
6008 (X'1778')	COMMDS must be accessed in update mode.
6009 (X'1779')	CDS in memory is bad (for example, due to an abend); reaccess or refresh is required.

Reason Code	Description
6010 (X'177A')	CDS cannot be named "ACTIVE".
6030 (X'178E')	Data set not found.
6031 (X'178F')	Caller has insufficient authority.
6032 (X'1790')	CI size is invalid (not 4K).
6033 (X'1791')	Access token not initialized.
6034 (X'1792')	Unable to perform access.
6035 (X'1793')	Access mode is read; data set is empty.
6036 (X'1794')	Invalid data set type.
6037 (X'1795')	A required resource for accessing the data set is unavailable.
6038 (X'1796')	A list ID field is invalid.
6039 (X'1797')	DIV was unable to process the VSAM SHAREOPTIONS.
6040 (X'1798')	Data set previously updated.
6041 (X'1799')	Unable to perform save.
6042 (X'179A')	Data set size has changed (COMMDS).
6043 (X'179B')	Reserve required and not held.
6044 (X'179C')	Same active ACDS specified for message IGD069
6050 (X'17A2')	Data set is invalid. Run ISMF VALIDATE.
6051 (X'17A3')	Unable to perform unaccess.
6052 (X'17A4')	CDS is in 8-name mode and the SMSplex is in 32 name mode; conversion was not requested.
6053 (X'17A5')	The CDS is in 32-name mode and the SMSplex is in 8-name mode.
6054 (X'17A6')	Save failed; another system converted the configuration or COMMDS to 32-name mode.
6055 (X'17A7')	No systems defined to match the system name.
6056 (X'17A8')	CDS formatted at higher level, operation failed. Edit to CDS that is formatted at higher release level is not allowed, but CDS was successfully accessed, CDS need to be unaccessed. Retry the operation on system at the formatted or higher level.
6058 (X'17AA')	Device services failed to return UCB for DD.

Reason Code	Description
6059 (X'17AB')	Data set size changed Indicates high access activity.
6060 (X'17AC')	Unable to perform reserve.
6061 (X'17AD')	Unable to perform release.
6062 (X'17AE')	SVC99 error; error detected by SMS.
6063 (X'17AF')	SVC99 error; dynamic allocation or unallocation.
6064 (X'17B0')	Unable to find UCB to perform reserve or release.
6065 (X'17B1')	SVC99 parameter list is invalid.
6066 (X'17B2')	Environmental error prevented dynamic allocation or the operation in progress.
6067 (X'17B3')	DIV failure; undeterminable failure.
6068 (X'17B4')	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 (X'17B5')	DIV failure; I/O error. See the system console for IEC messages that might be related to the error.
6070 (X'17B6')	No matching elements found.
6071 (X'17B7')	Element not found.
6072 (X'17B8')	Element was replaced.
6073 (X'17B9')	Invalid item count; must be 1.
6074 (X'17BA')	Invalid item length.
6075 (X'17BB')	Insufficient storage.
6080 (X'17C0')	Data set is in use.
6081 (X'17C1')	ENQ failed for data set.
6082 (X'17C2')	DEQ failed for data set.
6090 (X'17CA')	ACS routine not found.
6091 (X'17CB')	Data set is invalid.
6092 (X'17CC')	BCDEFLEN is zero.
6093 (X'17CD')	BCDEFLEN is bad.

Reason Code		Description
6094	(X'17CE')	Extension length is zero.
6095	(X'17CF')	Extension length is bad.
6096	(X'17D0')	Extension offset is zero.
6097	(X'17D1')	Extension offset is bad.
6098	(X'17D2')	Extension offset is out of ascending order.
6099	(X'17D3')	Extension lengths plus definition length do not add up to total length specified in header.
6101	(X'17D5')	Function not allowed for the active configuration.
6102	(X'17D6')	Subsystem not active; no active configuration available.
6103	(X'17D7')	Error detected by SSI.
6104	(X'17D8')	Access to active configuration not allowed in update mode.
6105	(X'17D9')	Conditional GETMAIN failed.
6110	(X'17DE')	System or system group name not found.
6111	(X'17DF')	System name or system group already exists.
6112	(X'17')	System/system group name list is full.
6113	(X'170')	No base configuration information found.
6114	(X'1700')	System name type specified is invalid
6130	(X'17F2')	Data set is not a linear data set.
6131	(X'17F3')	Access failed; the data set type is incorrect. The referenced data set was found to be a different type (ACDS, SCDS, COMMDS) than the type requested.
6132	(X'17F4')	COPYSCDS command input SCDS dataset is an ACDS dataset.
6133	(X'17F5')	COPYSCDS command input SCDS dataset is not formatted.
6134	(X'17F6')	COPYSCDS command input SCDS dataset is invalid.
6135	(X'17F7')	COPYSCDS command target dataset is ACTIVE ACDS or is same as the source dataset.
6136	(X'17F8')	Your configuration data set cannot be an extended format linear data set.
6700	(X'1A2C')	Object or object backup storage group connected to system group.

Reason Code	Description
6701 (X'1A2D')	Object or object backup storage group connected to more than one system.
6702 (X'1A2E')	Optical drive connected to system group.
6703 (X'1A2F')	Optical drive connected to more than one system.
6704 (X'1A30')	SG: Breakpoint value invalid.
6705 (X'1A31')	SG: Track Alloc Threshold value invalid.
6706 (X'1A32')	SC: OAM Enhancement error.
6707 (X'1A33')	Duplicate volume found in target CDS.
6708 (X'1A34')	Invalid alert threshold value specified
6709 (X'1A35')	Serialization error – invalid action.
6710 (X'1A36')	Processing priority invalid.
6711 (X'1A37')	Invalid RMODE31 value specified.
6712 (X'1A38')	MC: Invalid Volume Set Management Level Value.
6713 (X'1A39')	MC: Invalid Retention Method Value.
6714 (X'1A3A')	MC: If Retention Method is set to VRSEL, the fields "Volume Set Management Level" and "Retain While Cataloged" should not be set.
6715 ('X1A3B')	MC: If Retention Method is set to EXPDT, the field "Exclude from VRSEL" should not be set.
6732 (X'1A4C')	Invalid DS recovery value.
6733 (X'1A4D')	Invalid PPRC recovery value.
6734 (X'1A4E')	Invalid PPRC back value.
6735 (X'1A4F')	Invalid HFRR value.
6736 (X'1A50')	SGCL: No SG in collection.
6737 (X'1A51')	SGCL: Catalog list maximum exceeded.
6738 (X'1A52')	SGCL: Catalog name syntax incorrect.
6739 (X'1A53')	SGCL: SG array not sorted.
6740 (X'1A54')	Unknown storage group collection type SGCTYPE.

Reason Code		Description
6741	(X'1A55')	COPYPOOL name contains an incorrect length.
6742	(X'1A56')	COPYPOOL name contains incorrect syntax.
6743	(X'1A57')	Maximum number of copy pool storage groups exceeded.
6744	(X'1A58')	A storage group name contains incorrect syntax.
6745	(X'1A59')	A storage group name contains an incorrect length.
6746	(X'1A5A')	Incorrect number of versions specified.
6747	(X'1A5B')	No SGs were specified for this storage group collection.
6748	(X'1A5C')	COPYPOOL BACKUP storage group name contains incorrect syntax or length.
6749	(X'1A5D')	Only POOL storage groups can specify a COPYPOOL BACKUP storage group.
6750	(X'1A5E')	Invalid DS separation profile name.
6751	(X'1A5F')	Invalid OVERFLOW value.
6752	(X'1A60')	Invalid extend SG name.
6753	(X'1A61')	Internal SMS parm list IGDIOPL contains incorrect information.
6754	(X'1A62')	DS separation profile record format is not fixed.
6755	(X'1A63')	Open of DS separation profile failed.
6756	(X'1A64')	Space constraint relief must also be specified when maximum volume count is specified.
6757	(X'1A65')	Invalid maximum volume count value.
6758	(X'1A66')	SGCL: Missing dump class when auto dump is ON.
6759	(X'1A67')	SGCL: Dump sys/sys group name syntax incorrect.
6760	(X'1A68')	SGCL: Dump class name syntax incorrect.
6761	(X'1A69')	Base: Lock set length invalid.
6762	(X'1A6A')	Base: Lock set name invalid.
6763	(X'1A6B')	Base: Lock structure length invalid.
6764	(X'1A6C')	Base: Lock structure name invalid.

Reason Code	Description
6765 (X'1A6D')	Base: Lock set count invalid.
6766 (X'1A6E')	Base: Lock set point is 0.
6767 (X'1A6F')	Base: Lock structure list out of order.
6768 (X'1A70')	Base: Lock structure index invalid.
6769 (X'1A71')	Base: Lock structure pointer is 0.
6770 (X'1A72')	SC: Lock set length invalid.
6771 (X'1A73')	SC: Lock set name invalid.
6772 (X'1A74')	DC: Invalid media type or recording technique for key.
6773 (X'1A75')	DC: Key label and key code set mismatch (that is, *keylabel1* specified without specifying *keycode1*).
6774 (X'1A76')	DC: Invalid key label length. Maximum length is 64 characters.
6776 (X'1A78')	DC: Invalid key code value. Must be 'L' or 'H'.
6777 (X'1A79')	DC: SMBVSP value invalid.
6778 (X'1A7A')	DC: Override without SPACE.
6779 (X'1A7B')	DC: EATTR value invalid.
6780 (X'1A7C')	Separation profile unexpected end of file. Profile was modified during SMS configuration activation.
6781 (X'1A7D')	Separation profile number of sep groups mismatch. Profile was modified during SMS configuration activation.
6782 (X'1A7E')	Separation profile number of sep group DSNs mismatch. Profile was modified during SMS configuration activation.
6783 (X'1A7F')	Error accessing data set separation profile. See message IGD06031I.
6800 (X'1A90')	DC: NAME invalid.
6801 (X'1A91')	RECORG is invalid.
6802 (X'1A92')	RECFM is invalid.
6803 (X'1A93')	Carriage control is invalid.
6804 (X'1A94')	LRECL is invalid.
6805 (X'1A95')	Key length is invalid.

Reason Code		Description
6806	(X'1A96')	Key off is invalid.
6807	(X'1A97')	Retention period is invalid.
6808	(X'1A98')	Expiration day/date is invalid.
6809	(X'1A99')	Primary space is invalid.
6810	(X'1A9A')	Secondary space is invalid.
6811	(X'1A9B')	Directory blocks space is invalid.
6812	(X'1A9C')	Volume count is invalid.
6813	(X'1A9D')	CI size is invalid.
6814	(X'1A9E')	CI free percent is invalid.
6815	(X'1A9F')	CA free percent is invalid.
6816	(X'1AA0')	VSAM share options is invalid.
6817	(X'1AA1')	Name is invalid for management class.
6818	(X'1AA2')	Minimum days on primary is invalid.
6819	(X'1AA3')	Retain days archival is invalid.
6820	(X'1AA4')	Expiration format is invalid.
6822	(X'1AA6')	Backup frequency is invalid.
6823	(X'1AA7')	Backup versions is invalid.
6824	(X'1AA8')	Backup retention is invalid.
6825	(X'1AA9')	Backup retention/no primary is invalid.
6826	(X'1AAA')	Maximum retention is invalid.
6827	(X'1AAB')	Name is invalid for storage class.
6828	(X'1AAC')	Millisecond response is invalid.
6829	(X'1AAD')	Sequential kilobyte response is invalid.
6830	(X'1AAE')	Name is invalid for storage group.

Reason Code	Description
6831 (X'1AAF')	Storage group type is invalid.
6832 (X'1AB0')	VIOMAX is invalid.
6833 (X'1AB1')	VIOUNIT is invalid.
6834 (X'1AB2')	System names are invalid.
6835 (X'1AB3')	System statuses are invalid.
6836 (X'1AB4')	Thresholds are invalid.
6837 (X'1AB5')	Default management class is invalid.
6838 (X'1AB6')	Volser is invalid.
6839 (X'1AB7')	SMS status is invalid.
6840 (X'1AB8')	MVS status is invalid.
6841 (X'1AB9')	Name length is invalid.
6842 (X'1ABA')	Unknown storage group for volume.
6843 (X'1ABB')	Storage group is type VIO, OBJECT, or OBJECT BACKUP or TAPE.
6844 (X'1ABC')	Volser length not equal to 6.
6845 (X'1ABD')	Duplicate system names found.
6846 (X'1ABE')	Allocation unit less than 1.
6847 (X'1ABF')	Scale factor (AVGREC) invalid.
6848 (X'1AC0')	Availability is invalid.
6849 (X'1AC1')	Direct response is invalid.
6850 (X'1AC2')	Direct bias is invalid.
6851 (X'1AC3')	Sequential response is invalid.
6852 (X'1AC4')	Sequential bias is invalid.
6853 (X'1AC5')	Level 1 days is invalid.
6854 (X'1AC6')	Command/automatic migrate is invalid.

Reason Code	Description
6855 (X'1AC7')	Number of GDG elements is invalid.
6856 (X'1AC8')	Data set deleted; backup version is invalid.
6857 (X'1AC9')	Administrator/end user backup is invalid.
6858 (X'1ACA')	RECORG and RECFM are mutually exclusive.
6859 (X'1ACB')	Autobackup system is invalid.
6860 (X'1ACC')	Autodump system is invalid.
6861 (X'1ACD')	Automigrate system is invalid.
6862 (X'1ACE')	Reason code is no longer used.
6863 (X'1ACF')	Reason code is no longer used.
6864 (X'1AD0')	Reason code is no longer used.
6865 (X'1AD1')	Minimum days on primary is required.
6866 (X'1AD2')	Level 1 days required.
6867 (X'1AD3')	Backup versions is required.
6868 (X'1AD4')	Backup versions (data set delete) is required.
6869 (X'1AD5')	Backup frequency is required.
6870 (X'1AD6')	Backup retention is required.
6871 (X'1AD7')	Retain days only backup is required.
6872 (X'1AD8')	MCPREL and MCPRECN are mutually exclusive.
6873 (X'1AD9')	Incorrect media type specified with performance scaling
6874 (X'1ADA')	Scaling and segmentation are mutually exclusive.
6875 (X'1ADB')	Incorrect media type specified with performance segmentation.
6880 (X'1AE0')	DSNTYPE is invalid.
6881 (X'1AE1')	RECORG and DSNTYPE are mutually exclusive.
6882 (X'1AE2')	Backup destination class is invalid.

Reason Code	Description
6683 (X'1A1B')	Expiration action is invalid.
6884 (X'1AE4')	Retain days archive is invalid.
6885 (X'1AE5')	Name is invalid.
6886 (X'1AE6')	Expiration date is invalid.
6887 (X'1AE7')	Destination is invalid.
6888 (X'1AE8')	Data set prefix is invalid.
6889 (X'1AE9')	Instruction DSN is invalid.
6890 (X'1AEA')	Instruction member is invalid.
6891 (X'1AEB')	Data set name is invalid.
6892 (X'1AEC')	Data set member invalid.
6893 (X'1AED')	Destination is required.
6894 (X'1AEE')	Output prefix is required.
6895 (X'1AEF')	Instruction DSN is required.
6896 (X'1AF0')	One DSN is required.
6897 (X'1AF1')	Guaranteed backup frequency is invalid.
6898 (X'1AF2')	Guaranteed backup frequency is required with AUTOBACKUP=Y.
6899 (X'1AF3')	Invalid block size limit value.
6900 (X'1AF4')	Time Since Creation value is invalid.
6901 (X'1AF5')	Time Since Last Use value is invalid.
6902 (X'1AF6')	Periodic value is invalid.
6903 (X'1AF7')	Group ID is not unique.
6904 (X'1AF8')	Object Cycle Start is invalid.
6905 (X'1AF9')	Object Cycle End is invalid.
6906 (X'1AFA')	Initial Access is invalid.

Reason Code	Description
6907 (X'1AFB')	Library name is invalid.
6908 (X'1AFC')	Library type is invalid.
6909 (X'1AFD')	Drive name is invalid.
6910 (X'1AFE')	Drive type is invalid.
6911 (X'1AFF')	Time Since Creation and Time Since Last Use fields are mutually exclusive.
6912 (X'1B00')	Time Since Creation and Periodic fields are mutually exclusive.
6913 (X'1B01')	Time Since Last Use and Periodic fields are mutually exclusive.
6914 (X'1B02')	Periodic Monthly and Periodic Quarterly fields are mutually exclusive.
6915 (X'1B03')	Periodic Monthly and Periodic Yearly fields are mutually exclusive.
6916 (X'1B04')	Periodic Quarterly and Periodic Yearly fields are mutually exclusive.
6917 (X'1B05')	Library status is invalid.
6918 (X'1B06')	Drive status is invalid.
6919 (X'1B07')	Group qualifier is invalid.
6920 (X'1B08')	Duplicate library names in one storage group.
6921 (X'1B09')	Storage Group INTERVAL MIGRATION cannot be selected unless AUTO MIGRATION has also been selected.
6922 (X'1B0A')	Copy technique is invalid in management class.
6923 (X'1B0B')	Accessibility is invalid in storage class.
6924 (X'1B0C')	Library name length is invalid in storage group.
6925 (X'1B0D')	Library name is invalid in storage group.
6926 (X'1B0E')	Primary space management requested without auto migration in storage group.
6927 (X'1B0F')	Library name length is nonzero or library name found in a storage that is not of type OBJECT, OBJECT BACKUP, or TAPE.
6941 (X'1B1D')	Entry default volume use attribute is invalid in library.
6942 (X'1B1E')	Eject default is invalid in library.
6943 (X'1B1F')	Library ID in library configuration database is invalid in library.

Reason Code	Description
6944 (X'1B20')	Library device type is invalid in library.
6945 (X'1B21')	Entry default unit name is invalid in library.
6946 (X'1B22')	Media type is invalid in data class.
6947 (X'1B23')	Recording technique is invalid in data class.
6948 (X'1B24')	Compaction type is invalid in data class.
6949 (X'1B25')	Default entry data class is invalid in library.
6950 (X'1B26')	Management class name length is invalid in aggregate group.
6951 (X'1B27')	Management class name is invalid in aggregate group.
6952 (X'1B28')	SES cache length is invalid in base configuration.
6953 (X'1B29')	SES cache name is invalid in base configuration.
6954 (X'1B2A')	Cache length is invalid in base configuration.
6955 (X'1B2B')	Cache name is invalid in base configuration.
6956 (X'1B2C')	SES cache count is invalid in base configuration.
6957 (X'1B2D')	SES cache pointer is zero in base configuration.
6958 (X'1B2E')	Pointer to aggregate backup parameters is 0 in management class.
6959 (X'1B2F')	Aggregate backup name is invalid in management class.
6960 (X'1B30')	Aggregate backup name length is invalid in management class.
6961 (X'1B31')	Destination name is invalid in management class.
6962 (X'1B32')	Destination name length is invalid in management class.
6965 (X'1B35')	Indication of number of aggregate backup versions is invalid in management class.
6966 (X'1B36')	Indication of retention period for only version is invalid in management class.
6967 (X'1B37')	Indication of retention period for extra versions is invalid in management class.
6968 (X'1B38')	Copy serialization option is invalid in management class.
6969 (X'1B39')	Retain only version is invalid in management class.

Reason Code	Description
6970 (X'1B3A')	Retain extra versions is invalid in management class.
6971 (X'1B3B')	Number of versions to be backed up is invalid in management class.
6972 (X'1B3C')	Destination pointer is zero.
6973 (X'1B3D')	Incorrect number of destinations.
6974 (X'1B3E')	Destination length is incorrect.
6975 (X'1B3F')	Destination name is incorrect.
6976 (X'1B40')	Destination name length is incorrect.
6977 (X'1B41')	Destination name is incorrect.
6978 (X'1B42')	Time is incorrect.
6979 (X'1B43')	NetView® class is incorrect.
6980 (X'1B44')	File encryption is incorrect.
6981 (X'1B45')	Transmit accompany files incorrect.
6982 (X'1B46')	Transmit control information is incorrect.
6983 (X'1B47')	Number of copies is not specified in aggregate group.
6984 (X'1B48')	Number of copies is invalid in aggregate group.
6985 (X'1B49')	Aggregate copy technique is invalid in management class.
6990 (X'1B4E')	SES cache list is not in ascending order in base configuration.
6991 (X'1B4F')	Cache set name is invalid in storage class.
6992 (X'1B50')	Cache set name length is invalid in storage class.
6993 (X'1B51')	Direct CF weight is invalid in storage class.
6994 (X'1B52')	Sequential CF weight is invalid in storage class.
6995 (X'1B53')	SES cache index is invalid in base configuration.
6996 (X'1B54')	Cache set pointer is zero in base configuration.
8000 (X'1F40')	Cloud construct validation error. Construct name length is invalid.

Reason Code	Description
8001 (X'1F41')	Cloud construct validation error. Construct name syntax is invalid.
8002 (X'1F42')	Cloud construct validation error. Identity is invalid.
8003 (X'1F43')	Cloud construct validation error. Cloud provider is invalid.
8004 (X'1F44')	Cloud construct validation error. URI endpoint is invalid.
8005 (X'1F45')	Cloud construct validation error. SSL version is invalid.
8006 (X'1F46')	Cloud construct validation error. SSL key is invalid.

Disposition processing reason codes

Reason Code	Description
7000 (X'1B58')	Invalid parameter list.
7001 (X'1B59')	Failure in catalog define of roll-in GDG.
7002 (X'1B5A')	Unable to find the passed DDNAME.
7004 (X'1B5C')	No volumes provided to catalog a data set on a managed, mountable volume.
7900 (X'1EDC')	Null path name provided.
7901 (X'1EDD')	Call to BPXXSYSC(UNL) failed to open for z/OS UNIX system services data set.
7902 (X'1EDE')	Invalid disposition specified for z/OS UNIX system services.
7998 (X'1F3E')	Enhanced volume list required to catalog a data set on a managed, mountable volume.
7999 (X'1F3F')	Processing not supported for data sets on managed, mountable volumes with disposition other than CATLG.

Intersystem communications reason codes

Reason Code	Description
10000 (X'2710')	Operator specified cancel.
10001 (X'2711')	Operator specified suspend.
10002 (X'2712')	Operator specified terminate.
10003 (X'2713')	Operator specified not to convert the COMMDS to 32 name mode.
10004 (X'2714')	Missing storage group, CDS is corrupted, module IGDICMC1
10050 (X'2742')	Internal COMMDS unavailable.
10051 (X'2743')	Invalid user return area.
10052 (X'2744')	Null configuration is active.
10053 (X'2745')	COMMDS was switched.
10054 (X'2746')	Device services failed to return UCB for DD.

Reason Code	Description
10055 (X'2747')	Unable to find UCB for access.
10100 (X'2774')	Invalid IGDICMRT control block.
10101 (X'2775')	Invalid current IGDICMTK control block.
10150 (X'27A6')	COMMDS validation: Invalid activation level.
10151 (X'27A7')	COMMDS validation: Invalid update count.
10152 (X'27A8')	COMMDS validation: Invalid volume count.
10153 (X'27A9')	COMMDS validation: Invalid volume length.
10154 (X'27AA')	COMMDS validation: invalid volume system data length.
10155 (X'27AB')	COMMDS validation: invalid system data length.
10200 (X'27D8')	Invalid IGDICMS0 function code.
10201 (X'27D9')	Invalid IGDICMC1 function code.
10202 (X'27DA')	Invalid IGDICMAC function code.
10203 (X'27DB')	Invalid IGDICMU1 function code.
10250 (X'280A')	SMS resource is unavailable.
10251 (X'280B')	SMS resource not serialized.
10300 (X'283C')	Intersystem communications abend occurred.
10301 (X'283D')	Null COMMDS accessed.
10302 (X'283E')	Null COMMDS specified.
10303 (X'283F')	Operational services ACDS activation failure.
10304 (X'2840')	Your current COMMDS chain is cyclic.

Interpreter/Dynamic allocation reason codes

Reason Code	Description
11000 (X'2AF8')	Internal IDAX logic error.
11001 (X'2AF9')	Invalid input condition; the SJF token is always required.
11007 (X'2AFF')	Device Pool Select SSI error.
11008 (X'2B00')	Request was not for retrieve.
11009 (X'2B01')	Job or step request.
11010 (X'2B02')	SJF failure occurred with return code 0.
11011 (X'2B03')	SJF failure occurred with return code 4.
11012 (X'2B04')	Request was not for find.
11013 (X'2B05')	REFDD related request.
11014 (X'2B06')	SJF error.
11015 (X'2B07')	Indeterminate unallowable error.
11016 (X'2B08')	LCS CRBXVOL service failed.
11100 (X'2B5C')	RECFM cannot be specified with RECORG.
11101 (X'2B5D')	DSNTYPE cannot be specified with RECORG.
11102 (X'2B5E')	Invalid DSNTYPE version number.

Message services reason codes

Reason Code		Description
12000	(X'2EE0')	The cell eye-catcher was not found.
12001	(X'2EE1')	An invalid value was specified for FREEMSG.
12002	(X'2EE2')	Unable to stack local recovery routine.
12003	(X'2EE3')	The MCSCH.VERSION is incorrect.
12004	(X'2EE4')	The MCSRS.VERSION is incorrect.
12005	(X'2EE5')	The MCSMCBID field is incorrect.
12006	(X'2EE6')	The MCSMCBVR field is incorrect.
12007	(X'2EE7')	The MSGPTR is invalid.
12008	(X'2EE8')	An unexpected abend occurred.
12009	(X'2EE9')	Invalid MLTYPE code.
12010	(X'2EEA')	The message length was too long.
12011	(X'2EEB')	The message chain eye-catcher was not found.
12012	(X'2EEC')	Unable to create recovery environment.
12020	(X'2EF4')	The message ID does not exist in the message index.
12021	(X'2EF5')	The subpool ID is invalid.
12022	(X'2EF6')	The variant number is invalid.
12023	(X'2EF7')	The REUSE value is invalid.
12024	(X'2EF8')	The PREFIX value is invalid.
12025	(X'2EF9')	The INCLUD value is invalid.
12026	(X'2EFA')	The required input parameters were not specified.
12027	(X'2EFB')	The SSSA control block ID is invalid.
12028	(X'2EFC')	The SSSA control block version is invalid.
12029	(X'2EFD')	REUSE and START_NEW_CHN are mutually exclusive.
12030	(X'2EFE')	SSSARMSG is set to prevent returning messages.
12031	(X'2EFF')	The SSSA is invalid.
12032	(X'2F00')	CPOOL abended.
12033	(X'2F01')	The message definition does not exist in any message module.
12040	(X'2F08')	The value for SMS construct or group name is too long.
12041	(X'2F09')	INCLUD = N and PREFIX = Y are mutually exclusive.
12042	(X'2F0A')	The required variable parameters were not specified.
12043	(X'2F0B')	Invalid TEXT type code.
12044	(X'2F0C')	Invalid DATA type code.
12060	(X'2F1C')	The descriptor code could not be determined.
12061	(X'2F1D')	HRDCPY is invalid for MLWTO message.
12062	(X'2F1E')	Invalid WTO type code in the message ID.
12063	(X'2F1F')	A multiline message was provided for a WTOR request.

Reason Code		Description
12064	(X'2F20')	A WTO abend occurred.
12065	(X'2F21')	A TPUT abend occurred.
12066	(X'2F22')	A RECORD abend occurred.
12067	(X'2F23')	The number of message lines exceeded the maximum.
12068	(X'2F24')	The pointer to the message is zero.
12069	(X'2F25')	Failed to pass check necessary for SVC35 after switching state and key.

Operational services reason codes

Reason Code	Description
13000 (X'32C8')	Not authorized to activate a configuration (RACF).
13001 (X'32C9')	Too many request elements are on the queue.
13002 (X'32CA')	The requested configuration no longer exists.
13003 (X'32CB')	An invalid request element was found on the queue.
13004 (X'32CC')	The active configuration was not found on the configuration chain.
13005 (X'32CD')	Too many configurations are on the configuration chain.
13006 (X'32CE')	Invalid configuration.
13007 (X'32CF')	Invalid parameter list.
13008 (X'32D0')	Invalid configuration root.
13009 (X'32D1')	No active configuration.
13010 (X'32D2')	Unexpected return code from SAF.
13011 (X'32D3')	Operator denied authorization to activate a configuration.
13012 (X'32D4')	Event audit failed.
13013 (X'32D5')	Status for update library on drive must be online or offline.
13014 (X'32D6')	Library or drive not found. Update failed.
13015 (X'32D7')	EXTRACT LIBRARY/DRIVE failed. Unexpected return/reason code from module IGDOPSD1.
13016 (X'32D8')	LIBRARY/DRIVE UPDATED failed. Unexpected return/reason code module IGDOPS10
13017 (X'32D9')	Library definition exists, no drive definitions exist, module IGDOPST1.
13018 (X'32DA')	Missing storage group, CDS is corrupted, module IGDOPSC1

Scheduling services reason codes

Reason Code	Description
14000 (X'36B0')	SMS resource connectivity conflict (causes the job to fail).
14001 (X'36B1')	A subfunction other than SSSAPLCO or SSSASCHD was specified.
14002 (X'36B2')	The saved scheduling information inconsistent with the current SMS configuration.
14003 (X'36B3')	A new list of system names has been determined.

Reason Code	Description
14004 (X'36B4')	The job has no SMS resource requirements.
14005 (X'36B5')	The returned system list is restricted because of unavailable resources.
14006 (X'36B6')	JES3 spool access facility (SPAF) read error.
14007 (X'36B7')	A SPAF token was not specified.
14008 (X'36B8')	SMS resources are temporarily unavailable (reschedule the job).
14009 (X'36B9')	The SPAF subsystem name was not specified.
14010 (X'36BA')	Volume not in active configuration.
14011 (X'36BB)	Invalid eye catcher in record
14012 (X'36BC')	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 (X'3A98')	PSW supervisor state.
15001 (X'3A99')	PSW problem state.
15010 (X'3AA2')	Initialization failed because of an MVS service.
15011 (X'3AA3')	Initialization failed because of an SMS error.
15012 (X'3AA4')	SMS is not defined to MVS as a subsystem.
15013 (X'3AA5')	Module IGDSSI00 has already been activated.
15014 (X'3AA6')	Invalid SSIVT.
15015 (X'3AA7')	Invalid SSIAT.
15016 (X'3AA8')	SMS address space verification failed.
15017 (X'3AA9')	Member IGDSMSxx does not exist.
15018 (X'3AAA')	AXSET failed in IGDSSI01.
15019 (X'3AAB')	LXRES failed in IGDSSI01.
15020 (X'3AAC')	ETCON failed in IGDSSI01.
15021 (X'3AAD')	ETCRE failed in IGDSSI01.
15022 (X'3AAE')	Missing keyword was claimed, but no keywords were missing.
15023 (X'3AAF')	Did not build the message.
15024 (X'3AB0')	Did not build the message.
15025 (X'3AB1')	Did not build the message.
15026 (X'3AB2')	SMS initialization failed.
15027 (X'3AB3')	IEEMB888 failed.
15028 (X'3AB4')	IEEMB887 failed.
15029 (X'3AB5')	Syntax error.
15100 (X'3AFC')	Length of macro IHADFVT is bad.

VOLREF services reason codes

Reason Code	Description
16000 (X'3E80')	The pointer to the CPL of the referencing data set is 0.
16001 (X'3E81')	The pointer to the CPL of the referenced data set is 0.
16002 (X'3E82')	The referencing data set disposition field contains an invalid value.
16003 (X'3E83')	The referenced data set disposition field contains an invalid value.
16004 (X'3E84')	Unable to build the recovery environment for VOLREF services.
16005 (X'3E85')	VOLREF services internal error.
16006 (X'3E86')	Failure in scheduling information processing for VOLREF services.
16007 (X'3E87')	The subfunction code was not the VOLREF services function code; valid codes are SSSAVOLR and SSSACPSW.
16008 (X'3E88')	SSSAASJ1 is 0.
16009 (X'3E89')	Neither the SJF token nor the CPL pointer is specified for the referenced DD.
16010 (X'3E8A')	SJF retrieval request failed.
16011 (X'3E8B')	SJF update request failed.
16012 (X'3E8C')	Both the SJF and the SPAF tokens are 0.
16013 (X'3E8D')	VOLREF services did not do any processing for this invocation.
16014 (X'3E8E')	Both the SJF and the SPAF tokens are specified.
16015 (X'3E8F')	Unexpected return code from the invoked routine.
16016 (X'3E90')	Delete of VOLREF services error recovery routine failed.
16017 (X'3E91')	A non-SMS-managed data set referenced an SMS-managed data set (VOLREF services).
16018 (X'3E92')	Catalog services failure.
16019 (X'3E93')	A construct access services failure occurred.
16020 (X'3E94')	VOLREF services internal error; DDFLAG value is invalid; valid values are 1 and 2.
16021 (X'3E95')	VOLREF services error recovery routine failed because no VOLREF services global data control block (IGDVRFGD) exists.
16022 (X'3E96')	The referenced data set is not cataloged (VOLREF services).
16023 (X'3E97')	The referencing data set is not cataloged (VOLREF services).
16024 (X'3E98')	The SPAF job token is 0.
16025 (X'3E99')	The SPAF scheduling information token is 0.
16026 (X'3E9A')	CLEANUP is on, SSSAASRA is 0.
16027 (X'3E9B')	VOLREF services internal error; both the SCHEDULE and the ALLOCATE flags are set on in module IGDVRF00.
16028 (X'3E9C')	VOLREF services did not do any processing for a new reference.
16029 (X'3E9D')	VOLREF services internal error; the CLEANUP, SCHEDULE, and ALLOC flags are all 0 in module IGDVRF00.
16030 (X'3E9E')	VOLREF services internal error; invalid SJF function code; valid codes are 0 = CLEANUP, 1 = RETRIEVE, 2 = UPDATE.
16031 (X'3E9F')	No SMS SWBs were copied to the target DD because the source DD has no SMS SWBs.
16032 (X'3EA0')	Processing did not complete for scheduling information because the referenced data set is new or temporary.

Reason Code	Description
16033 (X'3EA1')	Missing the required SJF token for a NEW/MOD data set.
16034 (X'3EA2')	Message service failure.
16035 (X'3EA3')	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 (X'3EA4')	The pointer to the global data block from the SW parameter list is 0.
16037 (X'3EA5')	The pointer to SJF local storage is 0 (in the VOLREF services global data block).
16038 (X'3EA6')	An SJF request to terminate failed.
16039 (X'3EA7')	VOLREF services internal error; the referenced data set has no storage class and the SMS flag is off.
16040 (X'3EA8')	Error; CTGWAVCT = 0 from IGDCAT00 when the return code = 0.
16041 (X'3EA9')	VOLREF internal error. The function code is not acceptable to IGDVRFS1; valid functions are UPDATE or RETRIEVE.
16042 (X'3EAA')	No SJF token was specified.
16043 (X'3EAB')	UPDATE and RETRIEVE are mutually exclusive.
16044 (X'3EAC')	An SJF update failed because SJF retrieve failed.
16045 (X'3EAD')	VOLREF internal error; SJF REQNUM greater than SJF array DIM or less than 1.
16046 (X'3EAE')	Add scheduling records failed.
16047 (X'3EAF')	SMS internal error; SMS class not found for an SMS volume.
16048 (X'3EB0')	SSSAASJ2 is 0.
16049 (X'3EB1')	The caller is neither ALLOCATION nor JES3.
16050 (X'3EB2')	The control block visual ID is not IGDVRMSW.
16051 (X'3EB3')	The pointer to the SJACP is 0.
16052 (X'3EB4')	An unexpected visual ID exists for control block IGDVRMGD; valid values are zeros or IGDVRMGD.
16053 (X'3EB5')	ACS does not allow the storage group which was derived from the referenced data set.
16054 (X'3EB6')	An ACS routine failed.
16055 (X'3EB7')	IGDIDMAR failed.
16056 (X'3EB8')	SSSA5SGN is 0 from an ACS routine.
16057 (X'3EB9')	VOL=REF to GDG is not allowed.
16100 (X'3EE4')	SSSAAVOL and SSSAMVL are mutually exclusive.
16101 (X'3EE5')	Internal error from IDAX.
16102 (X'3EE6')	Error returned by LCS.
16103 (X'3EE7')	LCS indicated volume record not found for managed-mountable data set.
16104 (X'3EE8')	No SJF token provided for referencing data set.
16999 (X'4267')	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'42AB')	Convert CNTPCVNT TTR failed.
17068 (X'42AC')	I/O error while writing EOF mark.
17070 (X'42AE')	Data set allocated as extended format.
17071 (X'42AF')	Data set not allocated as extended format.
17072 (X'42B0')	Extended format requirement could not be met.
17073 (X'42B1')	Allocation as non-extended format will be attempted.
17074 (X'42B2')	Guaranteed space volume list is invalid.
17075 (X'42B3')	Data set DSORG must be RS for data set to be extended format.
17076 (X'42B4')	Non-SMS data sets cannot be extended format.

Reason Code	Description
17077 (X'42B5')	An error has occurred in the storage group/volume selection.
17080 (X'42B8')	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.
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'42E0')	Bad return code from FAMS
17150 (X'43FE')	Data set is encrypted.
17151(X'43FF')	Data set characteristics do not allow encryption.
17152 (X'4300')	Extended Format converted.
17153 (X'4301')	Unable to allocate requested stripe count for encryption.
17154 (X'4302')	Data Set Key Label was found. Allocation failed.
17155(X'4303')	Insufficient authority to create encrypted data.
17156(X'4304')	Data Set characteristics do not allow encryption. Key label is ignored. Request continues.
17160 (X'4308')	Data set is compressible.
17161 (X'4309')	Bad return code from compression services.
17162 (X'430A')	Bad return code from compression services.
17163 (X'430B')	Data set characteristics do not allow compression.
17164 (X'430C')	Compression not allowed by catalog management.
17165 (X'430D')	Multivolume temporary data set cannot be extended format.
17167 (X'430F')	All volumes selected are fast replication-capable and one or more of the volumes is not Preserve Mirror-capable.
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.

Reason Code	Description
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 groups.
17227 (X'434B')	This message is issued when multi-tiered storage groups was requested and the data set was not allocated to the first storage group listed.
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.
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.
17246 (X'435E')	JFCBEXAD=0 (JFCB Extension is not provided by the caller) when volume count is greater than 5.
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.

Reason Code	Description
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.
17265 (X'4371')	ROACCESS=DISALLOW specified or ROACCESS=(ALLOW,TRKLOCK) not specified for read-only volume.
17271 (X'4377')	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')	Volumes 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.
17319 (X'43A7')	Class transition failure
17321 (X'43A9')	Class transition failure, invalid flags
17337 (X'43B9')	Error in AOM SFILIST
17338 (X'43BA')	Cannot allocate data set with requested stripe count.
17339 (X'43BB')	Cannot allocate data set in one single SFI.
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.

Reason Code	Description
17358 (X'43CE')	This message is issued when an attempt to create a new generation of a GDS fails because a generation with the same name exists but has not been rolled-in. The failure occurs because the user has specified GDS_RECLAIM=NO in the IGDSMSxx member of SYS1.PARMLIB or used the SETSMS command to specify GDS_RECLAIM=NO.
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.
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'43DA')	Error in UCBLOOK.
17371 (X'43DB')	Undefined VIO device type.
17372(X'43DC')	Data set separation could not be provided.
17373 (X'43DD')	Unexpected failure during data set separation processing.
17380 (X'43E4')	This message is issued to the hardcopy log when SMS detects that the cumulative space allocated on the storage group used for allocation has exceeded its high allocation threshold. It is issued at the first occurrence or at the first occurrence after the issuance of IGD17381I.
17381 (X'43E5')	This message is issued to the hardcopy log when SMS detects that the cumulative space allocated on the storage group used for allocation has fallen below its low allocation threshold or below 80% of its high allocation threshold. It is issued at the first occurrence after the issuance of IGD17380I.
17395 (X'43F3')	Unable to allocate all volumes of the data set in the same Storage Facility Image
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.

Reason Code	Description
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'445D')	Open failed for a UNIX file.
17502 (X'445E')	Create of a UNIX file failed.
17503 (X'445F')	Close failed for a UNIX file.
17510 (X'4466')	DSNTYPE PIPE specified without PATHNAME.
17511 (X'4467')	No PATHNAME specified.
17512 (X'4468')	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.
17807 (X'458F')	VSAM volume selection failure status message.
17808 (X'4590')	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 (X'0004')	Mismatch between format-1 DSCB and VVR/NVR.
05 (X'0005')	FREEMAIN error (SVC 5).
08 (X'0008')	Error accessing the VTOC (CVAFFILT READ).
12 (X'000C')	Error accessing the VVDS (VVDS Manager).

Reason Code	Description
32 (X'0020')	Parameter list error (eye-catcher, length, UCB address, or function requested).
64 (X'0040')	Major error (ABEND).
120 (X'0078')	GETMAIN error (SVC 120).
139 (X'008B')	CVAF error (SVC 139).

Device control facility reason codes

Reason Code		Description
18000	(X'4650')	Unsupported device.
18001	(X'4651')	An SJF error occurred.
18002	(X'4652')	A construct access services error occurred.
18003	(X'4653')	Invalid CASD parameter list.
18004	(X'4654')	An error recovery and recording error occurred.
18005	(X'4655')	The storage class was not found.
18006	(X'4656')	Invalid SSSCB offset.
18007	(X'4657')	The UCB was not found.
18008	(X'4658')	Invalid DCFDS parameter list.
18009	(X'4659')	No more subsystem to process.
18010	(X'465A')	An AOMSERV error occurred.
18011	(X'465B')	A 3990 storage subsystem could not be found.
18012	(X'465C')	UCB address is 0.
18013	(X'465D')	CMM error occurred.
18020	(X'4664')	No device counts were returned by AOMSERV.
18021	(X'4665')	No statistics were returned by AOMSERV.
18022	(X'4666')	No SSDP pointer was returned by AOMSERV.
18023	(X'4667')	The TIME macro failed.
18024	(X'4668')	Invalid day from TIME macro.
18025	(X'4669')	Invalid control block ID; expected IGDDCFCS.
18026	(X'466A')	Invalid control block ID; expected IGDDCFCD.
18027	(X'466B')	Unexpected return code.
18028	(X'466C')	Unable to stack recovery routine.
18029	(X'466D')	Abend occurred: no token returned.
18030	(X'466E')	Abend occurred: token returned.
18031	(X'466F')	SMS address space not available.
18032	(X'4670')	Create recovery environment failed.
18033	(X'4671')	Add recovery routine failed.
18034	(X'4672')	Zero divisor detected.
18035	(X'4673')	Device table index is 0.
18036	(X'4674')	Invalid search UCB (pointer is zero).
18037	(X'4675')	SSID is forced offline.

Reason Code		Description
18038	(X'4676')	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	(X'4677')	Invalid input to IGDDCFCA.
18050	(X'4682')	Invalid DD Index.
18051	(X'4683')	Invalid UCB ID.
18052	(X'4684')	No storage class for DD.

Operator commands reason codes

Reason Code		Description
19000	(X'4A38')	Do not issue a diagnostic message.
19001	(X'4A39')	Required keyword not found.
19002	(X'4A3A')	Required parameter not found.
19003	(X'4A3B')	Invalid keyword.
19004	(X'4A3C')	Command verb is not known.
19005	(X'4A3D')	SMS is not active.
19006	(X'4A3E')	Caller has insufficient authority.
19007	(X'4A3F')	The command was abnormally terminated.
19009	(X'4A41')	Unable to add recovery routine.
19010	(X'4A42')	Caller did not provide target device number.
19011	(X'4A43')	Caller's requested range exceeds maximum.
19012	(X'4A44')	Unable to obtain storage.
19013	(X'4A45')	MVS message services failed.
19014	(X'4A46')	Storage group not defined.
19015	(X'4A47')	Volume not defined.
19016	(X'4A48')	Invalid keyword value.
19017	(X'4A49')	Invalid delimiter.
19018	(X'4A4A')	Invalid syntax.
19019	(X'4A4B')	Invalid blank.
19020	(X'4A4C')	Null configuration.
19021	(X'4A4D')	No systems.
19022	(X'4A4E')	COMMDS experienced an error.
19023	(X'4A4F')	Function code is not in library drive.
19024	(X'4A50')	Construction type is not in storage group, volume library, or drive for VARY command.
19025	(X'4A51')	Unsuccessful retrieval of the base configuration.
19026	(X'4A52')	The requested OAM display is unknown.
19027	(X'4A53')	OAM execution failed.
19028	(X'4A54')	OAM execution failed due to bad parameter to OAM.
19029	(X'4A55')	Stop building the display.
19030	(X'4A56')	D SMS OAM not allowed.

Reason Code		Description
19031	(X'4A57')	The conversion of console name cannot be done, due to the error of CONVCON macro.
19032	(X'4A58')	UCBLOOK macro failed.
19033	(X'4A59')	Too many operands specified for keyword.
19034	(X'4A5A')	Display request not recognized for RLS related information.
19035	(X'4A5B')	Invocation of ?IGWSSCMD failed.

Error recovery reason codes

Reason Code		Description
20000	(X'4E20')	Invalid subcomponent ID; no match was found during an IGDERRC3 SCID table search.
20001	(X'4E21')	No ERRCA pointer was passed when called by one of the error recovery modules (FRR/ESTAE—that is, PTR_RCA is 0).
20002	(X'4E22')	An SDWA pointer was not passed by the calling module when an SDWA pointer was required (that is, PTR_SDWA is 0).
20003	(X'4E23')	The RECORD request was not completed by the IGDERRC3 SDWA record module.
20004	(X'4E24')	The caller of RECORD service is attempting to record for the second or subsequent time.
20005	(X'4E25')	IGDERRCD (or other error recovery service) is already in process for this ERRCA, and an infinite loop was avoided by simply returning.
20006	(X'4E26')	SYMREC services returned a nonzero return code; all or part of the symptom record may not have been recorded in logrec data set; 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	(X'4E27')	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	(X'4E28')	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	(X'4E29')	A recording services module is trying to access the SSIVT, but cannot do so because the SSIBSUSE pointer from the SSIB is null.
20010	(X'4E2A')	An active recovery environment already exists.
20011	(X'4E2B')	ESTAE create failed.
20012	(X'4E2C')	ESTAE cancel failed.
20013	(X'4E2D')	The recovery environment is not active.
20014	(X'4E2E')	The token of the recovery environment to be canceled does not match the token of the existing recovery environment.
20015	(X'4E2F')	The local recovery routine stack is full.
20016	(X'4E30')	The local recovery routine stack is empty.
20017	(X'4E31')	The local recovery routine to be deleted cannot be found in the local recovery routine stack.
20018	(X'4E32')	The logic trace table header does not exist.
20019	(X'4E33')	A valid ERRCA does not exist.

Reason Code		Description
20020	(X'4E34')	The abending error has not been recorded yet.
20021	(X'4E35')	The environment is not valid for a dump to be taken.
20022	(X'4E36')	Unable to obtain a lock.
20023	(X'4E37')	Unable to release a lock.
20024	(X'4E38')	SVC SDUMP failed.
20025	(X'4E39')	Branch SDUMP failed.
20026	(X'4E3A')	Trace type not N or E.
20027	(X'4E3B')	Trace data contains at least one zero pointer or length value.

Tape volume record update reason codes

Reason Code	Description
21000 (X'5208')	The deletion of the error recovery routine for the SMS tape volume record update failed.
21001 (X'5209')	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 (X'520A')	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 (X'520B')	The tape volume record update failed due to an error from CBRXVOL.
21004 (X'520C')	The retrieval of the tape volume record failed due to an error from CBRXVOL.
21005 (X'520D')	The retrieval of a storage group definition failed due to an error from IGDCNS.
21006 (X'520E')	The library name for the tape volume does not exist in any of the storage group definitions.
21007 (X'520F')	The addition of the error recovery routine for the SMS tape volume record update failed.
21008 (X'5210')	The USE attribute could not be changed in the tape volume record due to an error from CBRXLCS.
21009 (X'5211')	The library name for the tape volume could not be retrieved due to an error from CBRXLCS.
21010 (X'5212')	The retrieval of the storage group names failed due to an error from SJF.
21011 (X'5213')	Storage group names were not retrieved from SJF.
21012 (X'5214')	The request to retrieve and update the tape volume record failed because the data required to retrieve the tape volume constructs was incomplete.
21013 (X'5215')	The retrieval of the SMS construct names failed due to an error from SJF.
21014 (X'5216')	The caller of SMS tape volume record update requested an invalid function.
21999 (X'55EF')	An abend occurred during the execution of SMS tape volume record update.

Device pool SSI codes

Reason Code	Description
220009 (X'035B69')	Bad SJF reason code
220019 (X'035B73')	Device Pools not found.

SMS data set stacking exit codes

Reason Code	Description
23000 (X'059D8')	Unable to establish recovery environment for Data Set Stacking Service.
23001 (X'059D9')	Data Set Stacking Service internal error.
23002 (X'059DA')	Both SMS and managed mountable flags are on.
23003 (X'059DB')	Neither SJF token nor SMS construct information provided for data set being stacked on.
23004 (X'059DC')	SJF retrieval request failed.
23005 (X'059DD')	SJF update request failed.
23006 (X'059DE')	Unexpected return code from invoked routine.
23007 (X'059DF')	Delete recovery routine failed.
23008 (X'059E0')	Construct access services failure.
23009 (X'059E1')	Message services failure.
23010 (X'059E2')	SJF TERMINATE request failed.
23011 (X'059E3')	Neither SJF token nor data set name provided for primary data set.
23012 (X'059E4')	Construct not found.
23013 (X'059E5')	The ACS routine failed the allocation.
23014 (X'059E6')	SSSA5SGN is 0 from ACS routine.
23015 (X'059E7')	Internal error from IDAX.
23016 (X'059E8')	Unexpected error from LCS.
23100 (X'05A3C')	The ACS routine assigned a storage class to a data set being stacked on a non-SMS data set.
23101 (X'05A3D')	No storage groups of compatible types assigned to a data set which is part of a data set collection.
23102 (X'05A3E')	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 (X'05DBF')	Data Set Stacking Service failed; error undetermined.

Chapter 18. 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
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 (by 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 *z/OS MVS Programming: Assembler Services Reference IAR-XCT*, 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 z/OS message manuals) 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 z/OS message manuals. 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
U0001 -- (no message) Error during CCHHR <--> TTR conversion (see note)
```

Note: A program may encounter user ABEND U0001 with or without the DBABE option in effect. If user ABEND U0001 is encountered, it may indicate that the file mark at the end of a PDS member is missing. In some cases, the end-of-file mark can be restored with a DFSMSdss DUMP/RESTORE (or equivalent) operation on the data set. The reason code associated with ABEND U0001 is for use by IBM support in determining the exact nature of the error.

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 19. 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 logrec data set

Information written by recovery routines to the 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 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

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 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

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 20. 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 573
- "Catalog management diagnostic aids" on page 576
- "VSAM OPEN/CLOSE/End-of-Volume (O/C/EOV) diagnostic aids" on page 578
- "VSAM Record-Level Sharing (RLS) diagnostic aids" on page 582
- "VSAM record management (R/M) diagnostic aids" on page 593
- "VSAM record management return and reason codes" on page 611

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 574 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 that calls IO01 and then returns back to RI01, the trace table appears as follows:

```
INTER ... RI01 TP01 IO01 TP01 RI01
```

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" on page 574.

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 574.

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.

<table>
<tr><td>PARM</td><td>[TEST({[TRACE]
[AREAS (<i>areaid</i> [<i>areaid</i>...])]
[FULL ((<i>dumpid</i> [<i>begin</i> [<i>count</i>]]) [(<i>dumpid</i>...)....])]|</td></tr>
</table>

	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 45 on page 576 shows a portion of the output from the command:

```
PARM TEST ( FULL (LCTP,2,1) )
```

and a portion of the dump produced.

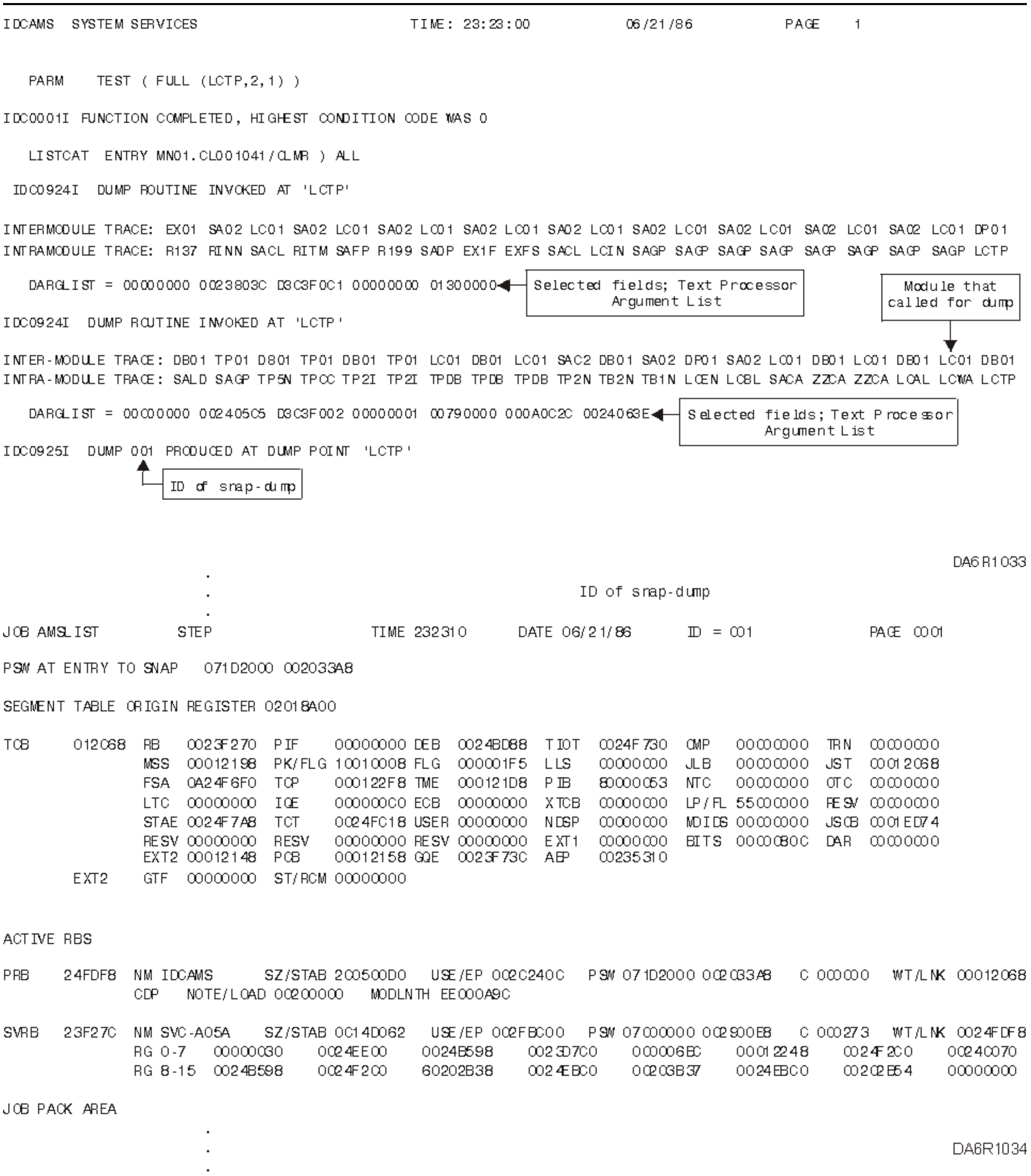
The trace tables and the selected area, DARGLIST, are printed each time the dump point LCTP is encountered. A full region dump is produced the second time that dump point LCTP is encountered.

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

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 = 00000000 002405C5 D3C3F002 00000001 00790000 000A0C2C 0024063E

IDC0925I  DUMP 001 PRODUCED AT DUMP POINT 'LCTP'
```

Selected fields; Text Processor Argument List
Module that called for dump
Selected fields; Text Procesor Argument List
ID of snap-dump
DA6R1033

```
                          .
                          .
                          .

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   00012C68
                 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  AEP   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    000006EC    00012248    0024F2C0    00240070
                RG 8-15   0024B598    0024F200    60202B38    0024EBC0    00203B37    0024EBC0    00202E54    00000000

JOB PACK AREA
                          .
                          .
                          .
```

Figure 45. Example of TEST Option Output

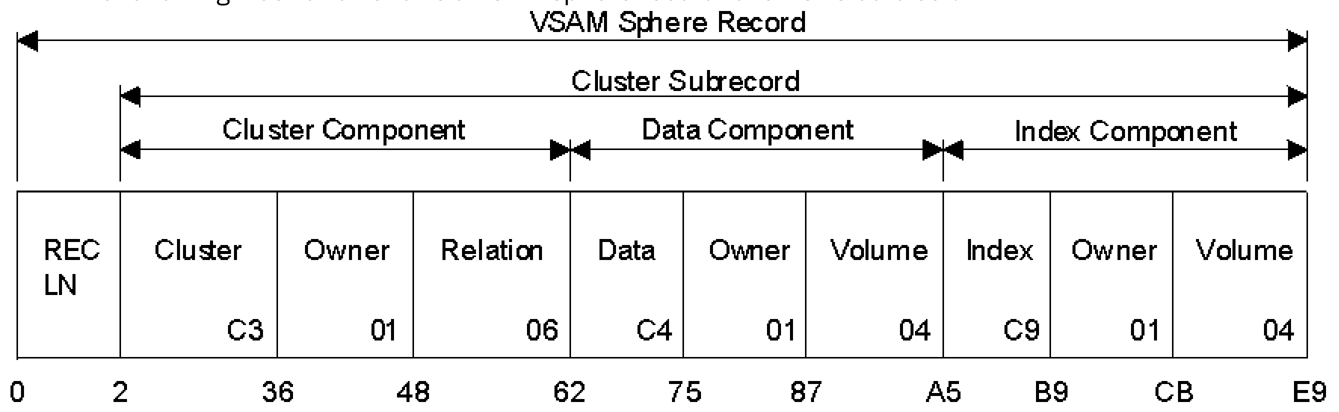
Catalog management diagnostic aids

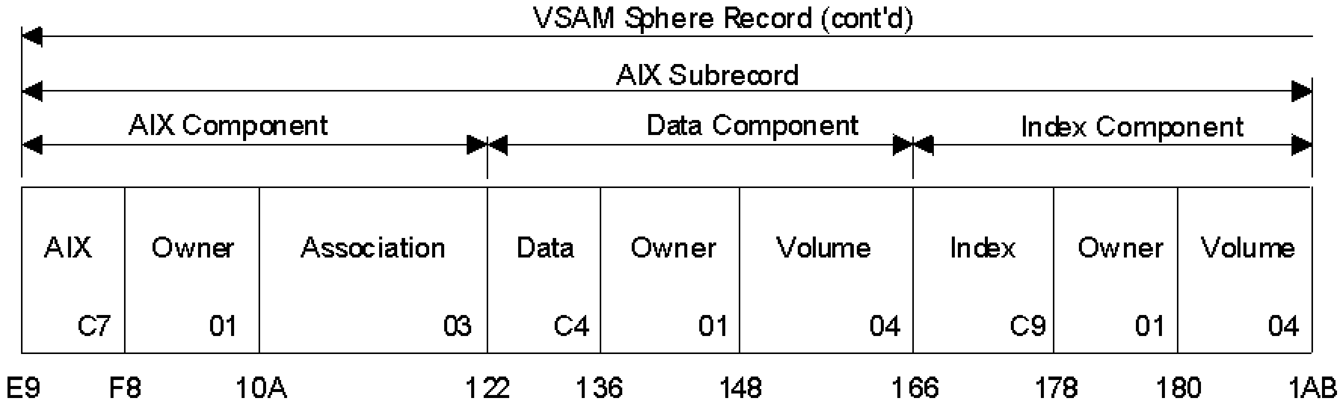
The DIAGNOSE command output is designed to help you diagnose catalog management problems.

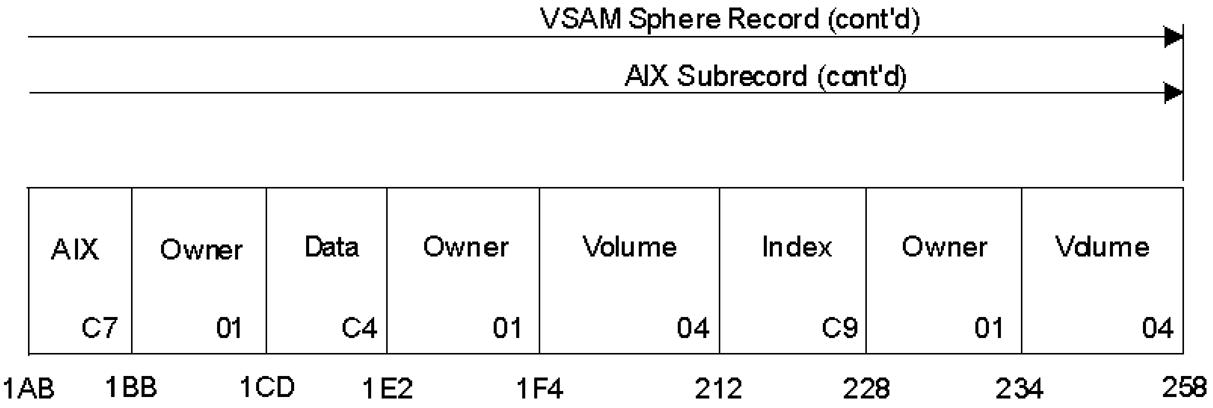
Sample DIAGNOSE output

Illustrations of catalog records follow, in the format in which they would be displayed after execution of the DIAGNOSE command.

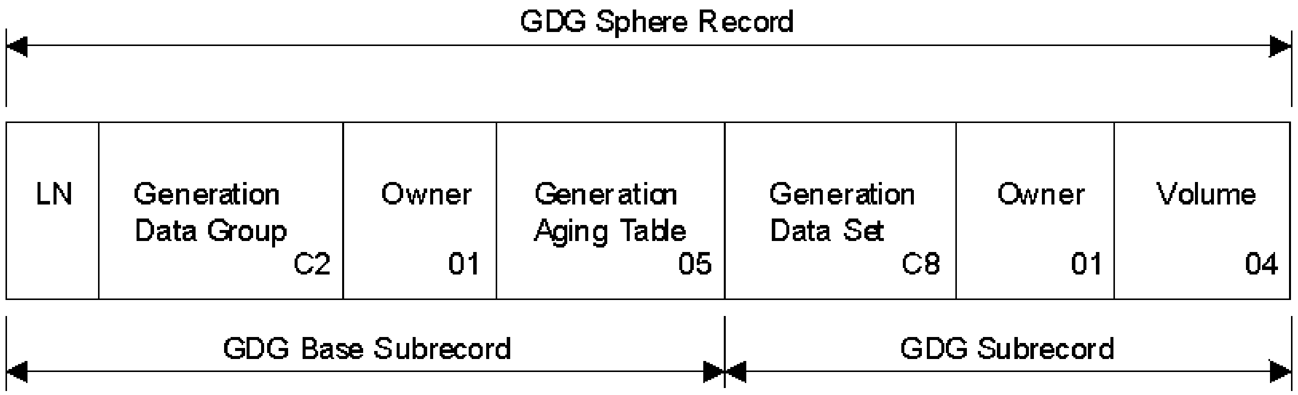
The following illustration shows a VSAM sphere record for a KSDS data set:

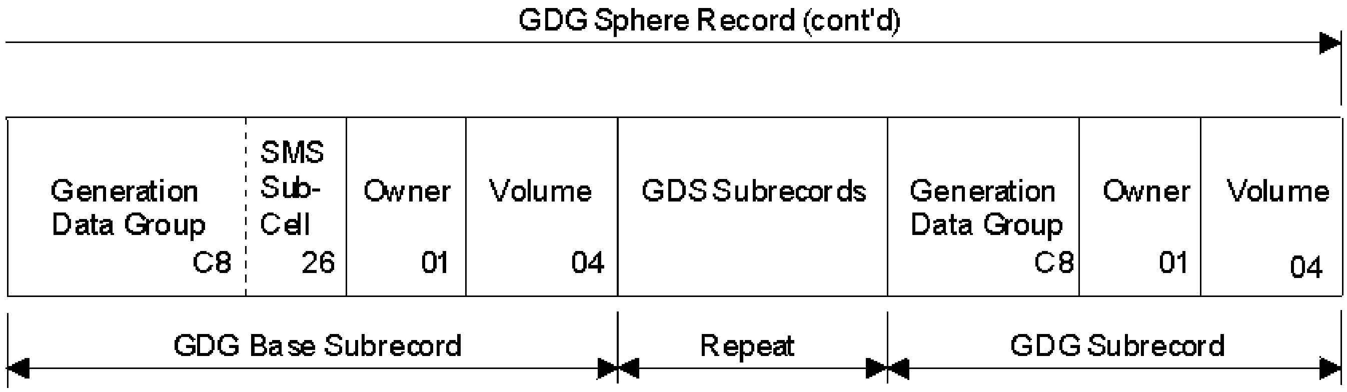




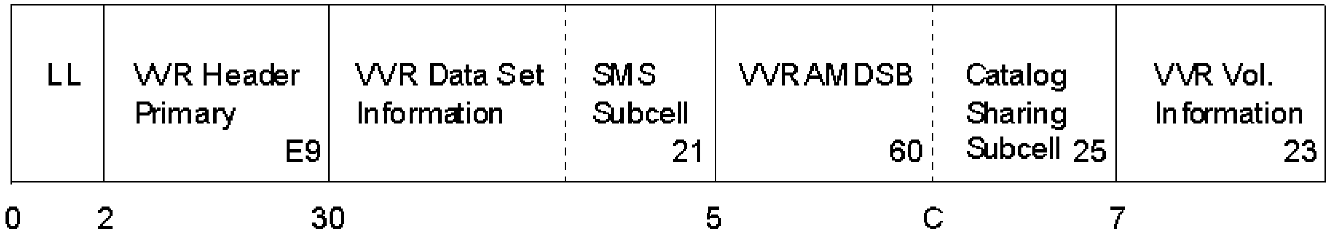


The following illustration shows a GDG sphere record consisting of a base record and 10 subrecords (as indicated in the generation aging table cell):

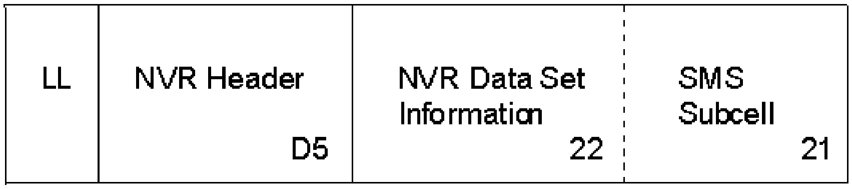




The following illustration shows a VVDS record (primary VVR):



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.

Note: The speed at which trace data is written out to GTF has been increased and this can lead to an increase in the number of lost event records (trace record that can not be written out due to the GTF performance). The number of lost event records will increase, especially when the program is running under the multi-thread environment. Therefore, it is recommended that you adjust the GTF parameters when starting GTF before tracing. Adjustments such as increasing the GTF buffer size, increasing amount of virtual storage that GTF will use to collect trace data, and using VSAM linear data set as the target data set will help to reduce the number of lost event records. Refer to *z/OS MVS Diagnosis: Tools and Service Aids* for information on how to customize GTF parameters.

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 124 on page 579 gives detailed information related to the X'F61' record.

Table 124. 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

Table 124. X'F61' Record Information (continued)

Offset	Name	Length	Description
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
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 (specified by 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 and EOV first-time failure data capture

VSAM provides first-time failure data capture (FFDC) by taking an SVC dump in VSAM OPEN, CLOSE and EOV when it encounters logic errors. No action is necessary to authorize these dumps. VSAM does not generate an ABEND and continues processing.

If you need further assistance after a dump has been taken, contact IBM service.

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” on page 582 and “VSAM record management return and reason codes” on page 611 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-->

Note:

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. Optionally (this is needed only when using CTRACE WRITER) create a member in SYS1.PROCLIB to allocate a CTRACE WRITER data sets. 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
   ```

4. When specifying the REPLY command, you should ensure that JOBNAME and ASID parameters are specified as directed by VSAM RLS support. Not specifying the JOBNAME and ASID parameters will result in CTRACE buffers being allocated for all address spaces, which may affect the operation of those address spaces, as this storage will not be freed up until end-of-job and end-of-task (EOJ/EOT), with the CTRACE off.
5. Do not issue two SYSSMS CTRACE start commands without an intermediate SYSSMS CTRACE stop command. Two consecutive SYSSMS CTRACE start commands will result in two sets of CTRACE buffers being allocated for the jobs and address spaces specified on the JOBNAME and ASID parameters, which may affect the operation of those jobs and address spaces.

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 125 on page 583.

Table 125. 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.

Table 125. Valid Trace Facility Options (continued)

Trace Event	Description
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
CERS
RDS
SCM
SHC
SHM
SLCH
SMLS
SMPM
SQM
VOC
VQUI
VRM
VSS

Note:

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 these options, 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
```

VSAM RLS index trap

For VSAM RLS, there is an index trap that checks each index record before writing it. The trap detects the following index corruptions:

- High-used greater than high-allocated
- Duplicate or invalid index pointer
- Out-of-sequence index record
- Invalid section entry
- Invalid key length.

If the index record is detected to be corrupted, an ABEND of 0F4 with a system dump is generated but the SMSVSAM Server is not recycled. Contact IBM service if you see any of the following:

- A dump with the following title:

```
DUMP TITLE=COMPID=DF122,CSECT=IDAVRBF4+xxxx,DATE=
   ID=xxxxxxx ,ABND=0F4,RC=00001008,RSN=61609Dxx
```

- The RPL feedback word xx08009D.

Submit the following information to IBM service:

1. The dump
2. The EXAMINE and LISTCAT outputs taken at that time
3. Printouts of the index and data components done at that time.

More documentation may be requested, depending on what is known about the problem at hand.

The next requests in this system that access this (base) data component will fail with RPL feedback word = xx08009E, so further damage to the data set is avoided. However, access from other systems are still allowed until this trap springs there.

This index trap is normally inactive. The following console command from any one system will dynamically activate the index trap throughout the sysplex (including new systems which come up later, and systems that go down after the command has been issued and come back up):

```
V SMS,MONDS(IGWVSAM.INDEX.TRAP),ON
```

You can issue the following command to deactivate the index trap:

```
V SMS,MONDS(IGWVSAM.INDEX.TRAP),OFF
```

To query the status of the index trap, issue the following command:

```
 D SMS,MONDS(SPECIAL.FUNCTION.STATUS
```

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 IGWFPMAN 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 V SMS,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 terminate 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 126 on page 587 can be used as a guide to what actions to take for specific initialization problems.

Table 126. 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),CONT
   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 VERBX SMSXDATA 'F(AS) JOB(SMSVSAM)' 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 latch contention

If you suspect that VSAM RLS latch contention is causing a hang or deadlock, use the following operator command:

```
DISPLAY SMS,SMSVSAM,DIAG(CONTENTION)
```

This command produces a console message that either:

1. Displays all latch contention active on the system (if any), or
2. Indicates that there is no latch contention on this system.

For more information on the DISPLAY SMS,SMSVSAM,DIAG(CONTENTION) command, refer to *z/OS MVS System Commands*.

Examples

- This example shows latch contention.

```
09.55.29 SYSTEM1           IGW343I VSAM RLS DIAG STATUS (V.01)
|---RESOURCE----| |------ WAITER ------| |--HOLDER---| ELAPSED
TYPE        ID    JOB NAME ASID   TASK   ASID   TASK     TIME
-------- -------- -------- ---- -------- ---- -------- --------
LATCH    7F158C70 SMSVSAM  003A 008DA250 003A 008D7218 00:00:06
    DESCRIPTION: IGWLYSPH - SHM OBJECT POOL
LATCH    7F151E78 SMSVSAM  003A 008D7218 003A 008DC1C8 00:00:21
    DESCRIPTION: IGWLYDTS - SHM OBJECT POOL
LATCH    7BAD43B8 SMSVSAM  003A 008DC1C8 002D 007F3000 00:19:09
LATCH    7BAD43B8 SMSVSAM  003A 008D5A48 002D 007F3000 00:22:09
LATCH    7BAD43B8 SMSVSAM  003A 008D6938 002D 007F3000 00:33:23
LATCH    07F1B1D0 SMSVSAM  003A 008D64F8 003A 008D6CF0 01:47:20
LATCH    07F1D3B8 SMSVSAM  003A 008D6CF0 0000 00000000 11:23:30
```

  In this example, the numbers in the "ELAPSED TIME" column represent the time a latch has been in contention.

  If latch contention is indicated, you can then take appropriate actions to resolve the hang or deadlock. These actions may range from canceling batch jobs or CICS regions (client requests) to terminating an instance of the SMSVSAM server. When deciding what action to take, consider the following:

  - If the holder of a latch that is holding other threads is a cross memory task from a client address space (such as a CICS transaction or batch job), you could then cancel that task or job. However, if the holder is a very important job, and the waiters are not as important, then you might decide to wait for the busy shift to finish and cancel the job during the off-shift.
  - If the holder of a latch is itself waiting for an ENQ that is held in another system, then you should consider issuing the DISPLAY SMS,SMSVSAM,DIAG(CONTENTION) command on the other system to see if there is a problem there.
  - If the holder of a latch is itself waiting for I/O or any other system resource, then you should investigate why that resource is not available or slow to respond.
  - If the holder of a latch is an SMSVSAM (VSAM RLS) thread, then you might need to restart the SMSVSAM address space. Before doing this, however, you should consider the consequences of restarting the server immediately, versus waiting for the off-shift. Even though some work might be stalled, there could be some other more important work that is still progressing.

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 messages DFHFC0164, DFHFC0165, DFHFC0166, and DFHFC0167. 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 582 and “VSAM record management return and reason codes” on page 611.

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 127. 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'61FF0015'	There is a nonRLS quiesce active within this address space. What the situation is across the sysplex is unknown.
8	X'61FF0016'	There is a QUICOPY active for this data set on this system.
8	X'61FF0017'	Thereis a QUIBWO active for this data set on this system.
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.

Table 127. SMSVSAM Return and Reason Codes (continued)

Return Code	Reason Code	Explanation
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.
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, because the SMSVSAM server failed during the copy.
8	X'61FF000C'	The task that 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'60EF0007'	SMSVSAM is not available.

Table 127. SMSVSAM Return and Reason Codes (continued)

Return Code	Reason Code	Explanation
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.
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 Table 128 on page 592 can be returned from the SMPM_CFPurge interface.

Table 128. 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 Table 129 on page 593 can be returned from the SMPM_CFQuery interface.

Table 129. 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'64xxFA13'	Unknown catalog error was encountered. Catalog error will also be returned.
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.

Return and reason codes from IGWMCDATA when adding data set masks

To monitor a data set and collect SMF42 subtype 16 records, you can specify data set masks. A return code of CdataMaskRc_Succ (0) means the mask was accepted. A return code of CdataMaskRc_Warn (8) means that the mask was not successfully processed, and the associated reason code contains qualifying information. The return and reason codes are described in Table 130 on page 593.

Table 130. IGWMCDATA Return and Reason Codes

Return Code	Reason Code	Explanation
8	X'300'	CDATAMASKRSN_NOHLQ. There was no high level qualifier.
8	X'310'	CDATAMASKRSN_INVALIDCHARACTERS. There were invalid characters in the mask. Not a valid data set name.
8	X'350'	CdataMaskRsn_DataAfterWildcard. Data was found after a wildcard (* or **) for a data set name.
8	X'360'	CDATAMASKRSN_INVALIDLENGTH. The length specified for the mask was invalid: either smaller than or equal to 0, or greater than 44, for data set names.

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 dynamic dump facility
- 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 EDT-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 with 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 "Keyword worksheet" on page 4. Append the 4-byte field to the keyword prefix **RC**.

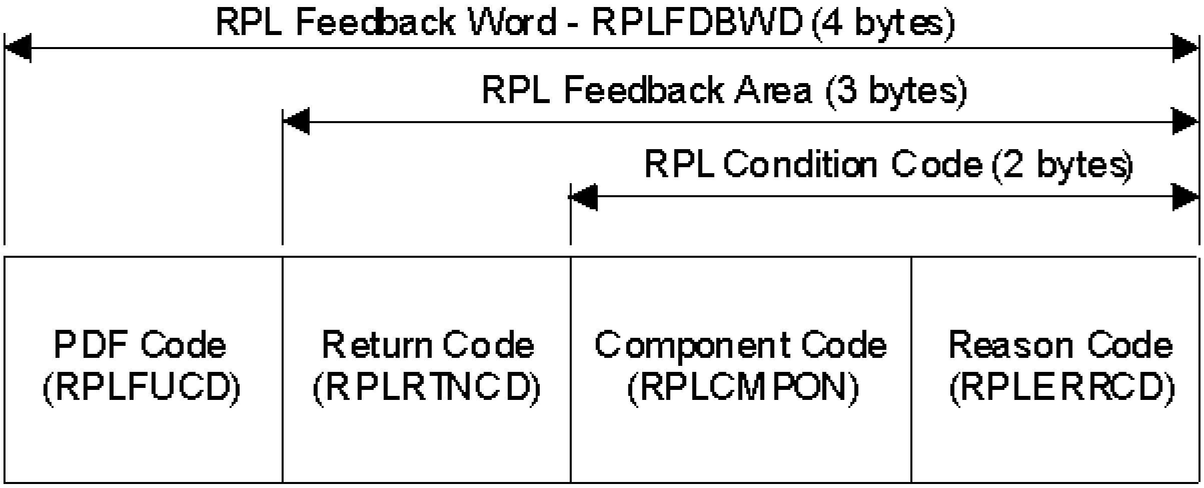
Example: The RPL feedback word is **2D08009C**; specify this modifier keyword as shown:

```
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 VDUMPON command or 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, identifies the reason for an error.


RPL Feedback Word - RPLFDBWD (4 bytes)
RPL Feedback Area (3 bytes)
RPL Condition Code (2 bytes)
PDF Code (RPLFUCD)
Return Code (RPLRTNCD)
Component Code (RPLCMPON)
Reason Code (RPLERRCD)

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 Commands* 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 can 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.

An apparent exclusive control conflict can also occur when the program is inserting a new record or updating an existing record to a compressed VSAM data set with spanned records, even though no other I/O is occurring against this data set. This occurs because of a CI or a CA split. Following the split, positioning is lost and must be reestablished, resulting in an RPL feedback word of x'D4080014'. This situation is similar to an exclusive control conflict, and requires that the VSAM user re-drive the 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 46 on page 596 shows how to do this:

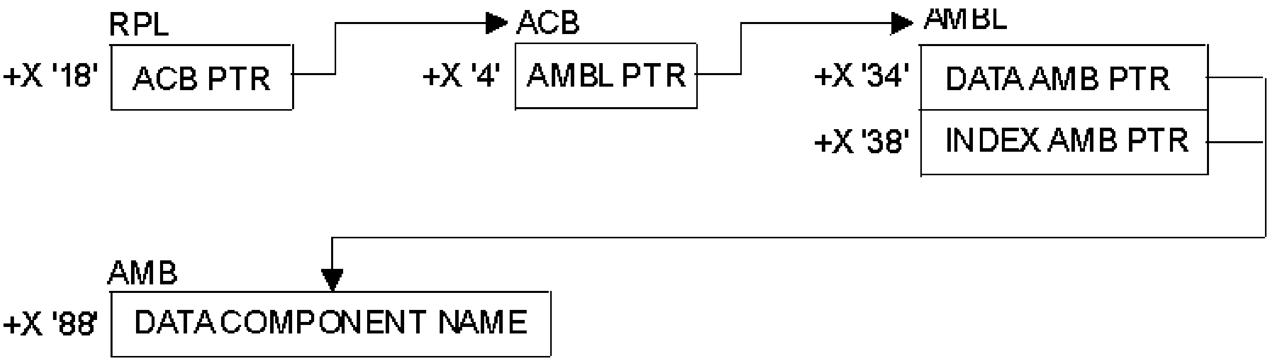


Figure 46. VSAM Record Management—How to Find a Damaged Data Set

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

Table 131 on page 597 lists typical symptoms of a damaged data set when IDCAMS detects the damage.

- Messages

Table 131. 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')
```

VSAM index trap

VSAM has an index trap that checks each index record before writing it. The index trap is on by default.

The index trap can be deactivated by the console command:

```
VARY SMS,MONDS(IGWVSAM.BASE.INDEX.TRAP),OFF
```

To restart it after disabling it:

```
VARY SMS,MONDS(IGWVSAM.BASE.INDEX.TRAP),ON
```

To query status:

```
DISPLAY SMS,MONDS(IGWVSAM.BASE.VSAM.DEBUG.FEATURES)
```

The trap catches the more common index breakages, but not all breakages:

1. High-used greater than high-allocated
2. Invalid index pointer
3. Duplicate index pointer
4. Out-of-sequence index record
5. Invalid section entry
6. Invalid key length
7. Horizontal pointer loop
8. Duplicate key
9. Incorrect segment count

The trap prevents a broken index, but the documentation generated might not be sufficient to determine the source of the problem. If the trap hits:

1. A console message is issued:

   ```
   IDAI1001E ERROR FOUND IN INDEX
   DSN:  SAMPLE.DATASET.INDEX
   REASON:  HIGH-USED GREATER THAN HIGH-ALLOCATED
   ```

   A similar message, IDAI1002E, is issued if an error is found in a catalog data set index.
2. A dump is generated.
3. The current request fails with logical-error RPL feedback 'xx08006D'.
4. No ABEND occurs.

If the data set is being shared properly, the systems programmer should:

1. Open a PMR and submit the following items:
   a. The dump
   b. An EXAMINE ITEST NODTEST, EXAMINE DTEST NOITEST, and LISTCAT of the data set taken after the trap hits
   c. DSS Prints of the index component after the trap hits
2. Rename and save the corrupted data set, in case IBM Service needs additional documentation

Note: IBM Health Checker for z/OS VSAM provides a check, VSAM_INDEX_TRAP, which ensures that the VSAM index trap is enabled and notifies users with an exception message if it is not enabled. When this check is active, it will be run once the IBM Health Checker for z/OS is started and once every 24 hours while IBM Health Checker for z/OS is active.

For more information on IBM Health Checker for z/OS, see *IBM Health Checker for z/OS User's Guide*.

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 Commands* 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.

   Note:

   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.

VSAM data trap

⚠ **CAUTION:** The VSAM data trap described in this topic is intended to be used under the direction of IBM Service. It can cause performance degradation due to the additional validation of data records before they are written to DASD. You should enable the VSAM data trap only after contacting IBM service when a problem is encountered related to a data CI and additional diagnostic data is needed for problem determination. The VSAM data trap should be disabled as soon as the data is collected.

If the data component of a key-sequenced data set (KSDS) is damaged, the problem may show up in several ways, including:

- Records out of sequence
- Records in the freespace area
- Problems with RDF or CIDF fields.

If data corruption occurs, the VSAM data trap can help to:

- Prevent the record from being added incorrectly
- Collect documentation close to the time of the problem. Without the VSAM data trap, diagnostic data is collected too long after the initial error and is usually not sufficient to diagnose the problem.

The VSAM KSDS data trap includes the following features and limitations. It:

- Is only for KSDS
- Does not test spanned records or when there can be only one record in a CI
- Cannot test for sequence errors between control intervals
- Tests key sequence in the CI as it is being written to DASD
- Examines the Control Interval (CI) to see if the keys have correct value as related to the keys for the previous, and next, control interval (CI).

Use the following MVS commands to turn the data trap off or on. By default, the VSAM data trap is off.

- Make VSAM data components eligible for the data trap:

```
V SMS,MONDS(IGWVSAM.BASE.DATA.TRAP),ON[,](data component name1,…,data component name8)
```

  At least one data component name must be provided. The following additional syntax rules apply.

  - The maximum length for the entire data set mask is 44 characters.
  - An asterisk (*) or percent sign (%) can be specified anywhere except in the first position.
  - % represents any single character. For example, %%% represents any 3 characters.
  - * by itself indicates that at least one qualifier occupies that position. * within a qualifier represents zero or more characters.
  - A qualifier can be a single *.
  - ** represents zero or more qualifiers.
  - ** cannot appear with any other characters within a qualifier.
  - Three or more adjacent asterisks (for example, ***) are not allowed within a qualifier.

  The data components are enabled when they are opened. To add more data components to the list, issue the command with new data component names. The data set specified in the command is enabled to use the data trap, and any data set previously enabled for the data trap is enabled again and remains eligible until the data set is closed.

  To make all VSAM data components eligible for the data trap, use an asterisk in place of the data component names, as follows:

  ```
  V SMS,MONDS(IGWVSAM.BASE.DATA.TRAP),ON(*)
  ```

- Turn the data trap off for future use:

  ```
  V SMS,MONDS(IGWVSAM.BASE.DATA.TRAP),OFF
  ```

  To disable the data trap for any existing data set being trapped, the data set must be closed.

- To determine if the data trap is active for the system:

  ```
  D SMS,MONDS(IGWVSAM.BASE.VSAM.DEBUG.FEATURES)
  ```

  This displays the status of VSAM diagnostic features. If the data trap is enabled, the command displays the data component name(s) found in the current data trap list created when you last enabled the data trap with the VARY SMS command. Any data set previously enabled for the data trap remains eligible until the data set is closed.

Examples

You issue the following commands:

1.
   ```
   V SMS,MONDS(IGWVSAM.BASE.DATA.TRAP),ON(dsn1)
   ```

   dsn1 is eligible to be checked by the trap.

2.
   ```
   OPEN dsn1
   ```

   dsn1 is now checked for data corruption by the data trap.

3.
   ```
   V SMS,MONDS(IGWVSAM.BASE.DATA.TRAP),ON(dsn2)
   ```

   dsn2 is eligible to be checked when it is opened. The list will only show *dsn2*; however, *dsn1* is still checked by the trap until it is closed.

Here is a slightly more complex example:

1.
   ```
   V SMS,MONDS(IGWVSAM.BASE.DATA.TRAP),ON(dsn1)
   ```

   dsn1 is eligible to be checked by the trap.

2.
   ```
   OPEN dsn1
   ```

   dsn1 is now checked for data corruption by the data trap.

3.
   ```
   V SMS,MONDS(IGWVSAM.BASE.DATA.TRAP),ON(dsn2)
   ```

   dsn2 is eligible for the trap. The list will only show *dsn2*; however, *dsn1* is still being checked by the trap because it has not been closed.

4.
   ```
   V SMS,MONDS(IGWVSAM.BASE.DATA.TRAP),OFF
   ```

   dsn2 is no longer eligible for the data trap, and the data trap is disabled for *dsn1*.

5.
   ```
   V SMS,MONDS(IGWVSAM.BASE.DATA.TRAP),ON(dsn2,dsn3)
   ```

   dsn2 and *dsn3* are eligible for the data trap. The list will show `dsn2,dsn3`; however, they are not checked by the trap until they have been opened. Because the trap is turned back on and *dsn1* has not been closed, *dsn1* is again checked by the data trap.

6.
   ```
   OPEN dsn3
   ```

   dsn3 is now checked for data corruption by the data trap, as is *dsn1*. *dsn2* is still not checked because it has not been opened.

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. If the REPRO fails, 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 Media Manager. See *z/OS MVS Diagnosis: Tools and Service Aids* for details about using CCWTRACE.

VSAM record management dynamic dump

VSAM Record Management provides two kinds of dynamic dumps:

Automatically generated dumps
User directed dumps

Automatically generated dumps

VSAM Record Management will automatically generate a dump when certain unexpected errors are detected. No action is necessary by the user to authorize these dumps.

When a dump is generated, z/OS will issue a message to indicate in which dump data set the dump was captured, as shown in the following example:

```
*IEA911E COMPLETE DUMP ON SYS1.DUMP00
*DUMPID=001 REQUESTED BY JOB (CATALOG )
*FOR ASIDS(0022,002B)
*INCIDENT TOKEN: LOCAL    SYSTEM1  05/18/2006 21:56:50
```

User directed dumps

In addition to the automatic dumps previously described, you can also optionally specify whether VSAM Record Management dynamic dumping is to occur for specific events identified by a combination of a return code, error (reason) code, problem determination function (PDF or FUNC) code, and component code returned in the request parameter list (RPL) feedback area. (See "Return codes from the record-management (request) macros" on page 611.)

You can request a user directed dump (or change the parameters for an existing dump request) with the MODIFY CATALOG,VDUMPON command. You can turn off the dump request with the MODIFY CATALOG,VDUMPOFF command. See *z/OS MVS System Commands* for a complete description of the MODIFY CATALOG,VDUMPON and MODIFY CATALOG,VDUMPOFF commands.

For example:

```
MODIFY CATALOG,VDUMPON(**,8,**,20)
```

requests a dump for any return code 8, error code 20 (exclusive control conflict), with any PDF code and any component code.

z/OS acknowledges the command withe following messages:

```
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC359I CATALOG REPORT VDUMP OUTPUT 503
*CAS*************************************************************
*    STATUS  FUNC CODE  RETURN CODE  COMPONENT  ERROR CODE      *
*       ON      ***         008         ***        020          *
*CAS*************************************************************
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

If a dump is later generated, z/OS will issue a message similar to that generated for an automatic dump.

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 579 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

There are two methods for starting the R/M trace function:

- Using PARMLIB member IDAVDT*xx* and started task IDAVDT.
- Using AMP=('TRACE=(subparameters)') on the DD card of the JCL. Note that this method has several limitations: It does not support dynamically allocated data sets, cannot be enabled dynamically and requires that you take down the application to insert trace parameters and open the data set before the trace can be enabled.

With either method, begin as follows:

1. Review the GTF reserved common storage buffer size to ensure that the buffer is large enough to hold the resulting trace records. The default size is 40K. IBM recommends that you increase it to at least 1024K. Refer to *z/OS MVS System Commands* for information on how to define GTF reserved common storage buffer.
2. 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.

Then complete the tasks for the method you choose, as described in "Using PARMLIB member IDAVDTxx and IDAVDT" on page 603 and "Using AMP=('TRACE=(subparameters)') in the DD card" on page 605.

When tracing catalogs opened in the Catalog address space, you must issue this command to allow tracing to start:

```
F CATALOG,CLOSE(catalogname)
```

Using PARMLIB member IDAVDTxx and IDAVDT

After completing the steps in "Starting the record management trace function" on page 603, start the R/M trace function as follows:

1. Define up to 8 trace entries in PARMLIB member IDAVDT*xx*. This is described in "Syntax of Trace entries in IDAVDTxx" on page 603.
2. Start the IDAVDT started task with the START IDAVDT command. Message IDAT0001I is issued to confirm that IDAVDT has started successfully.
3. Use the MODIFY command to request specific trace functions, as described in "Syntax of the MODIFY command with IDAVDT" on page 604.

Stop the IDAVDT started task with the STOP IDAVDT command. Before terminating, IDAVDT will invalidate all trace entries in the dynamic trace save area to prevent VSAM OPEN from comparing data sets from future OPENs with the trace entries. If you would like the status of trace entries to remain unchanged when IDAVDT terminates, you must use the KEEP option of the READIN function on the MODIFY command.

Syntax of Trace entries in IDAVDTxx

You define trace entries in PARMLIB member IDAVDT*xx* using the following syntax:

```
VTRACE    DSNAME(data-set-name)
          JOBNAME(job-name)
          HOOK(hook-id [,hook-id])
          PARM1(trace-parm-1)
          [PARM2(trace-parm-2)]
          [ECODE(ANY|rsncode)]
          [KEYVALUE(target-key-value) | KEYVALUE(x'target_key_value_hex')]
          END
```

VTRACE
Specifies the current entry is to define a VSAM record management trace entry.

DSNAME(*data-set-name*)
Specifies the base cluster name of the targeted data set. You can use the asterisk (*) as a pattern matching (wildcard) character. An asterisk matches zero or more characters when used with other characters, or one or more characters if used alone. For example, SMITH.TEST*.DATA would match SMITH.TEST.DATA as well as SMITH.TESTALL.DATA. SMITH.*.DATA would match SMITH.TESTALL.DATA, but not SMITH.DATA. To match zero or more characters for a data set name qualifier, specify two asterisks. For example. SMITH.**.DATA would match SMITH.DATA.

JOBNAME(*job-name*)
Specifies the job where the targeted data set was opened or will be opened.

HOOK(*hook-id* [,*hook-id*])
Specifies the list of VSAM predefined hook points which the trace should execute during VSAM record management processing.

PARM1(*trace-parm1*)
Specifies which control blocks VSAM record management trace should capture.

PARM2(*trace-parm2*)
Specifies which control blocks VSAM record management trace should capture when tracing the AIX®.

ECODE(ANY|*rsncode*)
Specifies tracing should occur only if an error code is being returned to the caller.

ANY
Tracing is performed for any nonzero return code.

rsncode
Tracing is preformed only if the RPLFDBK code matches *rsncode*.

KEYVALUE(*target-key-value* | x'*target_key_value_hex*')
Specifies tracing should occur only if the current processing key matched *target-key-value* or hexadecimal value x'*target_key_value_hex*'.

For a description of the subparameters, such as the values for HOOK, PARM1 and PARM2, see "Subparameters for trace" on page 605.

Syntax of the MODIFY command with IDAVDT

You request specific trace functions with the MODIFY (F) command. The syntax is:

```
F IDAVDT,function
```

The values for *function* are:

READIN[=*xx*][,KEEP]
Parse the trace entries in IDAVDT00 (the default) or IDAVDT*xx* and insert them into the dynamic trace save area located in common storage. You can optionally add KEEP, to keep trace entries valid after IDAVDT terminates.

ENABLE=*n*[,ASID=*asid*]
Enable a trace entry. *n* is the trace entry number in the dynamic trace save area. You can optionally specify the ASID of the job where the target data set is opened, to optimize the search time.

DISABLE=*n*[,ASID=*asid*]
Disable a trace entry. *n* is the trace entry number in the dynamic trace save area. You can optionally specify the ASID of the job where the target data set is opened, to optimize the search time.

DISPLAY=*n*
Display a trace entry, including the status for the trace entry (enabled or disabled). *n* is the trace entry number in the dynamic trace save area.

VALIDATE=*n*|ALL
Validate a trace entry, or all trace entries, that are stored in the dynamic trace area.

INVALIDATE=*n*|ALL
Invalidate a specific trace entry, or all trace entries, that are stored in the dynamic trace area.

Use this if you no longer want future OPENs to be impacted by the trace entry. Otherwise, OPEN will continue to look for a name match for every data set that is opened even if the IDAVDT started task is not active. Once a trace entry has been invalidated, VSAM OPEN will no longer compare that entry with the data sets being opened and all future ENABLE and DISABLE commands against the invalidated entry will fail. Use VALIDATE to validate trace entries that were invalidated by INVALIDATE.

INVALIDATE will not affect data sets that are already opened with VSAM Record Management Trace processing.

Using AMP=('TRACE=(subparameters)') in the DD card

After completing the steps in "Starting the record management trace function" on page 603, start the R/M trace function as follows:

1. Place 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*.

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 that 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.

For a description of the subparameters, such as the values for HOOK, PARM1 and PARM2, see "Subparameters for trace" on page 605.

Subparameters for trace

This topic describes the subparameters for trace.

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.

Table 132 on page 606 lists the predefined trace point IDs, their associated modules, and their functions.

Table 132. 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	IDA019RZ	All I/O occurring during a CI split.
0009	IDA019RF	Start of a CA split.
0010	IDA019RF	After completion of a CA split.
0011	IDA019RZ	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.
	IDA019R2	Non-shared resources—after I/O, no errors.
0016	IDA019RW, IDA019SY	Shared resources—after I/O, error occurred.
	IDA019R2	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.
0024	IDAM19R3	Prior to SVC 121 for reads of CIS.
0025		Reserved.
0026	IDA019SC	Start of a CI reclaim.
0027	IDA019SC	After completion of a CI reclaim.
0028–0255		Reserved.

Note:

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 Using Data Sets* 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

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 133. PARM1 Subparameter Bits (Byte 0)

Byte 0	Notes	Description
1...		Reserved.
.1..	1	ACB–access method control block.
..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 134. PARM1 Subparameter Bits (Byte 1)

Byte 1	Notes	Description
1...	2	BSPH–buffer subpool header.
.1..	2	BUFC–buffer control block.
..1.	1	CMB–cluster management block.
...1		CMWA-Compression work 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 135. PARM1 Subparameter Bits (Byte 2)

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...		IXT-Index Trap Save area.
.... .1..		IXSPL–index search parameter list.
.... ..1.	1	LPMB–logical-to-physical mapping block.
....1	2	PLH–placeholder.

Table 136. PARM1 Subparameter Bits (Byte 3)

Byte 3	Notes	Description
1...		RPL–request parameter list.

Table 136. PARM1 Subparameter Bits (Byte 3) (continued)

Byte 3	Notes	Description
.x..		Reserved.
..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 137. PARM1 Subparameter Bits (Byte 4)

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.
... ..x.		Reserved.
... ...1		No PLH suppression.

Table 138. PARM1 Subparameter Bits (Byte 5)

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

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) that is shown in Table 139 on page 610.

Table 139. PARM2 Subparameter Bits (Byte 5)

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.
   - The compression work area, CMWA, will only be captured while tracing at Hook 20 through 23 during compression processing.
5. The VSAM Index Trap save area, IXT, will only be captured while tracing at Hook 8 (All I/O occurring during a CI split) and Hook 11 (All I/O occurring during a CA split).

Ending the record management trace function

Tracing of a data set is terminated for a job when the data set is closed. To terminate tracing without closing the data set, you can stop GTF. You can then resume tracing by starting GTF again with the USR=FF5 parameter.

If you are using started task IDAVDT, you can end tracing for a specific data set, while continuing tracing for other data sets, with this command:

```
F IDAVDT,DISABLE=n
```

where *n* is the trace entry for the data set. You can stop the IDAVDT started task with this command: `STOP IDAVDT`.

Printing the record management trace output

VSAM trace records are formatted and printed with IPCS. "Printing GTF records" on page 581 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 trace data suppression

Meaningless data and duplicated data in the index or data buffer will not be captured into the trace dataset. This reduces the size of the trace data set and number of GTF call needed.Suppressed data is replaced with a sentence indicating how many bytes are being skipped and the reason. For example:

```
** THE FOLLOWING 1024 BYTES OF DATA ARE SAME AS THE PREVIOUS LINE **
```

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 140 on page 611.

Table 140. 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 141 on page 611 .

Table 141. 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')	• The SMSVSAM server is not active. • The OPEN connection is invalid. • DFSMStvs is not active.

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 142 on page 612 describes the reason codes that may be found in the RPL feedback area when register 15 is 0.

Table 142. 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.
28(X'1C')	A CI split for the CI was interrupted. The CI was read as nonupdate with address access. This warning condition indicates that duplicate data records may exist. The RBA of the interrupted split can be acquired from RPLDDDD or, if the data set is an EA data set, then from the lower six bytes of the field RPLRBAR.
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, because 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.
48(X'30')	An error occurred during CA Reclaim. The erase was successful.
56 (X'38')	The ERASE completed successfully, but CA Reclaim failed due to insufficient virtual storage in the address space to dynamically add the CA Reclaim control blocks.

Table 142. Return Codes from the Record-Management(Request) Macros - R15=0 (continued)

RPLRTNCD Code	Condition
60(X'3C')	The ERASE completed successfully, but CA Reclaim failed because the placement of index CI 2 does not follow VSAM convention.
68(X'44')	For LSR, not enough buffers were available so VSAM has successfully added additional buffers to complete the request. For NSR, VSAM added buffers to optimize I/O for spanned records.

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 143 on page 613).

Table 143. 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 144 on page 613 gives the contents of the registers when VSAM exits to the LERAD routine.

Table 144. 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 Using Data Sets* for additional information on the LERAD exit routine.

Table 145 on page 614 gives the logical-error return codes in the feedback field and explains what each means.

Table 145. RPL 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 subsystem. • For non RLS, the record resides on a volume that cannot be mounted.
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. For DFSMStvs, this indicates that DFSMStvs was unable to expand the pool for its context or unit of recovery-related control blocks.
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.
54(X'36')	RPLCAREC	CA reclaim or CA reclaim recovery processing encountered an error.

Table 145. RPL Reason Codes found in the RPL Feedback Field when R15=8 and LERAD is not active. (continued)

RPLERRCD Code	Symbol	Condition
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 and DFSMStvs use.
64(X'40')	RPLNOPLH	• For RLS and DFSMStvs, 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 and DFSMStvs, add GETIX or PUTIX to any data set organization. • For non RLS and DFSMStvs, 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.
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 and DFSMStvs, PUT NUP attempt to change key specified by previous IDALKADD request. • For non-RLS and DFSMStvs, 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 145. RPL 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 and DFSMStvs 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.
106(X'6A')	RPLINVBK	An invalid internal control block has been detected. A diagnostic dump is taken and the write to DASD failed to prevent possible corruption of the data set.
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, not equal to record (slot) length specified for a relative record data set, or longer than 255 spanned-record segments.
109(X'6D')	RPLIXBD2	VSAM has detected an invalid index record.
112(X'70')	RPLKEYLC	KEYLEN specified was too large or equal to 0.
116(X'74')	RPLINLRQ	• Available for RLS and DFSMStvs. • 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 and DFSMStvs. • For an application that does not use RLS or DFSMStvs, a request was canceled from a user JRNAD exit.
128(X'80')	RPLIXHHP	Index is invalid, request cannot be completed.

Table 145. RPL Reason Codes found in the RPL Feedback Field when R15=8 and LERAD is not active. (continued)

RPLERRCD Code	Symbol	Condition
132(X'84')	RPLSRLOC	• Available for RLS and DFSMStvs. • For an application that does not use RLS or DFSMStvs, an attempt was made in locate mode to retrieve a spanned record.
136(X'88')	RPLARSRK	• Available for RLS and DFSMStvs. • For an application that does not use RLS or DFSMStvs, 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	• For LSR, not enough buffers are available to process the request and more could not be added dynamically. • For GSR, not enough buffers are available to process the request. • For RLS and DFSMStvs, the dataspace buffer pool was exhausted.
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. 6. A buffer is constantly invalidated while running with RLS. Among the likely causes are: • The CF cache structure is too small • The application constantly updates the same control interval.
158(X'9E')	RPLNOACC	For RLS, this system has found the data set corrupted and cannot access it. Please close the data set, restore it, and re-open to re-access it if necessary.
160(X'A0')	RPLBMWER	A request was issued to invalidate a modified buffer. For RLS, the 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 that 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.
165(X'A5')	RPLQNONR	For RLS and DFSMStvs, either the IDARECOV request was specified as TYPE=LL and the sphere was not in lost locks state for this subsystem, or TYPE=NONRLS was specified and the sphere was not in NONRLSUPDATE permitted state.
167(X'A7')	RPLQNSUP or RPLQINVR	For RLS, IDAQUIES types QUIBWO, QUICOPY, caller is not in supervisor state; invalid Quiesce request; or the eye-catcher in IFGQUIES is invalid.
168(X'A8')		For RLS and DFSMStvs, the RPLAREA was 0.
169(X'A9')	RPLQCTGF	For RLS and DFSMStvs, the IDAQUIES, IDARETLK TYPE=SS, IDARECOV TYPE=LL, or IDARECOV TYPE=NONRLS request failed because the Catalog Locate command issued for the specified sphere or component name failed.

Table 145. RPL Reason Codes found in the RPL Feedback Field when R15=8 and LERAD is not active. (continued)

<table>
<tr><th>RPLERRCD Code</th><th>Symbol</th><th>Condition</th></tr>
<tr><td><b>170(X'AA')</b></td><td>RPLQUNFL</td><td>The QUIOPEN, QUICEND, or QUIBEND request is rejected because the requested unquiesce operation is already started for this sphere.</td></tr>
<tr><td><b>172(X'AC')</b></td><td>RPLNSMSV</td><td>For RLS and DFSMStvs, the IDAQUIES, IDARETLK TYPE=SS, IDARECOV TYPE=LL, or IDARECOV TYPE=NONRLS request failed because the specified sphere is not an SMS VSAM data set.</td></tr>
<tr><td><b>176(X'B0')</b></td><td>RPLACBBD</td><td>For RLS and DFSMStvs, the shared-latch obtain failed for the Record Management request. Or the ACB specified in the record management request or in the IDARETLK TYPE=SS, IDARECOV TYPE=LL, or IDARECOV TYPE=NONRLS request is not a valid ACB open for RLS or DFSMStvs.</td></tr>
<tr><td><b>180(X'B4')</b></td><td></td><td>For RLS, an invalid request for a nonrecoverable data set.</td></tr>
<tr><td><b>181(X'B5')</b></td><td>RPLQRACF</td><td>This IDAQUIES request is rejected because the requestor does not have update authority to the sphere. Issued for the following:
<ul>
<li>This is a type QUICLOSE request. Successful completion of the request results in a catalog update to mark the sphere quiesced. Because the requestor does not have update authority, the request is rejected.</li>
<li>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. Because the requestor does not have update authority, the request is rejected.</li>
</ul></td></tr>
<tr><td><b>182(X'B6')</b></td><td>RPLQINPR</td><td>For RLS, the 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:
<pre>
 Offset Length Description
 ------ ------ -----------
   0   1   Type of quiesce event already in progress for this
sphere.
                 Quiesce type constants are defined in IFGQUIES
macro.
</pre></td></tr>
<tr><td><b>183(X'B7')</b></td><td>RPLMIGRA</td><td>IDAQUIES request rejected because data set is migrated. .</td></tr>
<tr><td><b>184(X'B8')</b></td><td>RPLABEND</td><td>For RLS, 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.</td></tr>
<tr><td><b>185(X'B9')</b></td><td></td><td>For RLS, the user task was canceled while the request was being processed.</td></tr>
<tr><td><b>186(X'BA')</b></td><td>RPLEOVER</td><td>End of volume initialization failed when DATASET tried to extend.</td></tr>
<tr><td><b>186(X'BA')</b></td><td>RPLEOVER</td><td>For RLS, end-of-volume initialization failed when DATASET tried to extend.</td></tr>
<tr><td><b>187(X'BB')</b></td><td></td><td>For RLS, an error occurred with partial EOV processing.</td></tr>
<tr><td><b>188(X'BC')</b></td><td>RPLNO241</td><td>For RLS, the storage in subpool 241 is not available.</td></tr>
<tr><td><b>189(X'BD')</b></td><td></td><td>For RLS, a lock for the VSAM request required space in the record table, which is full. You must modify the CFRM policy and rebuild the lock structure.</td></tr>
<tr><td><b>192(X'C0')</b></td><td>RPLIRRNO</td><td>Invalid relative record number.</td></tr>
<tr><td><b>193(X'C1')</b></td><td>RPLFLDMP</td><td>An RLS request failed during a read I/O and a dump was generated without terminating the VSAM server address space.</td></tr>
<tr><td><b>196(X'C4')</b></td><td>RPLRRADR</td><td>An addressed request was issued to a relative record data set.</td></tr>
<tr><td><b>200(X'C8')</b></td><td>RPLPAACI</td><td>Addressed or control-interval access was attempted through a path.</td></tr>
<tr><td><b>201(X'C9')</b></td><td>RPLDNOTF</td><td>For RLS or DFSMStvs, the IDARETLK TYPE=SS request failed because the specified data set does not exist.</td></tr>
<tr><td><b>204(X'CC')</b></td><td>RPLPUTBK</td><td>PUT-insert requests (or for RLS, IDALKADD requests) are not allowed in backward mode.</td></tr>
<tr><td><b>205(X'CD')</b></td><td></td><td><ul>
<li>DFSMStvs was unable to complete the request because DFSMStvs restarted while the unit of recovery was inflight. To continue processing, the application must issue a commit or a backout, then begin a new unit of recovery.</li>
<li>For LSR, invalid CONTOKEN.</li>
</ul></td></tr>
</table>

Table 145. RPL Reason Codes found in the RPL Feedback Field when R15=8 and LERAD is not active. (continued)

RPLERRCD Code	Symbol	Condition
206(X'CE')		• For DFSMStvs, indicates that the request was rejected because the data set is quiesced or quiescing for copy. Retry the request. • For applications that do not use RLS or DFSMStvs, this is a validity check error for share 3,4.
207(X'CF')		For DFSMStvs, indicates that transactional processing is currently unavailable because DFSMStvs is disabling or quiescing. Close all data sets so the process can complete.
208(X'D0')	RPLINVEQ	Invalid ENDREQ request.
209(X'D1')		• For DFSMStvs, indicates that the forward recovery log is unavailable because it is disabling. • For LSR, indicates a cache structure failure.
210(X'D2')		• For DFSMStvs, indicates that forward recovery logging failed because the record length is greater than the installation maximum supported by the log. • For shared resources, the buffer is being invalidated, or the buffer use chain changing.
211(X'D3')		• For DFSMStvs, indicates that a permanent I/O error was detected in the forward recovery log. For appropriate action, see the accompanying DFSMStvs logger messages. • For LSR, indicates that the cache request was purged.
212(X'D4')	RPLNOSPL	Unable to split index during a CA split.
213(X'D5')		• For DFSMStvs, indicates that the undo log is unavailable for processing. • For LSR, indicates no connectivity to the cache structure.
214(X'D6')		For DFSMStvs, indicates that a permanent I/O error was detected in the undo log. For appropriate actions, see the accompanying DFSMStvs logger messages.
216(X'D8')		For RLS, the LUWID specified in the RPL does not exist for the subsystem name specified in the ACB.
217(X'D9')		DFSMStvs is unable to complete the request because RRS, which had been available, went down and restarted. To continue processing, the application must issue a commit or a backout and then begin a new unit of recovery.
218(X'DA')		Unrecognizable return code from SVC 109.
220(X'DC')		For DFSMStvs, this reason code is no longer being used.
224(X'E0')	RPLMOIB	A MRKBFR request was issued for an invalid buffer.
228(X'E4')	RPLINVMD	One of the following: • Caller in cross-memory or SRB mode is not in supervisor state • RPL of caller in SRB or cross-memory mode does not specify SYN processing • NSR SRB-mode request without ACB ICI specifies ACB OUT. For MACRF=RLS: • Caller is not in primary ASC mode • Caller issued a record management request with an FRR in effect • Task that opened the ACB is not in the caller task hierarchy.
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. • For RLS and DFSMStvs, this is an internal error.
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.

Table 145. RPL Reason Codes found in the RPL Feedback Field when R15=8 and LERAD is not active. (continued)

RPLERRCD Code	Symbol	Condition
246(X'F6')	RPLCMSDE	Severe error returned from CMS for a decompress call.
247(X'F7')		Error in the last active record number (DFM).
248(X'F8')	RPLRST14	Register 14 return offset is negative.
249(X'F9')		• For DFSMStvs, indicates that undo logging failed because the record length is greater than the installation maximum supported by the log. • For LSR XI, indicates an invalid vector token.
250(X'FA')	RPLINVDT	No valid directory token exists. The data set cannot be decompressed.
251(X'FB')	RPLINTER	Internal VSAM RLS 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.

Reason codes associated with R15=12

Table 146. Reason codes associated with R15=12

RPLRTNCD code	Condition
4(X'04')	Read error for data component.
8(X'08')	Read error for index component.
12(X'0C')	Read error for sequence set.
16(X'10')	Write error for data component.
24(X'18')	Write error for sequence set.
28(X'1C')	Available.
32(X'20')	Available.
36(X'24')	For RLS, coupling facility cache connectivity loss.
40(X'28')	For RLS, coupling facility cache structure failure.
44(X'2C')	For extended function data sets, the suffix for a physical record in the CI, at the RBA specified in the RPL, is invalid.

Reason codes associated with R15=16

Table 147. Reason codes associated with R15=16

RPLRTNCD code	Condition
12(X'0C')	DFSMStvs processing is currently unavailable because DFSMStvs is initializing.

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 148 on page 621 gives the contents of the registers when VSAM exits to the SYNAD routine.

Table 148. 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 149 on page 621 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 149. 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 150 on page 621 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 150. 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)

Table 150. Format of Physical-Error Messages (continued)

Field	Bytes	Length	Discussion
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.
	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'

Table 150. Format of Physical-Error Messages (continued)

Field	Bytes	Length	Discussion
If the type of the unit check can be determined, this message is replaced by one of the following:			
			'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 blocks. 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, with the return address in register 14.

Register 15 contains a return code, which is explained in Table 151 on page 623.

Table 151. 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. Because 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 152 on page 624).

Table 152. 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.
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.

Table 152. Control Block Manipulation Error Codes (continued)

Code	Applicable Macros[1]	Condition
26(X'1A')	S	A request was made using a field name that 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 A. Accessibility

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

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

Accessibility features

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

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

Consult assistive technologies

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

Keyboard navigation of the user interface

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

- *z/OS TSO/E Primer*
- *z/OS TSO/E User's Guide*
- *z/OS ISPF User's Guide Vol I*

Dotted decimal syntax diagrams

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

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

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number 3 is followed by a series of syntax elements with dotted decimal number 3.1, all the syntax elements numbered 3.1 are subordinate to the syntax element numbered 3.

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

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

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

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

? indicates an optional syntax element

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

! indicates a default syntax element

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

*** indicates an optional syntax element that is repeatable**

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

Notes:

1. If a dotted decimal number has an asterisk (*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you can write `HOST STATE`, but you cannot write `HOST HOST`.

3. The * symbol is equivalent to a loopback line in a railroad syntax diagram.

+ indicates a syntax element that must be included

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

Notices

This information was developed for products and services that are offered in the USA or elsewhere.

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

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

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

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

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

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

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

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

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

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

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

IBM Corporation
Site Counsel
2455 South Road

Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

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

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

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

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

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

COPYRIGHT LICENSE:

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

Terms and conditions for product documentation

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

Applicability

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

Personal use

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

Commercial use

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

Rights

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

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

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

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

IBM Online Privacy Statement

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

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

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

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

Policy for unsupported hardware

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

Minimum supported hardware

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

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

Programming interface information

This publication documents information NOT intended to be used as Programming Interfaces of DFSMSdfp.

Trademarks

Index

SC23-6863-30