z/OS
Version 2 Release 3

DFSMS Object Access Method Planning, Installation, and Storage Administration Guide for Object Support

IBM

SC23-6866-30

Note

Before using this information and the product it supports, read the information in "Notices" on page 597.

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: 2019-06-24

Contents

Figures

Tables

About This Document

This document introduces OAM and explains how to do the following tasks:

- Plan for the installation of OAM.
- Install OAM.
- Customize OAM.
- Administer OAM.
- Operate OAM.

This document discusses using OAM's object support which you can use to manage objects on disk (DB2 or file system), optical, and tape volumes. For information on using OAM to manage tape library dataservers, see the *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

This document is for the system programmers, storage administrators, and system operators who perform these tasks.

Required product knowledge

To use this document effectively, you should be familiar with:

- Data Facility Storage Management System (DFSMS)
- Customer Information Control System (CICS®)—optional, depending on your installation
- IBM DATABASE 2™ (DB2®)
- Information Management System (IMS)—optional, depending on your installation
- Interactive Storage Management Facility (ISMF)
- Network File System (NFS)--optional, depending on your installation
- zFS--optional, depending on your installation
- z/OS UNIX--optional, depending on your installation
- File systems--optional, depending on your installation

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).

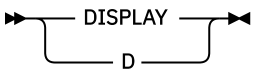
How to Read Syntax Diagrams

There is one basic rule for reading the syntax diagrams: Follow only one line at a time from the beginning to the end and code everything you encounter on that line.

The following rules apply to the conventions that are used in the syntax diagrams for all the OAM commands:

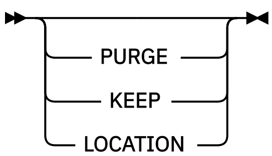
- Read the syntax diagrams from left to right and from top to bottom.

- Each syntax diagram begins with a double arrowhead (►►) and ends with opposing arrows (►◄).
- An arrow (—►) at the end of a line indicates that the syntax continues on the next line. A continuation line begins with an arrow (►—).
- Commands and keywords are shown in uppercase letters.
- Some commands and keywords have alternative abbreviations; these appear as part of the stack for that command or keyword. For example, the alternative abbreviation for `DISPLAY` is `D`.



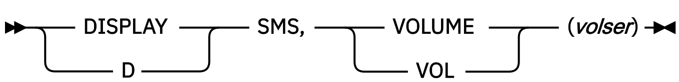
- Where you can choose from two or more keywords, the choices are stacked one above the other. If one choice within the stack lies on the main path, a keyword is required, and you must choose one. In the following example you must choose either `DETAIL` or `STATUS`.

- If a stack is placed below the main path, a keyword is optional, and you can choose one or none. In the following example, `PURGE`, `KEEP`, and `LOCATION` are optional keywords. You can choose any one of the three.



- Where you can choose from two or more keywords and one of the keywords appears above the main path, that keyword is the default. You may choose one or the other of the keywords, but if none is entered, the default keyword is automatically selected. In the following example you may choose either `DETAIL` or `STATUS`. If neither is chosen, `STATUS` is automatically selected.

- Words or names in italicized, lowercase letters represent information that you supply. The values of these variables may change depending on the items to which they refer. For example, *volser* refers to the serial number of a volume, while *storgrp_name* refers to the name of a storage group.
- You must provide all items enclosed in parentheses ( ). You must include the parentheses. In the following example, you must supply the volume serial number (*volser*) and it must be enclosed in parentheses.



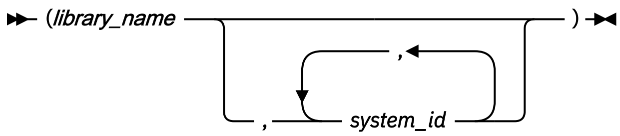
You would code this as follows:

```
D SMS,VOL(volser)
```

The variable *volser* is the serial number of the volume you wish to display.

- The repeat symbol shown below indicates that you can specify keywords and variables more than once. The repeat symbol appears above the keywords and variables that can be repeated. For example, when a comma appears in the repeat symbol, you must separate repeated keywords or variables with a comma.

In the following example, you may specify the *library_name* and one or more system identification numbers (*system_id*) that are separated by commas. You must enclose the name of the library and all of the system IDs in parentheses.



You would code this as follows:

```
(library_name, system_id, system_id, system_id)
```

The variable *library_name* is the name of the library you are working with, and *system_id* names three different instances of system identification numbers.

- Some diagrams contain *syntax fragments*, which serve to break up diagrams that are too long, too complex, or too repetitious. Syntax fragment names are in mixed case and are shown in the diagram and in the heading of the fragment. The fragment is placed below the main diagram.

Syntax fragment name

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 xxiii.

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 DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support, SC23-6866-30
  - 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

This edition contains updates for Version 2 Release 3 (V2R3).

New information

This edition includes the following new information:

- Added to show comparison of classic and multiple OAM configurations. See "Comparing multiple OAM and classic OAM configurations" on page 2 for more information.
- Update to provide guidance on interacting with multiple OAM address spaces. See "Operator command applicability in a multiple OAM configuration" on page 261 for more information.
- Added new section describing encoding scheme necessary for OAM packages. See "DB2 default encoding scheme" on page 101 for more information.
- Added new section describing MVS LIBRARY Command Applicability. See "MVS LIBRARY command applicability" on page 263 for more information.

Changed information

This edition includes the following topics that contain changed information:

- Changed by removing catalog reference. See "Collections" on page 3 for more information.
- Changed by removing sentence regarding instances of OAM in an OAMplex. See "OAMplex" on page 8 for more information.
- Changed by removing list item regarding OAM DB2 database. See "OAMplex restrictions" on page 9 for more information.
- Changed by updating CBR1100I OAM status. See "Displaying OAM Status" on page 306 for more information.
- Changed by differentiating between a classic OAM configuration and a multiple OAM configuration. See "OAM address space" on page 34 for more information.
- Changed by updating introductory paragraph to discuss configuration implications and remove catalog references. See "OTIS address space" on page 35 for more information.
- Changed by removing the 'Catalogs' list item. See "Setting up the planning team" on page 61 for more information.
- Changed by updating the table describing OAM product section fields. See "OAM SMF record product section" on page 516 for more information.
- Changed by adding DB2ID keyword. See "OSREQ TSO/E command syntax" on page 243 for more information.
- Changed by adding new row for 'Refresh OTIS DB2 Tasks'. See "Overview of operator tasks" on page 255 for more information.
- Changed by adding a discussion of classic OAM configuration versus multiple OAM configuration. See "Analyzing your business environment" on page 62 for more information.
- Changed by updating list item to address a multiple OAM configuration. See "System software" on page 70 for more information.

- Changed by removing reference to collection entries in the catalog. See "DASD storage" on page 72 for more information.
- Changed by adding a note to address the possibility of a multiple OAM configuration. See "Planning to use a file system" on page 80 for more information.
- Changed by updating paragraph to reflect removal of OAM collection entries from the catalog. See "Coding ACS routines to implement class transitions" on page 87 for more information.
- Table 21 on page 96
- Changed by removing step "Defining user catalogs". See "High-level installation and migration checklists" on page 97 for more information.
- Changed by updating the statement about preparing the Unix file system hierarchy. See "3 Configure the z/OS Unix file system" on page 103 for more information.
- Changed by updating syntax and keyword descriptions to address the possibility of a multiple OAM configuration. See "5 Changing system libraries" on page 104 for more information.
- Changed SETDISK parameters by updating L2DIR keyword description to address the possibility of a multiple OAM configuration. See "SETDISK parameters" on page 108 for more information.
- Changed by updating keyword descriptions. See "SETOPT keyword definitions" on page 132 for more information.
- Changed by adding note regarding keywords and values in a multiple OAM configuration. See "SETOSMC statements for use in the OSMC environment" on page 138 for more information.
- Changed by updating keyword descriptions. See "SETOSMC keyword definitions" on page 140 for more information.
- Changed by updating syntax and keyword descriptions. See "ONLYIF statements" on page 143 for more information.
- Changed by updating description to address a multiple OAM configuration and introducing a new keyword. See "5h Updating the PROCLIB" on page 146 for more information.
- Changed by updating description to reflect the removal of OAM collection entries from the catalog. See "8 Merging object tables and OCDB for an OAMplex" on page 154 for more information.
- Changed by adding paragraph regarding the prevention of bind errors. See "9 Creating and binding DB2 packages" on page 155 for more information.
- Changed by adding &DB2SSID read-only variable to the list of read-only variables. See "ACS routine input variables" on page 167 for more information.
- Changed by removing collection definitions from the catalog. See "Storing objects in a collection" on page 170 for more information.
- Changed by adding the OAM DB2 SSID parameter and its associated description. See "Defining storage groups and relating the libraries to the storage groups" on page 170 for more information.
- Changed by generalizing notes regarding the existing OAM subsystem identifications. See "Defining management classes" on page 175 for more information.
- Changed by removing steps that describe copying the Integrated Catalog Facility and moving collection definitions into it. See "Procedures for moving OAM to another system" on page 179 for more information.
- Changed by removing reference to OAM collections. See "Merging OAMs into an OAMplex" on page 180 for more information.
- Changed by including instructions for a multiple OAM configuration. See "Adding OAM systems to an existing OAMplex" on page 180 for more information.
- Changed by reflecting changes of OAMUTIL CHGCOL utility. See "Modifying default storage and management classes" on page 199 for more information.
- Changed by reflecting changes of OAMUTIL CHGCOL utility. See "The CHGCOL procedure" on page 199 for more information.
- Changed by reflecting changes of OAMUTIL CHGCOL utility. See "Manual procedure" on page 199 for more information.

- Changed by generalizing the existing OAM subsystem identification. See "Tuning object retrieval response time" on page 208 for more information.
- Changed by generalizing the existing OAM subsystem identification. See "Tuning the storage management cycle" on page 210 for more information.
- Changed by removing list item referring to collection name entries in a catalog. See "Establishing recovery procedures" on page 217 for more information.
- Changed by removing list items. See "Recovering DB2 databases" on page 217 for more information.
- Changed by removing a list item. See "Documentation for your IBM representative" on page 240 for more information.
- Changed by removing a list item. See "Documentation for your IBM representative" on page 242 for more information.
- Changed by adding a note regarding the OAMUTIL REFORMAT utility. See "Reformatting a 3995 optical disk" on page 303 for more information.
- Changed by updating syntax and keywords. See "OSREQ TSO/E command syntax" on page 243 for more information.
- Changed by adding an error code for the DB2ID keyword. See "OSREQ TSO/E command processor return codes" on page 252 for more information.
- Changed by updating to add the prefix loc. to all package names. See "CBRPBIND" on page 461 for more information.
- Changed by updating table describing the OAM product section fields. See "OAM SMF record product section" on page 516 for more information.
- Changed by updating descriptions of the CBRADUP fields. See "Auto-delete installation exit parameter list" on page 575 for more information.
- Changed by adding definitions for 'OAM instance,' 'Multiple OAM configuration,' and 'Classic OAM configuration.' See "Glossary" on page 601 for more information.

No longer issued information

This edition includes the following topics that are no longer issued:

- Deleted section 'Defining user catalogs' from Chapter 3.
- Deleted section 'Recovering Collection name catalog entries' from Chapter 4.
- Deleted two sections of 'Inconsistencies between the catalog and the collection table' from Chapter 4.
- Deleted sample migration jobs CBRSMKBO, CBRSMB2, CBRSMPDS, and CBRSMVAF from Appendix B.

Summary of changes for z/OS Version 2 Release 2

This edition contains updates for Version 2 Release 2 (V2R2).

New information

This edition includes the following new information:

- Support for 64-bit virtual address buffers (BUFFER64 on OSREQ RETRIEVE and OSREQ STORE) is described in:
  - "OSREQ TSO/E command syntax" on page 243
  - "OSREQ activity subtypes 1–10 data section format" on page 517

Changed information

This edition includes the following topics that contain changed information:

- "Updating fields in the DB2 Volume Table and the Tape Volume Table" on page 374

Summary of changes for z/OS Version 2 Release 1

This edition contains updates for Version 2 Release 1 (V2R1).

The following sections summarize the changes to that information.

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

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

Changed Information

This edition includes the following changed information:

- "SETOAM statements for object tape storage" on page 109 has been updated for the new global level keywords TAPESDB and ALLOCRETRYMINUTES.
- "SETOSMC statements for use in the OSMC environment" on page 138 has been updated for the new global level keyword BACKUPDELETE.
- In "Tape volume table (TAPEVOL)" on page 504, VOLATTRF has been updated for TAPESDB.
- "Protecting against inadvertent object deletion" on page 11 has been rewritten.
- "High-level installation and migration checklists" on page 97 has new steps added.

These topics have been updated to reflect 3592 Model E07 read-compatibility processing:

- Table 4 on page 45
- "Tape encryption support" on page 52
- "Glossary" on page 601

Deleted Information

The topic "Optical devices that OAM no longer supports" has been deleted.

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. Understanding the Object Access Method

The object access method (OAM) uses a class of data referred to as objects. An *object* is a named stream of bytes. The content, format, and structure of that byte stream are unknown to OAM. There are no restrictions on the data in an object. For example, an object can be a compressed scanned image or coded data. Objects are different from data sets handled by existing access methods. The following characteristics distinguish them from traditional data sets:

- **No record orientation.** There are no individual records within an object.
- **Broad range of size.** An object can contain 1 byte or up to 2000 MB (2,097,152,000 bytes) of data. The maximum object size for the DB2 sublevel, file system sublevel, tape sublevel 1, and tape sublevel 2 of the OAM storage hierarchy is 2000 MB. The maximum object size for the optical level of the OAM storage hierarchy is 256M. See Updating the IEFSSNxx PARMLIB member.
- **Volume.** Objects are usually much smaller than data sets; however, they are more numerous and consume vast amounts of external storage.
- **Varying access-time requirements.** Reference patterns for objects change over time or cyclically, allowing less-critical objects to be placed on lower-cost slower devices or media.

This topic covers the following topics related to using OAM to manage objects on disk (DB2 or file system), tape, and optical volumes.

Comparing multiple OAM and classic OAM configurations

You can choose to implement a multiple OAM configuration or a classic OAM configuration.

Multiple OAM Configuration

A multiple OAM configuration allows you to have multiple instances of OAM on a single system. An instance consists minimally of the definition of an OAM subsystem, but typically also includes an associated OAM address space. In a multiple OAM configuration, an OAM address space is used either for tape library-related processing or for object processing. It is also possible to have multiple OAM address spaces on a single z/OS system. In a multiple OAM configuration, you can have multiple object address spaces and one separate tape library address space. Each object address space is associated with a unique DB2 subsystem. Additionally each object address space on a z/OS system in a multiple OAM configuration can optionally be a member of a unique OAMplex configuration.

In a multiple OAM configuration you can implement any of the following:

- Single OAM object instance without a tape library instance
- Single OAM tape library instance without any object instances
- Single OAM object instance with a tape library instance
- Multiple OAM object instances without a tape library instance
- Multiple OAM object instances with a tape library instance.

Even if you do not require multiple OAM object instances, you might consider a multiple OAM configuration. For example, you might want object related processing to be isolated in an object address space and tape library processing to be isolated in a tape library address space. This isolation optimizes processing and might make more resources available in each type of OAM address space. Additionally, this isolation allows an OAM object address space to restart without affecting the processing in a tape library address space. These benefits must be weighed against the need for managing multiple address spaces when both an object address space and a tape library address space are implemented.

In a multiple OAM configuration, the optical level of the OAM storage hierarchy is not supported within an object address space. Therefore, customers who want to implement a multiple OAM configuration must first take the following actions before attempting to implement a multiple OAM configuration:

- Transition any existing objects from the optical level of the OAM storage hierarchy to another level of the OAM storage hierarchy (required)
- Remove all optical related definitions for libraries and drives (recommended; warning messages are issued if this is not done)
- Update definitions in the CBROAM*xx* member of PARMLIB to remove any optical related keywords (recommended; if specified, optical related keywords are ignored).

Classic OAM Configuration

Customers who choose not to implement a multiple OAM configuration can instead use a classic OAM configuration. In a classic OAM configuration, only one instance of OAM exists. It consists of the definition of an OAM subsystem and optionally an associated OAM address space that can be used for both tape library and object processing. An OAM address space that is used for object processing in a classic OAM configuration is associated with a single DB2 subsystem and can optionally be a member of a single OAMplex configuration.

Switching Between Classic and Multiple OAM Configurations

This support was designed to be able to switch between classic and multiple OAM configurations without requiring an IPL. See Removing one or all OAM subsystems from the OAM configuration for the MODIFY OTIS command capability DELSUB. Once the OAM subsystem(s) are removed, the new subsystem(s) can be added using the SETSSI command. The corresponding changes to the OAM started procedure (for the OAM address space(s)) also need to match the changes being made to the corresponding OAM subsystem(s). For the OAM started procedure, see Modify, if necessary, then run CBRAPROC SAMPLIB job.

For both the OAM subsystem(s) and the OAM started procedure(s), the presence of the D= keyword for the DB2 SSID or group attachment name determines whether OAM comes up in a classic or a multiple OAM configuration.

Collections

In OAM, a *collection* is a group of objects typically having similar performance, availability, backup, retention, and class transition characteristics. You can use a collection to group a large number of objects. Every object must be assigned to a collection. Object names within a collection must be unique. However, the same object name can be used in multiple collections. Each collection belongs to only one Object storage group. Each storage group can contain from one to many collections.

Note: Catalog is no longer used for collections starting with z/OS V2R1 (coexistence enablement APAR OA51129) as well as the base release for z/OS V2R3.
Prior to this, the collection information was redundantly maintained in both the catalog and in DB2.
Information APAR II14842 should be reviewed before migrating to release V2R3 or higher. If II14842 precautions are not observed, then there can be a potential loss of object accessibility.
Significant changes were made to the way OAM handles collection entries that can surface an existing collection entry out of synch condition that can be present in your environment between the catalog and DB2 collection table.
This out of synch condition is likely caused by manual SPUFI changes, at any point in time, to the DB2 collection table.
Please review II14842 for additional details.

Application programming interface (OSREQ macro)

OAM provides an application programming interface known as the OSREQ macro. The OSREQ macro sets up (ACCESS) the environment for a user to change, store, retrieve, delete, and query information about an object, and then releases (UNACCESS) the resources required for this macro when they are no longer needed. OAM includes the functions necessary to manage the objects after they are stored.

Related reading: For more detailed information on the OSREQ macro, see *z/OS DFSMS OAM Application Programmer's Reference*.

System-managed storage for data and space management

OAM is a component of DFSMSdfp™, the base for the Storage Management Subsystem (SMS). OAM uses system-managed storage, which provide functions for data and space management. SMS provides the following benefits:

- Manages storage growth
- Improves the use of storage space
- Reduces the effort of device conversion and coexistence
- Provides centralized control of external storage
- Exploits the capabilities of available hardware

Use SMS to define a storage hierarchy for objects and the parameters for managing those objects. OAM uses this hierarchy definition and management parameters to place user-accessible objects anywhere in the SMS storage hierarchy.

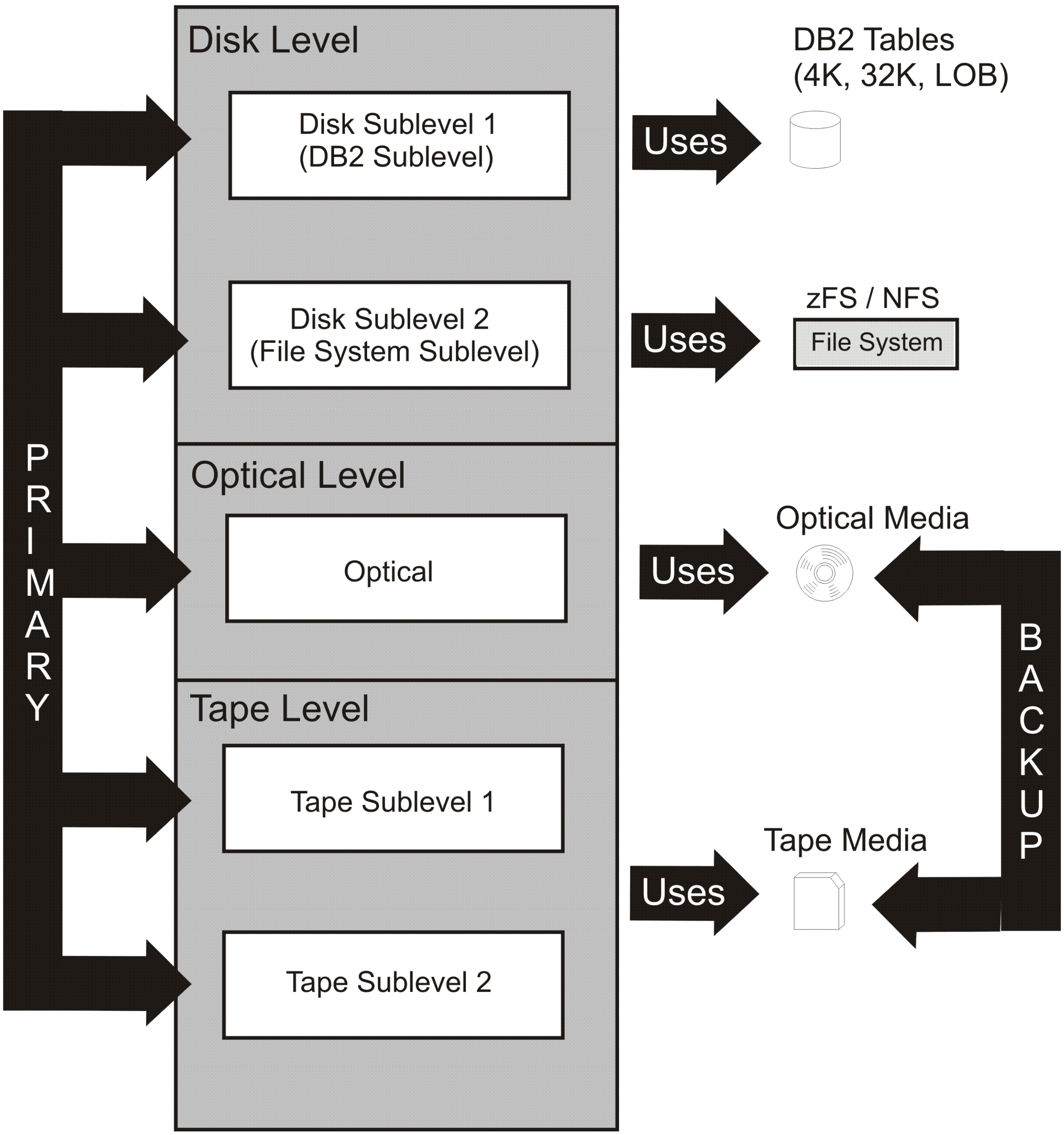


Figure 1. Object storage hierarchy

The object storage hierarchy can consist of:

- Disk sublevel 1, which is associated with DB2 tables on a direct access storage device (DASD). The following object storage DB2 tables provide disk sublevel 1 storage for objects:
  - 4 KB storage table
  - 32 KB storage table
  - LOB storage structure
- Disk sublevel 2, which is associated with an NFS or z/FS file system.
- Tape sublevel 1 volumes associated with a tape library device (SMS-managed, library-resident tape volumes), and tape volumes outside of a library device (non-SMS-managed, shelf-resident tape volumes)

- Tape sublevel 2 volumes associated with a tape library device (SMS-managed, library-resident tape volumes), and tape volumes outside of a library device (non-SMS-managed, shelf-resident tape volumes)
- Optical volumes inside a library device (SMS-managed, library-resident optical volumes), and optical volumes outside of a library device (SMS-managed, shelf-resident optical volumes)

Moving objects throughout the hierarchy

During the storage management cycle, OAM determines whether the primary copy of an object is correctly positioned in the OAM storage hierarchy. If the object is not correctly positioned in the OAM storage hierarchy, the primary copy of the object is moved to the correct storage medium. One of the following medium transitions can be performed for the primary copy of an object:

- Disk sublevel 1 to disk sublevel 2
- Disk sublevel 1 to optical
- Disk sublevel 1 to tape sublevel 1
- Disk sublevel 1 to tape sublevel 2
- Disk sublevel 2 to disk sublevel 1
- Disk sublevel 2 to optical
- Disk sublevel 2 to tape sublevel 1
- Disk sublevel 2 to tape sublevel 2
- Optical to disk sublevel 1
- Optical to disk sublevel 2
- Optical to tape sublevel 1
- Optical to tape sublevel 2
- Tape sublevel 1 to disk sublevel 1
- Tape sublevel 1 to disk sublevel 2
- Tape sublevel 1 to optical
- Tape sublevel 1 to tape sublevel 2
- Tape sublevel 2 to disk sublevel 1
- Tape sublevel 2 to disk sublevel 2
- Tape sublevel 2 to optical
- Tape sublevel 2 to tape sublevel 1

The location of an object in the hierarchy is unknown to the user of the programming interface. You do not need to supply device-dependent information. For example, there are no JCL DD statements and no requirements for device geometry, such as track size.

Note: Optical locations are not supported in multiple OAM configurations.

Related reading: See "Media selection for object storage" on page 24 for information regarding the criteria that is used in placing objects onto the appropriate media type.

Hardware and software interaction with OAM

Figure 2 on page 7 shows the hardware and software in a typical OAM object environment and illustrates the possible interactions. The OAM object environment is closely tied to SMS and DB2. OAM also interacts with the OAM thread isolation support (OTIS). OTIS is an OAM subsystem that provides OAM-to-DB2 functions that use a different thread to DB2 than the application program thread. Applications can use OSREQ, the application programming interface, to interact with OAM. Applications can also communicate directly with DB2. Each application is responsible for synchronizing its DB2 databases, whether the operation is generated by the application or by OAM.

In a multiple OAM configuration, OTIS starts with the first OAM subsystem (can be Tape Library or Object). OTIS is required to be active to remove OAM subsystems by F OTIS,DELSUB command.

Note: The artwork in this document uses tape cartridge and optical disk symbols to depict tape and optical storage libraries and devices. For a detailed list of the devices that OAM uses, see "Hardware" on page 68.

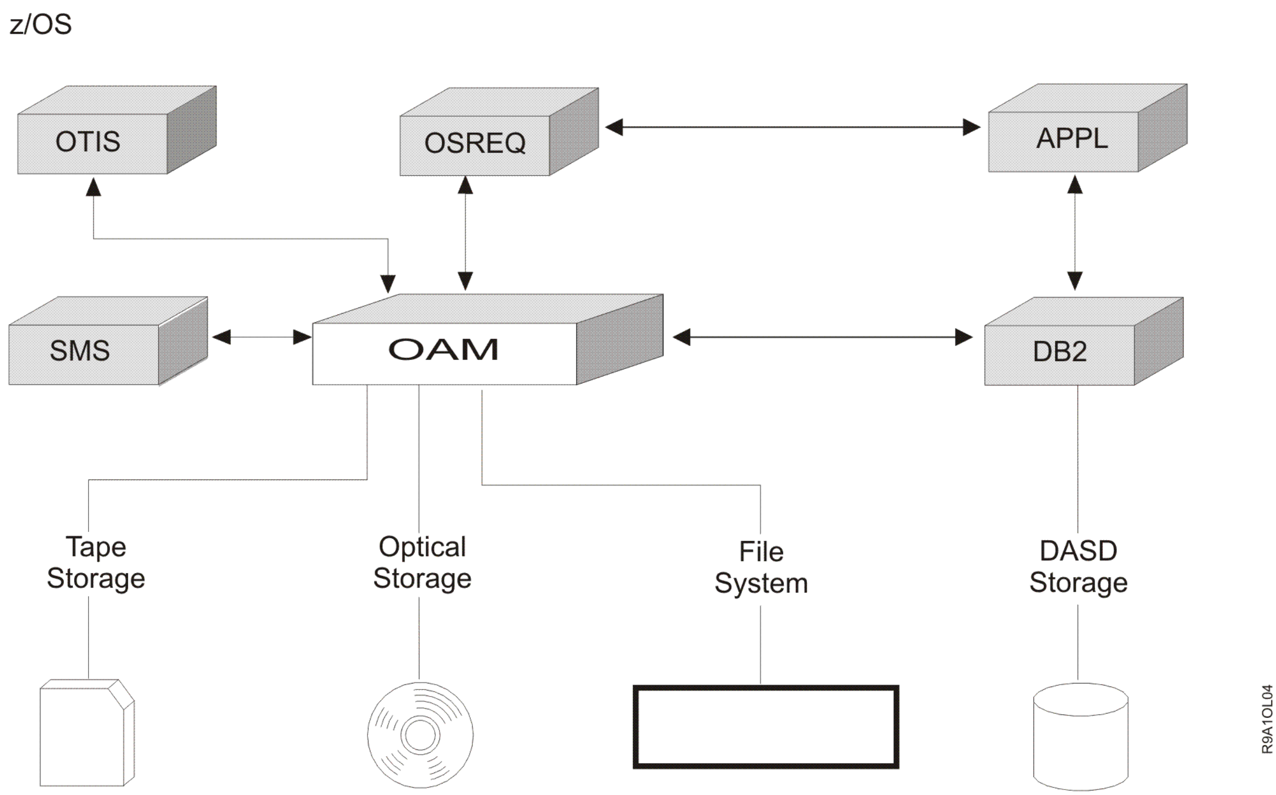


Figure 2. OAM Interactions with Software and Hardware

Related reading: For more information on OSREQ, see *z/OS DFSMS OAM Application Programmer's Reference*.

Parallel Sysplex® and OAM

OAM functions in a z/OS Parallel Sysplex. A Parallel Sysplex links many systems together and provides multisystem data sharing through the use of the cross-system coupling facility (XCF) component of z/OS. XCF services allow authorized applications on one system to communicate with applications on the same system or on other systems. XCF services also allow data to be shared between the applications on these systems. The system linking and multisystem data sharing makes the sysplex platform ideal for parallel processing. In a Parallel Sysplex, objects can be accessed from all instances of OAM and from optical hardware within the sysplex, and transactions can be processed more efficiently.

Related reading: For more information regarding a Parallel Sysplex, see Parallel Sysplex Overview (www.ibm.com/systems/z/advantages/pso/sysover.html).

OAMplex

An *OAMplex* consists of one or more instances of OAM running on systems that are part of a Parallel Sysplex. An OAMplex has a one-to-one correlation to an XCF group in a Parallel Sysplex. The XCF group associated with an OAMplex is joined by instances of OAM address spaces, running on separate systems in a Parallel Sysplex, sharing a common OAM database in a DB2 sharing group. Each instance of OAM is a member of the same XCF group. Also, the DB2 subsystems connected to these instances of OAM belong to the same DB2 data sharing group. The instances of OAM belonging to the same XCF group are able to communicate with each other through the XCF services. The DB2 data sharing group shares the DB2 database information (OCDB, OAMADMIN, and object databases) among OAMs belonging to the OAMplex. When different OAMs sharing a common database on DB2 join an XCF group to become an OAMplex, all object data and configuration information is known to all instances of OAM in the OAMplex. Any object, regardless of which OAM stored the object, can be retrieved by any instance of OAM in the OAMplex.

Requirement: In a Parallel Sysplex, only one OAM XCF group (OAMplex) can share a single common DB2 database. All instances of OAM running in XCF mode in a Parallel Sysplex sharing a common DB2 database *must* join the same XCF group. Multiple OAMplexes can exist within the Parallel Sysplex, but each OAMplex must use a different shared DB2 database. No two OAMplexes can share the same DB2 database. Additionally, OAMs that are not in XCF mode cannot share the DB2 database.

Recommendation: It is strongly recommended that object and object backup storage groups defined as enabled to one system in an OAMplex be enabled to all systems in the OAMplex. Additionally, it is recommended that the rules associated with an object or object backup storage group (with statements in the CBROAMxx parmlib member) be identical for each system in an OAMplex.

The following scenario illustrates one example of how failure to adhere to these recommendations could result in undesirable results.

- OAMGRP1 is the name of the OAMplex (OAM XCF group name).
- OAMGRP1 contains two OAM members: OAM1 on SYS1 and OAM2 on SYS2.
- Object storage group GROUP01 is enabled to SYS1.
- Object storage group GROUP02 is enabled to SYS2.
- Object backup storage group BACKUP1 is enabled to both SYS1 and SYS2.
- Tape volume VOLB01 belongs to object backup storage group BACKUP1 and contains backup copies of objects for both GROUP01 and GROUP02.
- If a 'MODIFY OAM,START,MOVEVOL,VOLB01,DELETE' command is issued on SYS1 then:
  - OAM1 will move all of the GROUP01 objects off of VOLB01, but since OAM1 does not have visibility to GROUP02, it will NOT move any GROUP02 objects off of VOLBO1.
  - After OAM1 has moved all of the objects (for the storage groups that are enabled to OAM1) off of volume VOLB01, then VOLB01 will be deleted from the OAM tape volume inventory. The net result is that the backup copies of the GROUP02 volumes were not moved and those backup copies are no longer available.

Refer to “Defining storage groups and relating the libraries to the storage groups” on page 170 for more information on enabling an object or object backup storage group. Refer to “5 Changing system libraries” on page 104 and see the step to "Create or Update CBROAMxx PARMLIB members" for more information on CBROAMxx Parmlib statements.

OAM uses the XCF messaging facilities to communicate between systems, synchronize resource information, and coordinate where transactions should be processed.

OAM can be running in XCF mode (in an OAMplex), or non-XCF mode (not in an OAMplex). When OAM is running as part of an OAMplex on a system in a Parallel Sysplex, you must initialize that instance of OAM with a CBROAM*xx* PARMLIB member using the OAMXCF statement, which specifies an XCF member name and an XCF group name.

If all instances of OAM involved with the transaction belong to the same OAMplex, you can retrieve any OAM object from any z/OS system in a Parallel Sysplex. This is allowed regardless of which OAM in the sysplex stored the object or on which medium (DB2 sublevel, file system sublevel, optical, tape sublevel 1, or tape sublevel 2) the object resides. See "Shipping request limitations for larger data objects" on page 9.

The system uses *transaction shipping* to send and receive requests between OAMs within the OAMplex. If each instance of OAM in an OAMplex shares the same configuration, transaction shipping allows any OAM in a Parallel Sysplex to write objects to, retrieve objects from, or delete objects from any 3995 optical volume in the Parallel Sysplex. Requests to read data from or write data to 3995 optical volumes that reside in a 3995 optical library being managed by a different OAM on a separate z/OS system in the same Parallel Sysplex are serviced by sending the request (using XCF) to the OAM running on the z/OS system that controls the 3995 optical library dataserver. This configuration is possible only as long as both the requesting and responding OAMs are members of the same OAMplex. 3995 optical library dataservers are still controlled and managed by a single OAM running on a single z/OS system. If a system failure occurs, you can switch control of a 3995 optical library dataserver to another OAM running on another z/OS system in the same OAMplex.

Requirement: When multiple OAMplexes exist within a Parallel Sysplex, each OAMplex must have a unique set of OAM resources (media for object storage) defined in its configuration.

The object tape environment also uses the basic concept of transaction shipping. However, MVS dynamic allocation handles the required tape resource allocation, because OAM does not control tape resources. Tape resources are allocated as needed and only for the time required for their use.

For object tape processing, tape drives must be available to any OAM in an OAMplex where a tape request may need to be processed. Tape transactions are shipped across systems *only* when the requested tape volume for a *retrieve* request is allocated and mounted on a tape drive that is in use by another OAM in the OAMplex, or when the OAMplex has different support levels (full support versus coexistence support). For example, if a *retrieve* request originated on a coexistence-support system and another system in the OAMplex is available and has full support for the request, the *retrieve* request can be shipped to the full-support system when there are no eligible tape drives on the system where the request originated to satisfy the request. Tape *write* requests are not sent across systems for processing.

Shipping request limitations for larger data objects

Shipping requests to another system requires that OAM obtain storage for that object on the receiving system. To ensure that the OAM address space does not run out of virtual storage when the maximum object size is expanded above 50MB (with the MOS keyword in the OAMx statement of the IEFSSNxx parmlib member) the following limitations are in effect:

- Tape RETRIEVE requests for an object larger than 50MB will not be shipped to the system in which the volume is currently mounted. Instead, they will wait to be processed on the system where the request originated. If the originating system is not capable of honoring the request, the read request will fail on that system. To avoid problems, ensure that all systems in the OAMplex are capable of honoring the request.
- Optical RETRIEVE and STORE requests for objects larger than 50MB must be processed on the system to which the optical library and drives are attached. If not, OAM will fail the request with return code 12, reason code X'0813'.

When planning for data objects larger than 50MB in an OAMplex, consider storing and processing them only on disk (using DB2 tables or a file system) or on tape.

OAMplex restrictions

There are some restrictions with an OAMplex that you should keep in mind:

- Any instance of OAM running on a system in a Parallel Sysplex that is not running in XCF mode cannot share any resources (optical libraries, optical drives, optical volumes, or tape volumes for object data) that another instance of OAM owns.
- Any OAM not running in XCF mode cannot share its DB2 databases with any other instances of OAM.
- Optical libraries that are defined in a source control data set (SCDS) as connected to a system where OAM is not running in XCF mode must be logically connected to only that system.
- When you define optical libraries in an SCDS as logically connected to multiple systems, all instances of OAM on those systems must be part of a single OAMplex.
- See "OAMplex and file systems" on page 10 for additional restrictions on using file systems within an OAMplex.

It is important that these restrictions be implemented and adhered to. OAM cannot detect or prohibit processing that does not conform to these standards, so unexpected results can occur if these restrictions are not strictly enforced.

OAMplex and file systems

When a file system is to be used by OAM in an OAMplex, all systems in the OAMplex must meet the following requirements (in addition to those identified in "OAMplex" on page 8):

- For any storage group that will include the file system sublevel, the object storage group definitions in the SCDS must be set to ENABLE for all systems in the OAMplex.
- The SETDISK statements in the CBROAM*xx* member of PARMLIB must be identical for all systems in the OAMplex. To meet this requirement, it is recommended that the ONLYIF statement be used so that a single CBROAM*xx* member can be used by all systems in the OAMplex.
- The file systems identified in the SETDISK statements must be available to all systems in the OAMplex because OAM file system operations could be performed from any system in the OAMplex. To do this, these file systems must be defined through z/OS UNIX as a *shared file system*. In a shared file system environment, a file system mounted by one system participating in a shared file system is accessible to all systems participating in the shared file system. Establishing a shared file system and using zFS capability for sysplex aware is described in *z/OS UNIX System Services Planning*.

OAM cannot detect or prohibit configurations that do not meet the requirements. An improper configuration can lead to unpredictable results.

OAMplex and Recall to Disk: There are cases when an OAMplex configuration will result in read requests being shipped from one system in the OAMplex to another system in the OAMplex. When the read request is the result of an application OSREQ RETRIEVE request, then it is possible that the object will need to be recalled to the disk level of the OAM storage hierarchy. Whether the object is recalled to the DB2 sublevel or file system sublevel of the disk level of the OAM storage hierarchy depends on the capabilities of both the originating system and the target system involved in the request and on the configuration of the target system:

- If both systems are z/OS V1R13 or above, the recall disk sublevel (DB2 or file system) will be determined by the configuration in place on the target system, not the originating system, consistent with the existing protocol for other OAMplex related processing.
- If both systems are **not** at z/OS V1R13 or above, the target system will recall the object to the DB2 sublevel.

OAMplex and file system deletion processing: In an OAMplex, it should be assumed that file system physical deletions could be performed by any OAM instance in the OAMplex. Physical file system deletions will be accomplished by processing the entries in the File System Delete Table. OAM determines which OAM instance in the OAMplex will perform the physical file system deletions and the actual timing of the physical file system deletions.

OAM components

The functions of OAM are performed by its three components:

1. The **Object Storage and Retrieval (OSR)** component is an application programming interface for OAM. Applications operating in Customer Information Control System (CICS), Information Management System (IMS), TSO, and z/OS use OSR to store, retrieve, query, and delete objects, and to change information about objects. OSR stores the objects in the storage hierarchy and maintains the information about these objects in DB2 databases. OSR functions, invoked through the application programming interface, require the OAM thread isolation support (OTIS) application for administrative processing.
2. The **Library Control System (LCS)** component writes and reads objects on the file system and on tape and optical disk storage, and deletes objects from the file system. It also manipulates the volumes on which the tape and optical objects reside. The LCS component controls the usage of optical hardware resources that are attached to the system.
3. The **OAM Storage Management Component (OSMC)** provides storage management cycle processing which determines where objects should be stored in the OAM storage hierarchy, manages object movement within the object storage hierarchy, manages object expiration attributes that are based on the installation storage management policy that is defined through SMS, creates the requested backup copies of the objects, and manages the expiration of the volumes that contain objects.

   OSMC Utilities and Functionality

   In addition to the storage management functions described above, OSMC provides a number of utilities that are explicitly initiated by an operator command as well as other implicit functionality that can be configured to be performed automatically when events occur such as certain OSREQ API invocations. This publication may generically reference OSMC or the collection of OSMC activities and the following list identifies these OSMC activities - all of which consume system resources and must be considered in planning and operations.

   - Storage management cycle
     - All object storage groups
     - Specific object or object backup storage group
   - Library space management cycle for an optical library
   - DASD space management cycle for an Object storage group
   - Volume Recovery utility
   - Single Object Recovery utility
   - Move Volume utility
   - Recycle utility
   - Immediate Recall of objects to a disk sublevel following an OSREQ retrieve
   - Immediate Backup of objects following their initial OSREQ store

Related reading: For more information on how LCS controls the library management for tape library dataservers (automated and manual), see the *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Protecting against inadvertent object deletion

OAM provides the following mechanisms to help you ensure that objects are not deleted accidentally or prematurely:

CBRHADUX exit
The CBRHADUX auto-deletion user exit can optionally be used to define a set of objects that are not to be deleted by OSMC expiration processing, even though the objects' expiration criteria had been met. The CBRHADUX user exit is invoked only for OSMC expiration processing. It is not invoked, and therefore offers no protection, for OSREQ DELETE requests.

CBRUXSAE exit
The CBRUXSAE authorization user exit can optionally be used to verify the caller has authority to perform an OSREQ DELETE request.

Deletion-hold
When an object is in deletion-hold mode, it cannot be deleted from the OAM inventory (either by OSREQ DELETE or by OSMC expiration processing). An object can be put into deletion-hold mode by a DELHOLD=HOLD parameter on the OSREQ API. It can be released from deletion-hold mode by a subsequent OSREQ API request with the DELHOLD=NOHOLD parameter specified.

Deletion-protection
When deletion-protection is enabled (at a global or storage group level) OAM will not allow an object to be deleted prior to its expiration date. Deletion-protection differs from retention-protection in that deletion-protection can be turned on and off by the installation, and deletion-protection has no restrictions on the expiration date changing.

Event-based-retention
When an object is in event-based-retention mode, its expiration date is not calculated until OAM has received notification that an external event has occurred. An object is placed into event-based-retention mode with an RETPD=-2 (negative 2) parameter on the OSREQ API. At that point its expiration date is set to the special value of 0002-02-02 and the object is waiting for notification of an external event before calculating the expiration date. The external event notification comes in the form of receipt of an OSREQ API request with the EVENTEXP=*nnnn* parameter.

Retention-protection
Retention-protection provides OAM's most stringent protection to ensure that an object has not been modified or deleted prior to its expiration date. When retention-protection is enabled for a given object, OAM will not allow that object to be deleted prior to its expiration date. Additionally, OAM will not allow the expiration date to be changed to an earlier date. It will however, allow the expiration date to be changed to a later date. If an object is stored into an object storage group that has retention-protection enabled, then that object is considered retention-protected for the life of the object. Installations cannot disable retention-protection for a retention-protected object.

Note: OAM does not allow the modification of data for an existing object, so the procedure to modify an object's data is to delete the original object and then store the modified data with the original object name. Therefore, protecting an object from being deleted will also prevent it from having its data modified.

See *z/OS DFSMS OAM Application Programmer's Reference* for more information.

CBROAMxx PARMLIB member statements

The CBROAMxx PARMLIB member establishes the environment under which OAM runs. You can customize the CBROAMxx PARMLIB member by updating it with statements that alter the operating environment independently of ISMF and SMS. The statements include SETDISK, SETOAM, SETOPT, SETOSMC, SETTLIB, OAMXCF, and ONLYIF. Once you have updated the CBROAMxx PARMLIB member with one or more of these statements, you must restart OAM.

You can add comments in columns 1-71 by using the standard /* */ pairs. Do not use columns 72-80.

SETDISK statement for file systems used by OAM

The CBROAM*xx* PARMLIB member contains one or more SETDISK statements to configure the file system sublevel of the disk level in the OAM storage hierarchy. For each object storage group in which a file system sublevel will be defined, a SETDISK statement must be specified to provide the file system type and the file system directory to be used for the storage group. OAM uses these values to store objects in, and retrieve objects from, the file system. The file system type and file system directory for the object storage group must be carefully selected, as these are static values and cannot be changed.

For more information on coding the SETDISK statement, see "SETDISK statements for file system sublevel" on page 108.

Note: The SETDISK statement is the only mechanism in OAM to communicate the file system type and file system directory for the storage group. The file system type and file system directory specified is not recorded by OAM within DB2 database tables. The SETDISK statement must therefore continue to exist to provide access to objects in the file system sublevel. Therefore, backing up the CBROAMxx member of PARMLIB to preserve these critical SETDISK statements must be included in your backup strategy. If a symbolic link is used, the value of the symbolic link should also be included in your backup strategy.

SETOAM statement for object tape volumes

The CBROAM*xx* PARMLIB member contains one or more SETOAM statements. These statements contain keywords that you can use to tailor the object tape function. These statements can supplement or override information that was previously specified when the applicable Object or Object Backup storage group was defined using ISMF. Some keywords apply to all of the Object or Object Backup storage groups that use the object tape function, and others apply only to the group for which they have been explicitly specified.

For more information on coding the SETOAM statement, see "SETOAM statements for object tape storage" on page 109. For more information on changing the SETOAM values dynamically or defining the values when the CBROAMxx PARMLIB member is not used at initialization, see "Updating SETOAM values" on page 367.

SETOPT statement for options

The CBROAM*xx* PARMLIB member contains one or more SETOPT statements. The SETOPT statement and its associated keywords define general rules or OPTIONS at global and storage group levels that OAM uses to span all of the OAM environments of disk, optical, and tape.

For more information on coding the SETOPT statement, see "SETOPT statement for options" on page 13. For more information on changing the SETOPT values dynamically or defining the values when the CBROAMxx PARMLIB member is not used at initialization, see "Updating SETOPT values" on page 370.

SETOSMC statement for OSMC processing

The CBROAM*xx* PARMLIB member contains one or more SETOSMC statements. The SETOSMC statement and its associated keywords determine the valid values of settings, at a global or storage group level, for OSMC processing. They associate an Object storage group with the Object Backup storage group that stores the first or second backup copies of objects. The SETOSMC statement determines which Object Backup storage groups contain the first and second copies of the objects that are associated with an Object storage group. If you do not provide a SETOSMC SECONDBACKUPGROUP statement, and specify a second backup group, OAM does not create second backup copies of objects. SETOSMC statements can also be used to enable and customize object recall to disk processing.

For more information on coding the SETOSMC statement, see "SETOSMC statements for use in the OSMC environment" on page 138.For more information on changing the SETOSMC values dynamically or defining the values when the CBROAM*xx* PARMLIB member is not used at initialization, see "Updating SETOSMC values" on page 371.

SETTLIB statement for tape library settings

The optional SETTLIB statement and its associated keywords in the CBROAMxx PARMLIB member can be used to configure various tape library related settings. For more information, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*

OAMXCF statement for parallel sysplexes

The CBROAM*xx* PARMLIB member contains one or more OAMXCF statements that allow you to use OAM in a Parallel Sysplex. The OAMXCF statement allows you to provide group and member names for OAMplexes and instances of OAM to be associated with various XCF groups and members to allow data sharing within the sysplex. There are also timeout values that are assigned to optical and tape request types to determine the number of seconds that OAM is to wait for completion of a read or write request that was shipped to another OAM within the OAMplex.

For more information on coding the OAMXCF statement, see "OAMXCF statements in an OAMplex" on page 144. For more information on changing the OAMXCF values dynamically or defining the values when the CBROAM*xx* PARMLIB member is not used at initialization, see "Using the UPDATE command to set OAMXCF values" on page 373.

ONLYIF statement for parallel sysplexes

The CBROAMxx PARMLIB member contains one or more ONLYIF statements to allow an installation to share a single CBROAMxx member across all the systems in a sysplex as well as across the multiple OAM address spaces in a specific system, improving usability. See "ONLYIF statements" on page 143 for more information.

ISMF library management role within OAM

Use the Interactive Storage Management Facility (ISMF) to integrate OAM into system-managed storage. Use the Library Management application of ISMF, which interacts with DB2, to define optical disk drives and libraries into the OAM configuration database. Use the Storage Group application of ISMF, which interacts with SMS, to define the same drives and libraries into the specified SCDS and make them a part of the SMS configuration when that SCDS is activated.

For object tapes, other information supplied by the SETOAM statement of the CBROAM*xx* PARMLIB member can supplement or override the ISMF information assigned to the Object or Object Backup storage group.

Note: OAM identifies tape volumes eligible for reading and writing objects through information provided by the Tape Volume Table in the OAM configuration database and by the Object or Object Backup storage group to which the volume is assigned. Therefore, definition of tape libraries and tape drives to the OAM configuration database through the use of ISMF is not required for object tapes. However, tape libraries are defined to an SCDS and to the tape configuration database. Tape drives are dynamically allocated by the system as needed to satisfy requests to read or write objects.

Upon activation of an SCDS having optical libraries and optical disk drives defined, as well as tape drives available for allocation as needed, an operator on any console within an SMS complex can issue commands targeted for any library or drive within the configuration.

Related reading:

- For more information on how IBM tape drives are allocated to the configuration, see "Using dynamic allocation for tape drives" on page 57, and *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.
- For more information concerning ISMF usage in an optical environment, see Appendix A, "Sample optical hardware configurations," on page 385, and *z/OS DFSMS Using the Interactive Storage Management Facility*.

Installation storage management policy overview

Each installation defines a storage management policy that allows effective object storage management without requiring user intervention. Using ISMF, the storage administrator and system programmer define an installation storage management policy in an SMS configuration. OAM manages object storage according to the active policy. Disk (DB2 tables or file system), optical, and tape can all be used as the primary storage media for storing objects. Backup copies of objects can only be stored on optical or tape volumes. See Figure 3 on page 15 for a pictorial overview of this process.

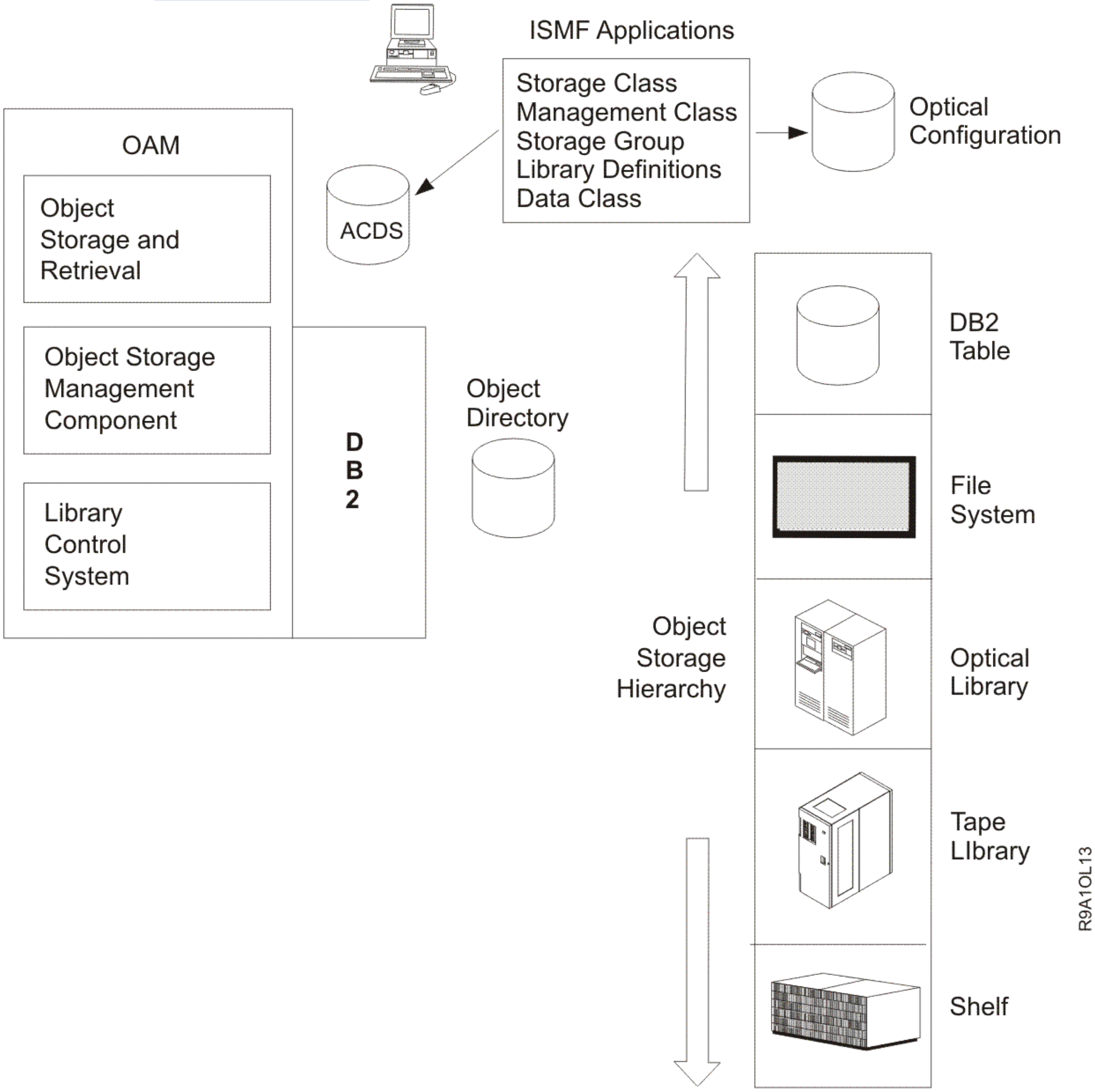


Figure 3. Overview of the Installation Storage Management Policy

An SMS configuration consists of the following elements:

- **Base configuration.** The base configuration identifies the systems in an SMS configuration and contains installation defaults. It also applies to SMS-managed data sets as well as objects. Only object-related functions are discussed in this document.
- **Active control data set (ACDS).** The ACDS controls the storage management policy for the installation.
- **Automatic class selection (ACS) routines.** The storage administrator uses the ACS routines to assign storage group, storage class, management class, and data class constructs to data sets or objects that are based on customer-defined criteria. ACS routines are invoked with user-input variables, and they make decisions based on the environment called. The ACS routines use input values to set new values which causes changes in the SMS handling of the data.
- **Optical library and optical drive definitions.** Optical storage hardware must be defined to the system through ISMF before it can be used. (Tape drives are dynamically allocated for use when required. They are defined to the system through the use of the Hardware Configuration Definition [HCD], not ISMF.)

- **OAM configuration.** OAM stores the optical library and drive definitions in the OAM configuration database (DB2) and in the ACDS through the SMS constructs.
- **SMS constructs.** Constructs are lists of attributes that are assigned to objects and storage areas. An SMS configuration can contain the following types of constructs. However, OAM uses only four of them (storage group, storage class, management class, and data class) to manage object storage. An SMS configuration can contain multiple constructs of each type.
  - The **Storage group** construct allows you to define a storage hierarchy and manage that hierarchy as if it were one large, single storage area. See "Understanding the storage group construct" on page 17 for information on establishing and manipulating storage groups.
  - The **Storage class** construct allows you to define different levels of performance objectives and availability requirements for objects. See "Understanding the storage class construct" on page 24 for information on assigning an object to a storage class.
  - The **Management class** construct allows you to define backup, retention, and class transition attributes for objects. See "Understanding the management class construct" on page 31 for specific information on defining the management class attributes.
  - The **Data class** construct allows you to define specific data attributes that are required for your installation's tape storage. See "Understanding the data class construct" on page 32.
  - **Aggregate group** allows you to group a collection of data objects to prepare for disaster recovery, application transfer or archiving, or data migration. An aggregate groups allows the data to be referred to collectively or individually. OAM *does not* use aggregate groups.
  - A **copy pool** is a defined set of storage groups that contain data that DFSMShsm can back up and recover collectively using fast replication. OAM *does not* use copy pools.

Related reading:

- For more information on using SMS, see *z/OS DFSMS Introduction* and the *z/OS DFSMSdfp Storage Administration*.
- For information on using ISMF, see the *z/OS DFSMS Using the Interactive Storage Management Facility*.

SMS constructs and ACS routines

This section discusses the SMS constructs used by OAM and their relationship with the automatic class selection (ACS) routines. OAM defines the management policy parameters in the SMS constructs of storage group, storage class, management class, and data class.

OAM associates these parameters with every object that it stores. The storage administrator defines the associations through ACS routines. The parameters include the following:

- Object retention rates
- The media on which OAM stores object collections
- Legal requirements for object retention
- Retrieval response time
- Location of object collections in the storage hierarchy
- How long OAM should hold the object collection at that level in the hierarchy
- Number of backup copies required (0, 1 or 2)
- Whether the first backup copy is written at the time the object is initially stored or during a subsequent storage management cycle
- The media type to which OAM should direct backup copies of objects
- Affiliation of libraries with relevant storage groups

Understanding the storage group construct

For data sets, the storage group construct simplifies the task of administering external data. By putting a number of homogeneous data sets into one storage group, they can be viewed as one entity.

An Object storage group is composed of a set of volumes. Each installation develops Object storage groups according to its individual needs. Storage groups can be used to segregate different types of data (such as production versus development) and aggregate like types of data.

By separating the physical volumes from the service level (as defined by the storage class construct), Object storage groups also allow installations to change the physical aspects of storage without affecting the logical requirements of data access. For example, a new volume or device can be added to the storage group without affecting end-user routines.

In an OAM environment, Object storage groups allow the storage administrator to define an object storage hierarchy. The *object storage hierarchy* classifies storage areas according to location and, therefore, according to retrieval response time. Each object storage hierarchy must contain an *object directory*, containing control information about each object. Additionally, the hierarchy can have:

- DB2 object storage tables on disk
- A file system (NFS or z/FS)
- Optical volumes that are associated with optical libraries (real or pseudo), and operator-accessible optical disk drives
- Tape volumes that are associated with tape libraries or stand-alone tape drives

During an object's lifetime, it can move from one OAM storage hierarchy level (storage location) to another, ascending or descending depending on its performance objectives.

Related reading: For more information on storage groups for data sets and objects, see *z/OS DFSMSdfp Storage Administration*.

Using Object, Object Backup, and Tape storage groups

In addition to the storage groups that are defined by each installation for its data sets, OAM uses three special storage group types: Object, Object Backup, and Tape. OAM uses these storage groups as follows:

- An Object storage group contains primary objects. See "Assigning Object Storage Groups" on page 18 for more information on assigning storage groups.
- An Object Backup storage group contains the first or the second backup copy of each object for which the management class construct requires a backup. See "Assigning Object Backup storage groups" on page 19 for more information on assigning Object Backup storage groups.
- A Tape storage group contains tape volumes that are associated with an automated tape library dataserver (ATLDS) or a manual tape library (MTL). See "Assigning Tape Storage Groups" on page 18 for more information on assigning storage groups.

A *primary* object is the primary copy of an object in the object storage hierarchy, which is stored in the Object storage group on a disk sublevel (DB2 or file system), optical, or tape. A *backup* object is the first backup copy of an object, which is stored in the Object Backup storage group specified as a first backup storage group. A *second backup* object is the second backup copy of an object, which is stored in the Object Backup storage group specified as a second backup storage group.

You can retrieve the primary, backup, or second backup copy of an object by using the OSREQ RETRIEVE command. Specify VIEW(PRIMARY), VIEW(BACKUP), or VIEW(BACKUP2) on the RETRIEVE request. You can also get automatic access to the backup copies for retrieval by using the Automatic Access to Backup facility.

When a primary or backup copy of an object residing on optical or tape media is retrieved, OAM can also create a temporary copy of the object in disk sublevel 1 or disk sublevel 2 for a user-defined number of days. This can significantly improve the performance rate for subsequent retrieves of this object. This object recall to disk process can be initiated explicitly using a RECALL keyword on the OSREQ RETRIEVE request, or implicitly using defaults defined through SETOSMC statements in the CBROAMxx Parmlib member. Refer to "Recalling objects to disk" on page 209 for details.

Related reading: See *z/OS DFSMS OAM Application Programmer's Reference* for more detailed information on the OSREQ macro.

Defining an Object or Object Backup Storage Group

To define an Object or Object Backup storage group, use the ISMF Storage Group application. Use SETOAM statements in the CBROAM*xx* PARMLIB member to specify the tape-related options that can supplement or override these ISMF specifications for the Object or Object Backup storage group definitions.

If a tape unit name is associated with an Object or Object Backup storage group on the SETOAM statement in the CBROAM*xx* PARMLIB member, the backup copies are written to tape volumes. In this instance, any optical libraries that are associated with the Object Backup storage group that is defined using the ISMF storage group define panel are ignored for writing backup copies of objects to that Object Backup storage group. If the SETOAM statement does not direct the Object Backup storage group to tape media, OAM writes the backup copies to optical media. Additionally, if there is no SETOAM statement in the CBROAM*xx* PARMLIB member, then OAM automatically writes the backup copies to optical media.

Other information that is supplied by the CBROAM*xx* PARMLIB member can supplement or override the ISMF information that is assigned to the Object or Object Backup storage group.

If an Object or Object Backup storage group belongs to an OAM that is a member of an OAMplex, it can be connected to more than one system in an SMS complex. The libraries that are defined for these storage groups can also be connected to multiple systems within the OAMplex. If the OAM is not part of an OAMplex, each Object or Object Backup storage group can be connected to only one system in the SMS complex.

For information on changing the SETOAM values dynamically and on defining the values when the CBROAM*xx* PARMLIB member is not used at initialization, see "Using the UPDATE command to set SETOAM, SETOSMC, and SETOPT values" on page 366.

For more information on assigning backup copies of objects to various media types, see "Determining which media to use for backup copies" on page 19.

Assigning Tape Storage Groups

A Tape storage group can be assigned to an Multiple Virtual Storage (MVS™) scratch tape when it is first used to store an OAM object. The Tape storage group is assigned to the tape volume through the ACS routines at tape volume allocation.

A single tape volume can be associated with a Tape storage group and an Object storage group, or a Tape storage group and an Object Backup storage group. When space is needed to write a backup copy of an object, a tape volume is assigned to an Object Backup storage group. As a result, a single tape volume which is originally allocated inside of an ATLDS or MTL can be associated with both a Tape storage group and an Object Backup storage group.

Assigning Object Storage Groups

An Object storage group is associated with an Object Backup storage group through SETOSMC statements in the CBROAM*xx* member of PARMLIB. Through these statements, you can associate an Object storage group with a first and a second Object Backup storage group. If no storage groups are specified, then the defaults for the configuration are used.

Note: You can access only volumes that are associated with the Object or Object Backup storage groups that are defined in the active SMS configuration. A volume that is associated with an Object or Object Backup storage group that is not defined in the active SMS configuration cannot be accessed. Objects that are already written on that volume cannot be retrieved, and OAM cannot write new objects to that volume. A message is issued at OAM initialization for each volume that is associated with the Object or Object Backup storage group that is not defined in the active configuration. (Message CBR0182I is issued for optical volumes or CBR0210I for tape volumes.) To remedy this problem, define the Object or Object

Backup storage groups to the active configuration by activating an SCDS that contains the Object or Object Backup storage groups.

See "SETOSMC statements for use in the OSMC environment" on page 138 for a sample SETOSMC statement that you can use when assigning Object storage groups.

Assigning Object Backup storage groups

You can direct OAM to create a first and a second backup copy of objects using the existing NUMBER OF BACKUP VERSIONS (DATA SET EXISTS) field that is located in the SMS management class definition. OAM uses this field to determine how many backup copies of an object are to be made when OSMC processing is done for an object's storage group, if there are SETOSMC statements in the CBROAM*xx* member of PARMLIB indicating that second backup copies are to be created. See page "Defining management classes" on page 175 for the specifics of defining this field.

An Object Backup storage group can serve as either a first backup storage group, or as a second backup storage group, but not both. OAM fails initialization if SETOSMC specifies the same Object Backup storage group name as both a first backup storage group and a second backup storage group.

Assigning Object Backup storage groups differs from assigning the Object storage group in that you must use SETOSMC statements to indicate that you want the second backup copy stored in an Object Backup storage group that is not the default Object Backup storage group. If you do not provide SETOSMC statements, then OAM does not process second backup copies of objects.

See "Displaying SETDISK, SETOAM, SETOPT, SETOSMC, and SETTLIB parameters" on page 353 for the specifics of assigning Object Backup storage groups using SETOSMC statements.

Determining which media to use for backup copies

OSMC uses the AUTO BACKUP parameter on the management class to determine if backup copies of an object should be written. OSMC schedules writes of **two** backup copies if all of the following items are true:

- The AUTO BACKUP parameter equals Y.
- The number of backup versions that is specified in the management class field, NUMBER OF BACKUP VERSIONS (DATA SET EXISTS), is greater than or equal to two (≥2).
- A SECONDBACKUPGROUP keyword is specified in a SETOSMC statement in the CBROAM*xx* PARMLIB member.

The backup copies of the object are written to the Object Backup storage groups assigned to the Object storage group to which the object belongs. Using the SETOAM statements, you can specify that the backup copies of the object be written to the same removable media type or to different removable media types, or both. The media that is selected for the backup copies might be optical or tape. If OAM is initialized with a CBROAM*xx* PARMLIB member containing SETOAM statements for the Object Backup storage groups, and the SETOAM statements include valid TAPEUNITNAME specifications, the backup copies are written onto tape media. If the TAPEUNITNAME is not valid, OAM initialization fails. If no valid SETOAM statements exist for a given Object Backup storage group, all backup copies written to that Object Backup storage group are written to optical media.

Note: If the Management Class associated with the object has AUTO BACKUP = Y and BACKUP FREQUENCY = 0, then the first backup copy is created at the time the object is initially stored.

OSMC schedules a write of only **one** backup copy if all of the following items are true:

- The AUTO BACKUP parameter equals Y.
- The NUMBER OF BACKUP VERSIONS (DATA SET EXISTS) parameter is less than two.

OAM writes the single backup copy to the designated Object Backup storage group onto the media type that is assigned for that storage group.

Tip: The default number of backup versions in the NUMBER OF BACKUP VERSIONS (DATA SET EXISTS) field is two.

Note: If the Management Class associated with the object has AUTO BACKUP = Y and BACKUP FREQUENCY = 1, then the first backup copy is deferred to the first OSMC Storage Group cycle after the object was initially stored.

First and second backup copies of objects can never reside on the same tape volume or optical disk volume. The primary copy is stored on a volume belonging to the Object storage group. The first backup resides on a separate volume belonging to the Object Backup storage group designated as the first object backup storage group, and the second backup resides on a separate volume belonging to the Object Backup storage group designated as the second object backup storage group. A single tape volume or optical disk cannot belong to more than one Object or Object Backup storage group. The first and second backup copies of objects belong to separate Object Backup storage groups, therefore cannot reside on the same volume.

The optical disk volume that is chosen to store the primary copy or first or second backup copies of the object is an optical volume (primary copy) or backup optical volume (backup copy) that is contained in one of the real or pseudo optical disk libraries that are listed in the Object or Object Backup storage group definitions.

Allocating a scratch tape for the tape storage group

If a scratch tape volume is being allocated to store the primary or backup copy of the object, MVS scratch tape allocation chooses a library-resident tape volume (from an ATLDS or MTL) that is associated with the Tape storage group. If a scratch tape volume is not needed, an existing tape volume that is associated with an Object or Object Backup storage group that is defined in the current SCDS and capable of being mounted and handling the write request is allocated.

Using the DATACLASS parameter to determine compaction

If the storage class indicates tape media and the Tape storage group that specifies the tape volume allocation is for a tape library dataserver, the allocation process uses the DATACLASS parameter on the SETOAM statement to determine tape compaction or no tape compaction for the volume. If the TAPECOMPACTION or NOTAPECOMPACTION keywords on the SETOAM statement are specified for a particular storage group, the data is written in compacted or noncompacted format as specified. If these keywords are not specified at storage group level of the SETOAM statement, the DATACLASS parameter of the SETOAM statement at the global level is used to determine tape compaction or no tape compaction. Should the DATACLASS parameter not be specified at the OAM global level, information that is passed on the DEVSUP parameter after allocation processing is used to determine whether the allocated tape volume should have tape compaction or no tape compaction.

See “Media selection for object storage” on page 24 for more information.

Using the TAPEUNITNAME parameter for volume allocation to a stand-alone drive

If the volume allocation is done using a stand-alone tape drive and no OAM scratch tape is available, OAM allocates a scratch tape using the TAPEUNITNAME parameter that is specified in the subparameter of the STORAGEGROUP parameter on the SETOAM statement. If no TAPEUNITNAME parameter is associated with the Object storage group that is assigned to the object, the storage of the object fails. The TAPEUNITNAME is stored in the UNITNAME column of the TAPEVOL table. The TAPEVOL table contains a single row for each tape volume containing OAM objects.

Requirement: The TAPEUNITNAME parameter is required and is specified on all dynamic allocations so that the device which is allocated is compatible with the tape to be mounted.

Object tape data set low-level qualifier

To assist the tape management system in mounting an appropriate media type in the stand-alone environment (non system-managed tape environment), a global keyword DSNWITHSGNAME can be specified in the SETOAM statement in the CBROAMxx PARMLIB member. When specified, the object storage group name is appended to the OAM primary and backup tape data set names. The tape management system can parse the data set low-level qualifier (storage group) to determine from which pool a scratch volume should be selected for a mount request in the stand-alone environment. Refer to “SETOAM keyword definitions for global level” on page 112 for more information.

Using SETOAM to direct objects to a specific device type for an object storage group

You can use a SETOAM statement to direct all objects for an Object or Object Backup storage group to a specific tape type by specifying the TAPEUNITNAME(*device-type*) parameter that is associated with the storage group.

The following are examples of directing objects to specific tape device types using the TAPEUNITNAME parameter of the SETOAM statement:

- If 3490 is the TAPEUNITNAME parameter, then objects written to the subject Object or Object Backup storage group would all have this unit name recorded in the tape volume table. Objects written to the Object or Object Backup storage group would all be written to 3490 devices.
- If the TAPEUNITNAME(3490) is changed to a new device type, such as TAPEUNITNAME(3590), OAM continues to write to the available usable tape volumes that are associated with the Object or Object Backup storage group using 3490 devices until an MVS scratch allocation is required. When an MVS scratch tape is required to handle an out-of-space condition for the Object or Object Backup storage group, that scratch tape is written on a device that is specified by the SETOAM TAPEUNITNAME statement for that group, which is 3590 in this example.

  OAM continues to read the objects that were previously written on 3490 devices for the Object or Object Backup storage group so long as there is a 3490 device available for allocation at the time the read is requested.

  Note: If a tape volume that was previously written on a 3490 device is entered into an ATLDS or MTL, that tape volume continues to be used as long as the ATLDS or MTL to which it was entered contains a compatible device.

See “Using dynamic allocation for tape drives” on page 57 for information concerning compatible devices not being available to handle requests.

Grouping devices (esoteric unit names)

Devices can be grouped together and defined as one group to the system. For instance, a group of 3590 tape drives in the same room can be grouped together and defined as 3590GRP. These device groups are known as esoteric unit names or *esoterics*. If you specify an esoteric on the SETOAM command for a group, you must ensure that this esoteric exists if an OAM tape has that esoteric specified in the UNITNAME field of the TAPEVOL table. Do not change the contents of that esoteric to introduce incompatible device types. Should the esoteric name be deleted or changed, the volumes that are associated with the esoteric name cannot be allocated. Because the TAPEUNITNAME cannot be resolved, the tape volume that is required for the request is not mounted and the allocation request fails.

Using DB2 with object storage groups

Each Object storage group consists of a DB2 database. The DB2 database is referred to as the *object storage database*. The object storage database contains the following DB2 tables:

- An object directory containing entries for objects residing in a particular Object storage group. These entries contain control information needed to locate and manage the object.
- A storage table for objects less than or equal to 3980 bytes.
- A storage table for objects greater than 3980 bytes and less than or equal to 256 MB (268,435,456 bytes).
- A LOB storage table (LOB storage structure) for objects greater than 32,640 bytes. The LOB storage structure is optional for objects greater than 32,640 bytes and less than or equal to 256 MB. The LOB storage structure is required for objects greater than 256 MB and less than or equal to 2000 MB (2,097,152,000 bytes).

For more information on the structure of the object storage tables see “Object storage tables” on page 488.

Conceptual overview of storing a primary object to DB2 sublevel

A primary object to be stored to the DB2 sublevel of the OAM storage hierarchy can be inserted into the 4 K object table, the 32 K object table, or the LOB Storage Structure. Figure 4 on page 23 shows how OAM uses the object size, the LOB keyword in the IEFSSN*xx* PARMLIB member, and the SYSIBM.SYSTABLES table to determine which table should be used to store the object.

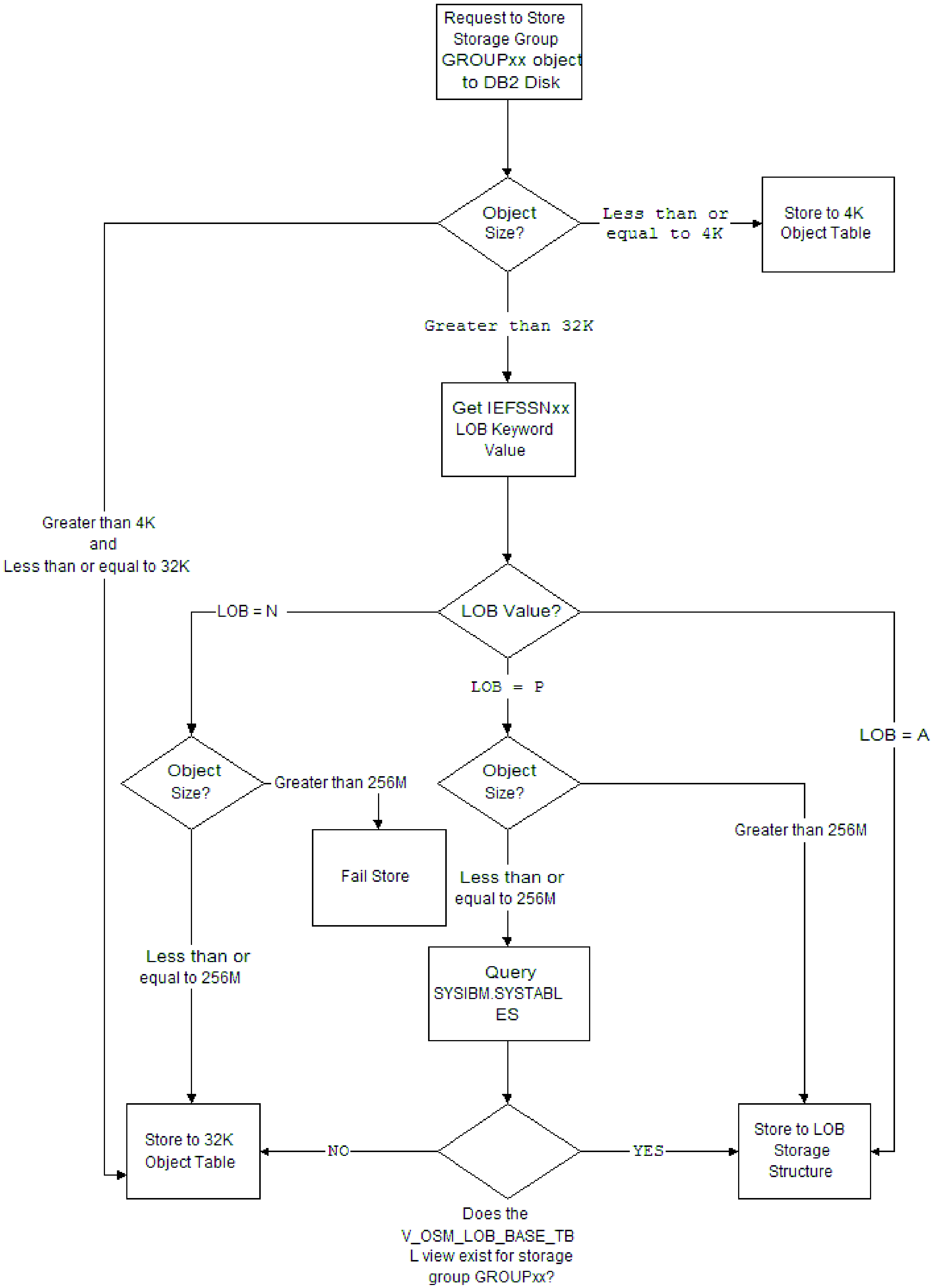


Figure 4. DB2 Storage Decision Tree

Understanding the storage class construct

The storage class construct enables storage administrators to separate the logical requirements for accessing data from the physical requirements for storing data. Storage class represents the level of service (performance objectives and availability requirements) for an object.

Every object is assigned to a storage class when it is created; therefore, every object is SMS-managed. This assignment determines where the object initially resides in the object storage hierarchy (DB2 sublevel, file system sublevel, optical, tape sublevel 1, and tape sublevel 2). See Figure 8 on page 34 for a diagram of this process. The assignment can change as part of a class transition or as the result of an explicit application request (OSREQ CHANGE).

Related reading:

- For more information on changing defaults, see “Modifying default storage and management classes” on page 199.
- For more information on using the OSREQ macro to customize your application interface, see *z/OS DFSMS OAM Application Programmer's Reference*.

Media selection for object storage

Each object that OAM stores is assigned a storage class and a management class. OAM uses the storage class to determine the initial placement of an object in the OAM object storage hierarchy. OAM also uses the storage class during the OSMC storage management cycle to determine the correct placement of the object when the storage management cycle processes that object. OAM uses the Initial Access Response (IARS) parameter in the storage class to determine if a primary copy of an object is stored on disk (DB2 tables or a file system) or on removable media (optical or tape). If the IARS parameter in the storage class that is assigned to the object is zero, the primary copy of the object is stored on disk. If the IARS parameter is nonzero, the primary object is stored on removable media.

If the IARS parameter specifies that removable media is to be used, the Sustained Data Rate (SDR) parameter of the storage class determines which removable media, optical or tape, is used to accept the primary copy of the object. If the SDR parameter of the storage class is *greater than* or *equal to* three (≥3), the primary copy of the object is stored on a tape volume. If the SDR parameter of the storage class is *less than* three (<3), the primary copy of the object is stored on an optical disk volume.

If the IARS and SDR parameters, taken together, specify either disk or tape media, the OAM Sublevel (OSL) parameter of the storage class determines which disk or tape sublevel is to be used to accept the primary copy of the object. If it has been determined that the primary copy of the object is to be stored on disk, then if the OSL parameter of the storage class equals 1, the primary copy of the object is stored on disk sublevel 1 (DB2 tables), whereas if the OSL parameter of the storage class equals 2, the primary copy of the object is stored on disk sublevel 2 (file system).

Similarly, if it has been determined that the primary copy of the object is to be stored on tape, then if the OSL parameter of the storage class equals 1, the primary copy of the object is stored on a tape sublevel 1 volume. If the OSL parameter of the storage class equals 2, the primary copy of the object is stored on a tape sublevel 2 volume.

Table 1 on page 24 provides a summary of how the IARS, SDR, and OSL values determine the initial placement of an object in the OAM object storage hierarchy.

Table 1. Summary of media selection for object storage

IARS value	SDR value	OSL value	Where object is stored
0	N/A	1	disk sublevel 1 (DB2 tables)
0	N/A	2	disk sublevel 2 (file system)
non-zero	<3	N/A	optical disk

Table 1. Summary of media selection for object storage (continued)			
IARS value	**SDR value**	**OSL value**	**Where object is stored**
non-zero	≥3	1	tape sublevel 1
non-zero	≥3	2	tape sublevel 2

Conceptual overview of storing a primary object

Figure 5 on page 26 shows how OAM uses the IARS, the SDR, and the OSL parameters to determine where to store a primary object.

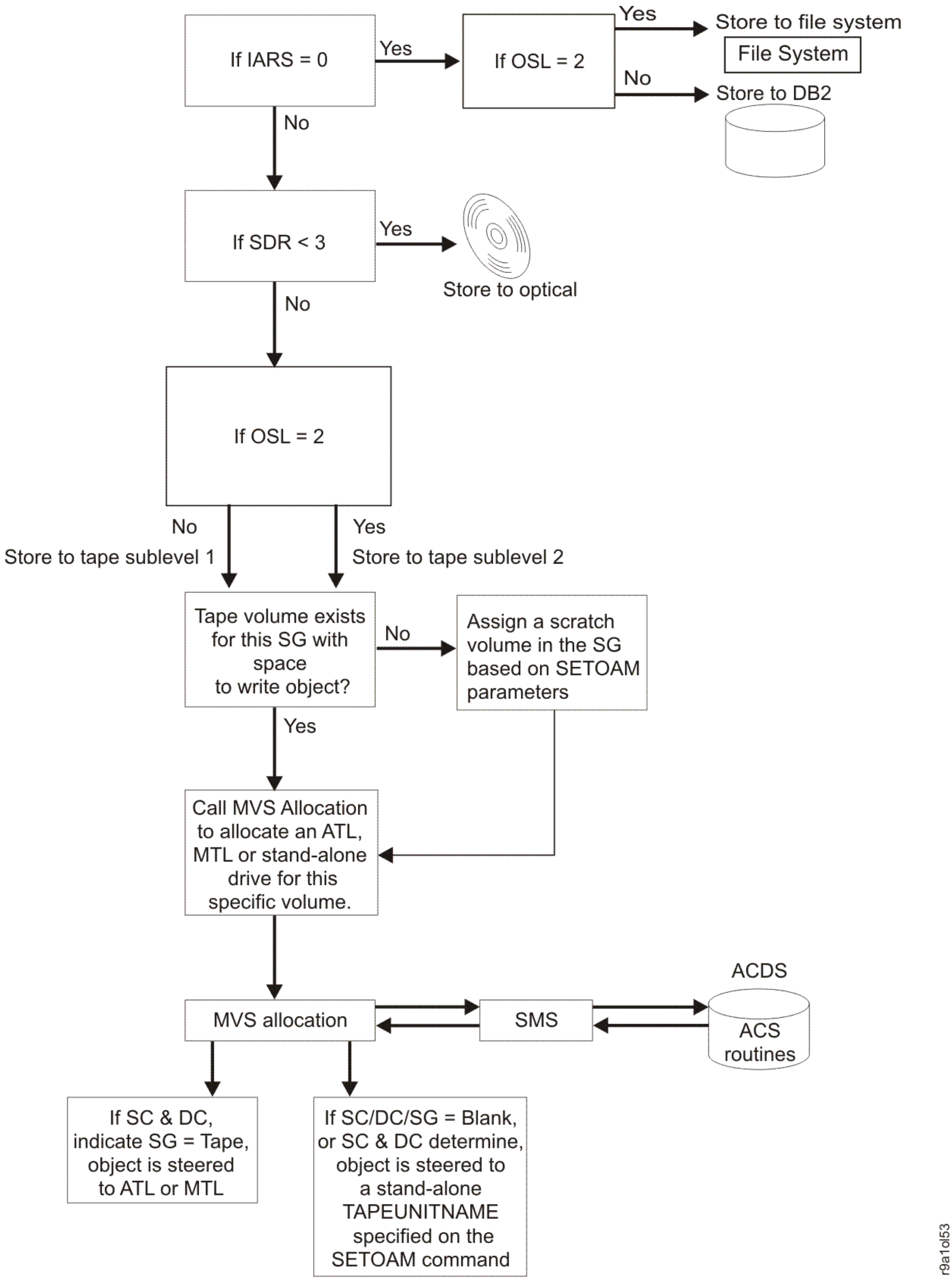


Figure 5. Conceptual Overview of Storing a Primary Object

Conceptual overview of storing a first backup copy of an object

Figure 6 on page 28 shows how OAM uses the AUTOBACKUP and the FIRSTBACKUPGROUP parameters to determine where to store a first backup copy of an object.

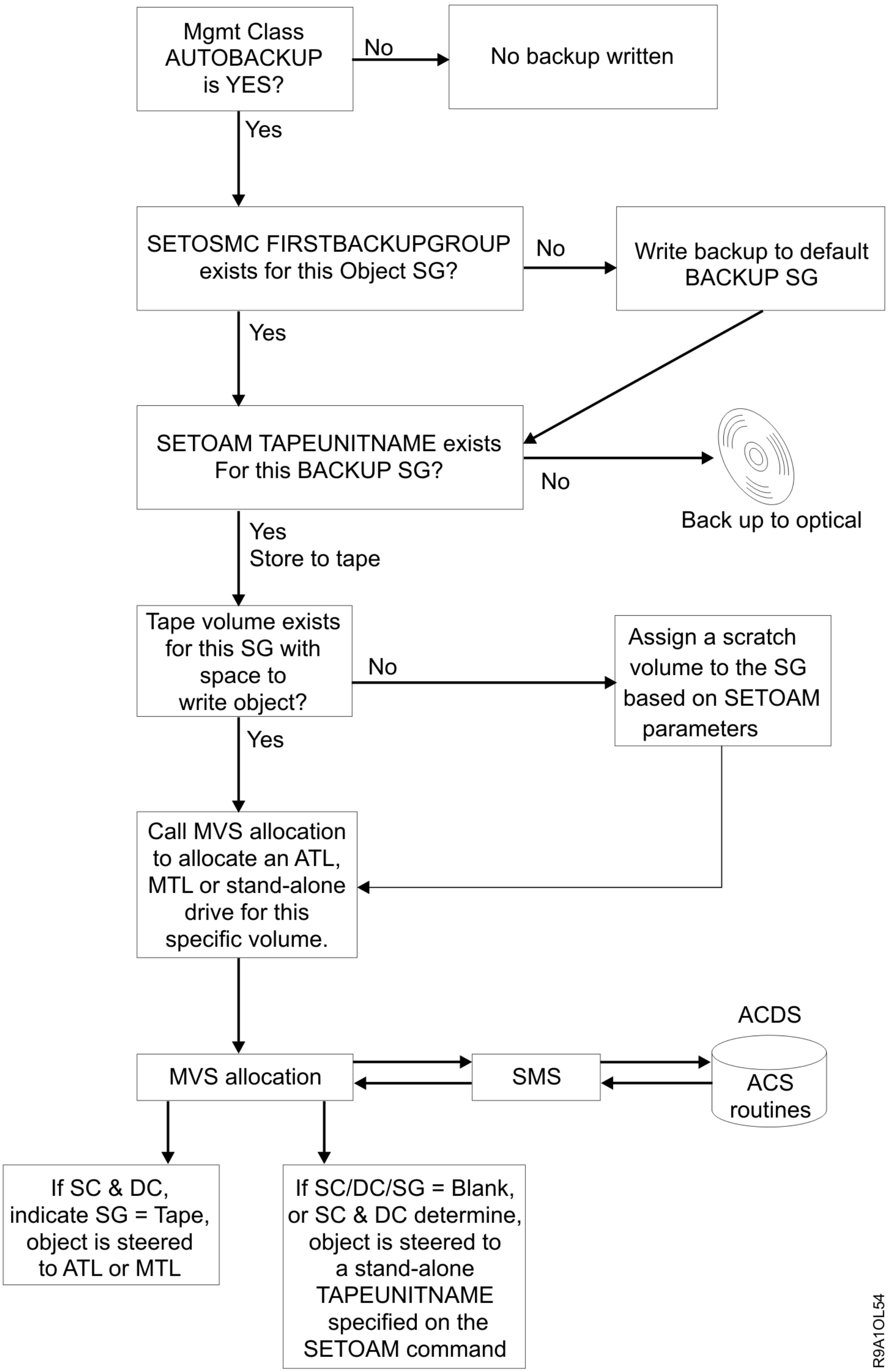


Figure 6. Conceptual Overview of Storing a First Backup Copy of an Object

Conceptual Overview of Storing a Second Backup Copy of an Object

Figure 7 on page 30 shows how OAM uses the AUTOBACKUP and the SECONDBACKUPGROUP parameters to determine where to store a second backup copy of an object.

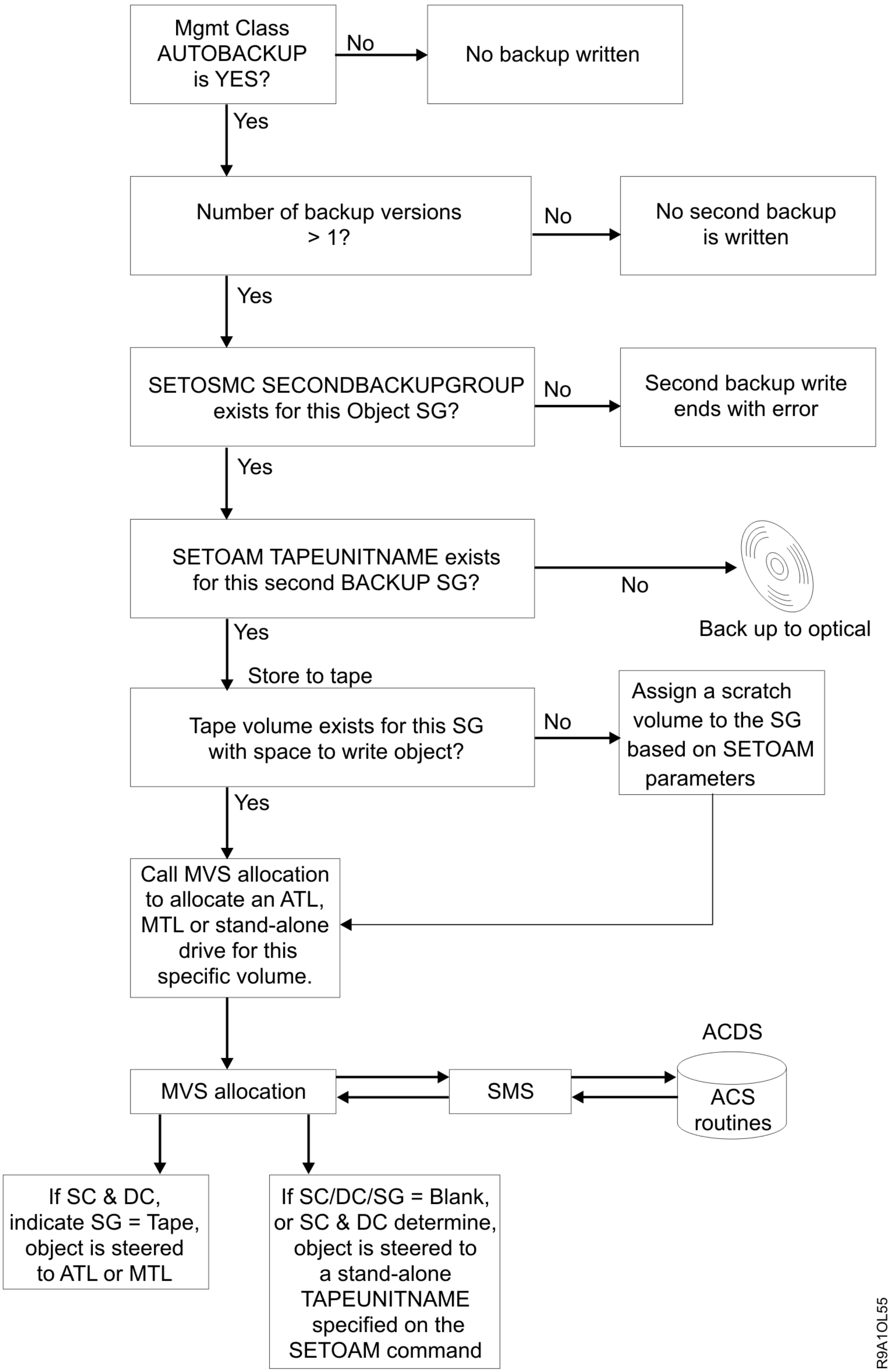


Figure 7. Conceptual Overview of Storing a Second Backup Copy of an Object

Object storage on the shelf

In addition to an object being stored inside a library (library-resident), an object can be assigned to a "shelf" (shelf-resident) location within the storage hierarchy. The following concepts apply to shelf storage:

- A shelf-resident volume can be either an optical or tape cartridge that is associated with stand-alone tape drives or operator-accessible optical drives within a pseudo library. The optical volume is physically stored on a shelf location near the drives that are associated with the pseudo-library to which the volume is assigned. The tape volume is physically stored on a shelf location near the drives that are associated with the TAPEUNITNAME parameter to which the volume is assigned.
- Assigning an object to storage class `shelf` (through an OSREQ STORE/CHANGE request or a class transition) does not cause the object to be physically moved to another volume. (The storage class ID row in the object directory table is the only change that is made.) Nor does the volume, on which the object resides, automatically get ejected from the library to which it is associated, even if all the objects on the volume indicate the storage class `shelf`. There is no storage class definition for shelf storage (no specific IARS value specific to shelf). This storage class assignment allows an installation a way of differentiating between performance objectives for objects that are actively accessed and those that must be archived, or those that are accessed the least. According to their storage management policy, the installation determines whether these objects having a storage class of **shelf** should be removed from the library and placed physically on a shelf location for storage.

Understanding the management class construct

The SMS management class is a list of class transition, backup, and retention attributes. OAM uses management class attributes to manage objects. Every object is assigned a management class when it is created. See Table 2 on page 32 for management class examples.

Class transition attributes allow OAM to change the way an object is managed based on its age, its usage, or a predefined, periodic calendar event (for example, the 30th day of every month or the first day of the quarter). You cannot use the PERIODIC and TIME SINCE CREATE transition attributes together. Class transitions occur when the OSMC storage management cycle is invoked. For objects requiring class transition, OSMC uses the ACS routines to determine if the objects should be managed using a different management class or if they should be placed at a different level of the storage hierarchy according to a different storage class.

OAM uses management class attributes to decide when to write the first backup copy and whether to write one or two backup copies of an object. AUTO BACKUP with BACKUP FREQUENCY determines when a backup copy is written.

- You can schedule a backup copy to be written immediately after the object is stored.
- You can make backup copies during the first storage management cycle after the object is stored, or during the first storage management cycle after a new management class is assigned for the object.

AUTO BACKUP and NUMBER OF BACKUP VERSIONS (DATA SET EXISTS) determine whether more than one backup copy is to be written.

Retention period and expiration attributes of the management class definition determine the OAM action for object expiration. An object can expire automatically based on its age, its usage, or a specific date (derived from its management class or a management-class-approved object-specific retention period, if provided). OSMC uses the auto-delete installation exit to automatically delete expired objects during the storage management cycle. Deletion-hold is used to prevent object deletion while object is in deletion-hold mode. Retention-protection is used to prevent object deletion prior to object's expiration date, and will not allow expiration date to be changed to an earlier date. Deletion-protection is used to prevent object deletion prior to object's expiration date.

Related reading:

- For information on changing management class defaults, see "Modifying default storage and management classes" on page 199.

- For information on how OAM determines whether to write one or two backup copies, see "Determining which media to use for backup copies" on page 19.
- For more information about deleting expired objects, see Appendix F, "Auto-delete installation exit," on page 573.

Table 2 on page 32 shows examples for specifying various management class attributes for objects.

Table 2. Management class examples for objects

Management Class Names	Time Since Creation (retention attribute)	Time Since Last Use (transition attribute)	Periodic (transition attribute)	AUTOBACKUP (backup attribute)	Number Of Backup Versions (Data Set Exists) (backup attribute)
MAGONLY	30 DAYS	N/A	N/A	NO	0
FOREVER	NOLIMIT	NOLIMIT	N/A	YES	2
MAGS	N/A	5 DAYS	N/A	NO	0
MAG30D	30 DAYS	N/A	N/A	YES	1
MAG30LIB	6 MONTHS	N/A	N/A	NO	0
MAG30SHF	7 YEARS	N/A	N/A	NO	0
OPT6LIBF	6 MONTHS	N/A	N/A	NO	0
OPT6SLH	7 YEARS	N/A	N/A	NO	0
TAPE30	30 DAYS	N/A	N/A	NO	0
TAPE30B	30 DAYS	N/A	N/A	YES	2
MLAST	N/A	N/A	MONTHLY on 30	NO	0
MFIRST	N/A	N/A	QUARTLY on FIRST	YES	2
Note: N/A = not applicable					

Understanding the data class construct

Data class is an SMS construct that determines the characteristics for a tape volume during scratch tape allocation. Data class determines the following attributes for a tape volume allocated in an ATLDS or MTL:

- Tape expiration date
- Retention period
- Recording technology
- Performance scaling
- Performance segmentation
- Tape device selection information (TDSI):
  - Compaction (YES | NO | BLANK)
  - Media type (BLANK | MEDIA1 | MEDIA2 | MEDIA3 | MEDIA4 | MEDIA5 | MEDIA6 | MEDIA7 | MEDIA8 | MEDIA9 | MEDIA10 | MEDIA11 | MEDIA12 | MEDIA13)
  - Recording technology (BLANK | 18-TRACK | 36-TRACK | 128-TRACK | 256-TRACK | 384-TRACK | EFMT1 | EFMT2 | EEFMT2 | EFMT3 | EEFMT3 | EFMT4 | EEFMT4)
  - Special attribute (NONE | READCOMPATIBLE)
- Key Labels
- Encoding Mechanisms

Data class determines the following attributes for a tape volume allocated to a stand-alone tape drive:

- Tape expiration date

- Retention period
- Performance scaling
- Performance segmentation
- Compaction
- Key Labels
- Encoding Mechanisms
- Recording technology (for tape drives that record in more than one format, for example, the 3592 Model E05 or Model E06)

Determining data class during scratch tape allocation

The data class of a volume is determined when a scratch tape volume is allocated. If the allocation is steered to an ATLDS or a MTL, the data class subparameter on the SETOAM statement of the CBROAM*xx* PARMLIB member for the Object or Object Backup storage group is used, if it is specified.

If the SETOAM statement does not specify the data class subparameter at the storage group level, use the DATACLASS parameter of the SETOAM statement for the OAM global level to specify the values for the tape volume. The DATACLASS parameter of the SETOAM statement at the global level applies to all tape volumes that belong to storage groups not having their own DATACLASS assigned. If you do not specify DATACLASS at the storage group or at the OAM global level, OAM uses the DEVSUP parameter default to determine tape compaction or no tape compaction for the tape volume. The ACS routines can also supplement or override the data class values that you specify for the tape volume from either the SETOAM statement or the DEVSUP parameter.

Recommendation for data class and ACS routines: Do not allow the ACS routines to assign or change the data class assignment of an OAM tape volume. Ensure that the ACS routines do not change DATACLASS specifications for OAM object tape data sets, including OAM.PRIMARY.DATA.*, OAM.BACKUP.DATA.*, or OAM.BACKUP2.DATA.* for any storage group. The data class for OAM tape volumes is determined by the SETOAM statement of the CBROAM*xx* PARMLIB member at MVS scratch tape allocation. The SETOAM statement provides this information either at the Object or Object Backup storage group level or at the OAM global level—whichever best suits the requirements for the tape volume that is being allocated. Allowing the ACS routines to alter this specification could create unexpected consequences (for example, no compaction of the data when the SETOAM statement specifies compaction). When you write the ACS routines, ensure that the data class construct for OAM tape volumes is kept intact.

See Figure 5 on page 26 for a diagram of the process of storing objects to tape.

Ignoring TDSI data class information for a stand-alone allocation

If the allocation is for a stand-alone tape drive, the information that you specify on the TAPEUNITNAME subparameter of the STORAGEGROUP parameter of the SETOAM statement determines the specific device type that is allocated for the tape volume. Tape volumes that are allocated to stand-alone tape drives are not SMS-managed (the objects on the tape volumes are SMS-managed, but the tape volumes are not). Therefore, the TDSI information (media type and recording technology) of the data class is not necessary for non-SMS-managed volumes. However, with the introduction of 3592 tape drives that read and write in multiple recording formats, in order to request a lower recording technology or to request data encryption, you need to specify the recording format in data class to ensure that the desired recording format is used during OPEN processing, otherwise the drive defaults to its highest non-encryption recording format.

Related reading: For more information concerning these data class attributes, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

ACS routines

Automatic class selection (ACS) routines determine, validate, or override the existing values that are assigned for the storage group, storage class, management class, and data class constructs for a collection. The classes assigned are those assigned to the collection or those that are explicitly stated on the OSREQ request to store or change an object. ACS routines are called to verify the storage class and management class that are stated on the OSREQ request. The ACS routines can accept the stated class, select a different one, or reject the stated class and return an error code.

Every object belongs to a collection. Each collection belongs to only one Object storage group. When an object is stored, it is automatically assigned to the Object storage group to which its specified collection belongs. Every object, when it is stored, is assigned a storage class and a management class. See Figure 8 on page 34 for a diagram of this process.

Note: Storage and management classes are optional on the OSREQ STORE request. If they are not specified, they are assigned the defaults from the collection. For a new collection, the ACS routines supply defaults for them.

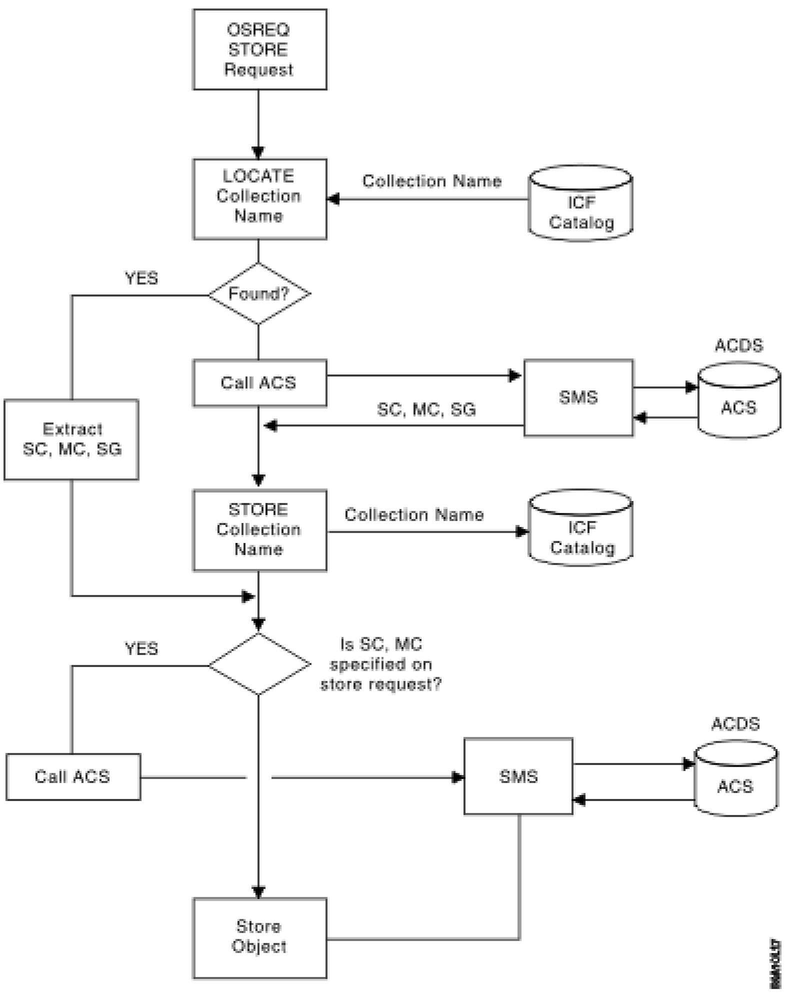


Figure 8. Storing an Object on Optical Media through an OSREQ STORE Request

Recommendation: For objects that reside on tape volumes, having the ACS routines alter the data class that was originally assigned to the tape volume through the SETOAM statement at MVS scratch tape allocation is not recommended. For more information, see the recommendation regarding data class and ACS routines.

OAM address space

An *OAM address space* is used to perform a subset of OAM processing. Start an OAM address space if you plan to use the file system sublevel, optical devices, or tape devices for storing objects and OSMC functions or if you plan to delete objects within your data storage environment.

Restrictions for OAM address spaces in a classic OAM configuration

- Only one OAM address space can be active in the z/OS system.
- Only one DB2 subsystem can be associated with the active OAM address space.
- Optical devices can only be directly accessed by the owning OAM address space. However, requests for the optical devices can be shipped to the OAM in an OAMplex that owns the optical devices by using the XCF messaging service.
- Specifying a region size other than 0 MB on the OAM-started procedure JCL can result in storage shortage abends, especially during an OSMC cycle.

These restrictions apply regardless of whether the address space is part of an OAMplex.

Restrictions for OAM address spaces in a multiple OAM configuration

- Only one tape library OAM address space can be active in the z/OS system. For more information on a tape library address space, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.
- Up to two object OAM address spaces can be active in the z/OS system.
- Each object OAM address space must be associated with a unique DB2 subsystem.
- A unique PROCLIB member with a unique name (OAMA, OAMB) is recommended for each OAM instance. A single member of PROCLIB can be used and started with a unique task identifier for each address space (S OAM.OAMA and S OAM.OAMB), but a unique value for the D= keyword is required for each address space. This approach requires specifying the D= value on the START command.
- The optical level of the OAM storage hierarchy is not supported.
- Specifying a region size other than 0 MB on the OAM-started procedure JCL can result in storage shortage abends, especially during an OSMC cycle.

These restrictions apply regardless of whether the address space is part of an OAMplex.

MEMLIMIT for OAM

To process objects greater than 256MB in the file system sublevel or on tape, the OAM address space uses 64–bit addressing and virtual storage above the 2G bar. Additional configuration may be required to specify a MEMLIMIT value that will provide virtual storage above the 2G bar to the OAM address space. Typically specifying REGION=0M on the OAM started procedure JCL implies a MEMLIMIT value that provides the OAM address space with the needed virtual storage above the 2G bar. This REGION=0M specification is recommended to avoid issues with virtual storage shortages and is the default in the sample OAM procedure. See "5h Updating the PROCLIB" on page 146.

The MEMLIMIT value can be explicitly set or implicitly provided in a number of different ways and can be overridden by an installation exit, so you must ensure the resulting MEMLIMIT specification is sufficient for OAM. For more information on MEMLIMIT, see *z/OS Migration* and *z/OS MVS Programming: Extended Addressability Guide*.

OAM usage of virtual storage above the 2G bar will result in significantly increased storage utilization on your system. For more information see "Auxiliary storage and real storage considerations" on page 69 and "System paging" on page 72.

OTIS address space

OAM uses the OAM thread isolation support (OTIS) for DB2 processing related to OAM collections and to remove OAM subsystems from the OAM configuration. Creating a new collection, for example, requires a DB2 thread that is separate from the OAM application thread. This is so collection table updates can be made without affecting any pending DB2 processing by the OAM application. Start the OTIS address space if you plan to use any of the functions that are associated with the OSREQ application interface, the OSMC Single Object Recovery or Recall to Disk functions, or need to remove a subsystem from the OAM configuration. Whether using a classic OAM configuration or a multiple OAM configuration, only one OTIS address space is needed. In a multiple OAM configuration, the single OTIS address space performs processing for all OAM instances.

Note: The OTIS address space is automatically started during IPL if a multiple OAM configuration is defined or if a value other than NONE is specified for the DB2SSID keyword in the IGDSMSxx member of SYS1.PARMLIB. The D A,OTIS command can be used to see if it is active.

Related reading: See *z/OS DFSMS OAM Application Programmer's Reference* for further information on the OSREQ application interface.

Optical storage

Optical disks are generally used for storing objects in a classic OAM configuration that are accessed infrequently, primarily because of their high capacity and performance characteristics. Optical storage is not supported in a multiple OAM configuration. This section provides an overview of optical storage and its role in OAM. Table 3 on page 36 lists the optical devices that OAM uses.

Table 3. Optical devices that OAM supports

Device name	Disk size	Storage slots	Internal drives	Operator-accessible	Media supported	Attaches to
3995-131	5.25 inch	144	4	1	sd-REWR	Attaches to host system.
3995-132	5.25 inch	144	4	1	sd-WORM	Attaches to host system.
3995-133	5.25 inch	144	4	1	sd/dd WORM/REWR	Attaches to host system.
3995-111	5.25 inch	144	4	0	sd-REWR	Attaches to Model 131.
3995-112	5.25 inch	144	4	0	sd-WORM	Attaches to Model 132.
3995-113	5.25 inch	144	4	0	sd/dd WORM/REWR	Attaches to Model 133.
3995-C3A	5.25 inch	0	0	1 or 6	sd/dd/qd/8*x* WORM/REWR	Attaches to host system.
3995-C32	5.25 inch	52	2	0	sd/dd/qd/8*x* WORM/REWR	Attaches to Model C3A.
3995-C12	5.25 inch	52	2	0	sd/dd/qd/8*x* WORM/REWR	Attaches to Model C3A/C32 pair.
3995-C34	5.25 inch	104	2 or 4	0	sd/dd/qd/8*x* WORM/REWR	Attaches to Model C3A.
3995-C36	5.25 inch	156	4 or 6	0	sd/dd/qd/8*x* WORM/REWR	Attaches to Model C3A.
3995-C16	5.25 inch	156	4 or 6	0	sd/dd/qd/8*x* WORM/REWR	Attaches to Model C3A/C36 pair.
3995-C38	5.25 inch	258	4 or 6	0	sd/dd/qd/8*x* WORM/REWR	Attaches to Model C3A.
3995-C18	5.25 inch	258	4 or 6	0	sd/dd/qd/8*x* WORM/REWR	Attaches to Model C3A/C38 pair.
3995-SW3*	5.25 inch	N/A	N/A	N/A	sd/dd/qd WORM/ REWR	This drive is used in the C3A, C1*x*, and C3*x* libraries.
3995-SW4**	5.25 inch	N/A	N/A	N/A	sd/dd/qd/8*x* WORM/REWR	This drive is used in the C3A, C1*x*, and C3*x* libraries.

Table 3. Optical devices that OAM supports (continued)						
Device name	**Disk size**	**Storage slots**	**Internal drives**	**Operator-accessible**	**Media supported**	**Attaches to**

Notes:

REWR = rewritable
WORM = write-once, read-many
sd = single-density (652 MB maximum disk size)
dd = double-density (1304 MB maximum disk size)
qd = quad-density (2600 MB maximum disk size)
8*x* = eight times density (5.2 GB maximum disk size)

* The 3995-SW3 drive in the C3A, C1*x*, and C3*x* libraries is capable of reading only from single-density WORM and rewritable media, not writing to it. It can read from or write to double- and quad-density media.

** The 3995-SW4 drive in the C3A, C1*x*, and C3*x* libraries is capable of reading only from single- and double-density WORM or rewritable media, not writing to it. It can read from or write to quad- and 8*x*-density media. Quad-density and 8*x*-density drives cannot coexist in the same library.

For information concerning approximate user data disk capacities for the supported media, see Table 19 on page 92.

The approximate user data disk capacity can vary depending on file sizes. Smaller file sizes take up more space on the disk than larger file sizes due to the increase in extents.

For simplicity, the following 3995 library models are referred to as C1*x* and C3*x* respectively: C12, C16, C18, C32, C34, C36, C38.

Related reading: For more information about the 3995 optical library, see the following documents:

- *3995 Introduction and Planning Guide*
- *3995 Introduction and Planning for C-Series Models*

Optical disk cartridges

Objects are stored on optical media called *optical disks*. Each optical disk is encased in a protective housing. Together, the disk and its housing are called an *optical disk cartridge*. An optical disk has recording surfaces on both sides. Each side is referred to as an *optical volume*.

Related reading: For information concerning optical disk cartridge capacities, see Table 3 on page 36, or the following documents:

- *3995 Introduction and Planning Guide*
- *3995 Introduction and Planning for C-Series Models.*

Optical recording techniques

Using laser technology, optical disk cartridges access *optical disk drives* to seek, read, write, and delete data on optical disks through the means of two optical media recording processes:

- Write-Once, Read-Many (WORM) media
  - Reads from and writes to 5.25-inch, 12-inch, single-, double-, quad-, and 8*x*-density WORM optical disk media.
- Magneto-Optic (MO) rewritable media
  - Reads from and writes to 5.25-inch, single-, double-, quad-, and 8*x*-density, rewritable optical disk media.

Note:

1. The term "rewritable" is used within this document to depict this type of optical disk media. Also, continuous composite WORM (CCW) media is included wherever the terminology double-, quad-, and 8*x*-density WORM media is used, unless otherwise stated.
2. The 3995-SW3 optical disk drive (in all the C3A, C1*x*, and C3*x* libraries) is not capable of writing to any 3995 single-density (WORM or rewritable) media. It is capable of reading this type of media, as well as reading from and writing to 3995 double- or quad-density WORM, rewritable, or CCW optical disk media type.

3. The 3995-SW4 optical disk drive (in all the C3A, C1*x*, and C3*x* libraries) is not capable of writing to any 3995 single-, or double-density (WORM or rewritable) media. It is capable of reading this type of media, as well as reading from and writing to any 3995 quad- or 8*x*-density WORM, rewritable, or CCW optical disk media type.

Write-once, read-many recording technique

Write-once recording is an irreversible process that uses heat from a laser beam to make holes in the surface of the optical disk. Once the record is created, it cannot be altered. If the data needs to be written again, a new record is created, but the space used by the original entry is not recovered. This type of media is advantageous in instances where a permanent record is needed (for example, signed application forms), or when data is stored that will never be altered or updated (for example, in the case of items being stored on microfiche, completed forms, or X-rays). Because of the permanent nature of the data recorded, you can access WORM optical disks an unlimited number of times (read-many). See Figure 9 on page 38 for a graphical depiction of WORM recording technique.

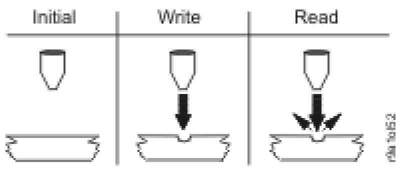


Figure 9. Write-Once, Read-Many (WORM) Recording Technique

Magneto-optic rewritable recording technique

Magneto-Optic (MO) rewritable recording is a reversible process that combines the use of magnetic and laser technologies to write, read, erase, and rewrite data. Rewritable recording is somewhat similar in concept to DASD recording. It uses a laser beam to heat the recording layer and then applies a magnetic field. The direction of magnetization changes only when the media is heated and the magnetic field is applied simultaneously. This process is used both at the time of recording and at the time of erasure. See Figure 10 on page 38 for a pictorial overview of the MO optical media recording process.

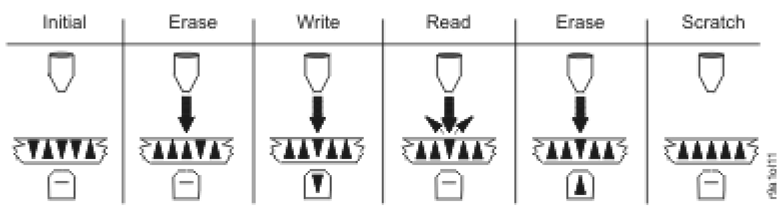


Figure 10. Rewritable Magneto-Optic (MO) Recording Technique

Optical volumes

A unique volume serial number identifies each optical volume. This unique volume serial number must be unique within the entire installation. No two volumes (regardless of the media type) can have the same volume serial number. The volume serial number must conform to z/OS volume serial number naming conventions as documented in the *z/OS MVS JCL Reference*.

Because optical volumes can reside either inside libraries or on shelves, the physical procedure for accessing volumes varies according to their location:

- When a library-resident optical volume is needed, the system mounts it on an optical disk drive in the optical library containing the volume. If a compatible optical library drive is unavailable (for example, the drive is offline or non-operational), the request to retrieve the object fails.
- When a shelf-resident optical volume is needed, the system requests that the volume be mounted on an operator-accessible optical disk drive for a 3995 volume. If an operator-accessible optical disk drive is

unavailable (for example, the drive is offline or non-operational), the request to retrieve the object fails. If the requested volume is unavailable, the operator can terminate the request.

Shelf storage can be local to the computer facility and, therefore, accessible to the optical disk drive operator, or it can be located elsewhere. The operator can enter a shelf-resident volume into any compatible optical library by placing the optical disk cartridge into the input/output station of the optical library. However, if there is an outstanding request to mount the volume on an operator-accessible optical disk drive, the attempt to insert the volume into the library is rejected.

Optical volume types

There are three types of optical volumes:

- Scratch
- Grouped
- Backup

Requirement: Both volumes comprising an optical disk *must* belong to the same Object or Object Backup storage group.

Scratch volumes

A scratch volume is an optical volume that is not yet associated with an Object storage group or an Object Backup storage group.

If an optical volume is not pre-formatted and labeled, it must be labeled with a unique volume serial number before it can be accessed by OAM. This task can be completed in either of two ways:

- Labeling an unlabeled optical disk on an operator-accessible optical disk drive in response to a LABEL command which is entered by the operator
- Labeling an unlabeled optical disk on a library-resident optical disk drive as a result of being inserted into the input/output station of an optical library

In either case, the operator is asked to supply the volume serial number for each side of the optical disk. The volume labels are written and rows are created in the volume table in the OAM configuration database. The volumes are marked as scratch volumes or grouped volumes depending on how the operator replies to the CBR4432D message during label processing.

Both sides of a scratch volume become either grouped or backup volumes when a scratch volume is used to satisfy an out-of-space condition in an Object or an Object Backup storage group.

Note: Because WORM optical volumes that are full or have very little free space are not as useful as scratch volumes, the operator is notified by message CBR4451I if the kilobytes that are free are less than the SCRENTRYTHRESHOLD parameter. The message contains the number of kilobytes that are free and the percentage of free space that this represents on the volume. This gives the operator the opportunity to fail the cartridge entry process by responding through message CBR4452D, thus causing the cartridge to be ejected from the library.

Grouped volumes

A grouped volume is an optical volume that is associated with an Object storage group. Volumes are grouped to subdivide the total available optical storage. A grouped volume contains objects from a single Object storage group.

A scratch volume becomes a grouped volume when OAM uses it to satisfy a write request that specifies an Object storage group name. When a scratch volume becomes a grouped volume, both volumes on the optical disk become grouped volumes that are assigned to the same Object storage group.

Taken together, all of the optical volumes that are associated with an Object storage group constitute the optical volume portion of the hierarchy in that Object storage group.

Backup volumes

A backup volume is an optical volume that is associated with an Object Backup storage group. Backup volumes are usually intended to provide disaster recovery or to meet legal storage requirements. They contain backup copies of objects whose primary copies reside elsewhere in the object storage hierarchy. OAM can create up to two backup copies of an object. A primary copy of an object resides on a grouped volume; backup copies reside on backup volumes.

A scratch volume becomes a backup volume when it is used by OAM to satisfy a write request for the Object Backup storage group. When a scratch volume becomes a backup volume, both volumes on the optical disk become backup volumes.

Optical media types

There are a number of optical disk media types that can be used with the 3995 optical device. You can use the following media types in the optical environment:

- 5.25-inch, single-density, WORM optical disk media
- 5.25-inch, double-density, WORM optical disk media
- 5.25-inch, single-density, rewritable optical disk media
- 5.25-inch, double-density, rewritable optical disk media
- 5.25-inch, quad-density, WORM optical disk media
- 5.25-inch, quad-density, rewritable optical disk media
- 5.25-inch, 8*x*-density, rewritable optical disk media
- 5.25-inch, 8*x*-density, WORM optical disk media

Note: Unless otherwise stated, continuous composite WORM (CCW) media is included wherever the terminology double-, quad-, and 8*x*-density WORM media is used. For more information on these media types and the libraries and drives that use them, see Table 3 on page 36.

When an object is stored using the OSREQ STORE macro, it is assigned to a specific Object storage group by the SMS storage group ACS routine. If the object is stored on an optical volume, OAM selects an optical volume residing in one of the optical libraries associated with the Object storage group to which the object has been assigned. See the description of the DEFAULT MEDIA TYPE parameter in "Defining real 3995 libraries" on page 398 for more information concerning the DEFAULT MEDIA TYPE option.

Optical disk drives

An optical disk drive uses laser technology to write data to and read data from an optical disk. The optical disk drives use removable media. The following are the different types of optical disk drives:

- Library-resident (in 3995 libraries)
- Operator-accessible (3995)
- Multifunction (library resident or operator-accessible in all 3995 models except 3995-131 and 3995-132)

Library-resident optical disk drive

A library-resident optical disk drive is inside an optical library. The cartridge transport mechanism in the library mounts and demounts the optical disk cartridges for the internal disk drives.

Operator-accessible optical disk drive

In addition to internal disk drives accessing the optical disk cartridges stored inside the library (library-resident), an operator-accessible optical disk drive is provided for users who need to read, write, or delete from an optical disk without storing or retrieving it from the library. When WORM optical media is used, the data is logically deleted from the optical disks. When rewritable optical media is used, the data is physically as well as logically deleted from the optical disks.

The operator-accessible drive cannot be accessed by the library's internal cartridge transport mechanism, and the operator-accessible drive does not have access to the optical disk volumes which are stored inside of the library. Operator-accessible drives are used to access shelf-resident optical disks. A human operator mounts and demounts the shelf-resident optical disk cartridges for the operator-accessible optical disk drives.

Multifunction optical disk drives

The optical disk drives 3995-133, -113, -SW3, and -SW4 are considered *multifunction* drives. These drives can be library-resident optical disk drives, or operator-accessible optical disk drives, or both. Multifunction drives can read and write a combination of media types. For information on the media types that multifunction optical disk drives use, see "Optical storage" on page 36.

The multifunction optical disk drive capability provides the flexibility to populate the libraries containing these optical disk drives with any combination of valid optical disk media. This can be done by using the DEFAULT MEDIA TYPE.

Related reading: See the description of the DEFAULT MEDIA TYPE parameter in "Defining real 3995 libraries" on page 398 for information concerning DEFAULT MEDIA TYPE, and "Defining optical drives" on page 406 for information on defining multifunction optical disk drives.

Optical libraries

An optical library is a set of optical volumes and the optical disk drives that are associated with those volumes. The volumes within the optical library are said to be library-resident optical volumes. Optical volumes can also be located outside of the optical library. These volumes are referred to as shelf-resident optical volumes.

Shelf-resident optical volumes can be associated with operator-accessible optical disk drives, or both, that are used to create a pseudo optical library. For more information, see "Pseudo optical library concept" on page 42.

An optical library can contain optical volumes that belongs to more than one Object storage group or the Object Backup storage group, or both.

A 3995 optical library can be specified as connected to more than one system within an OAMplex. However, the library must still be *physically* connected to only one z/OS system in an SMS complex at a time.

Optical libraries are defined to SMS and OAM using the ISMF Library Management application. For more information, see "Sample ISMF session for an IBM 3995 Optical Library Dataserver" on page 395.

Real optical libraries

A real optical library (see Figure 11 on page 42) is a storage device containing the following elements:

- An input/output station for entering into and removing cartridges from the library
- Optical disk drives
- A cartridge storage area for holding optical disk cartridges
- A cartridge transport mechanism for moving cartridges between the input/output station, slots in the cartridge storage area, and the optical disk drives

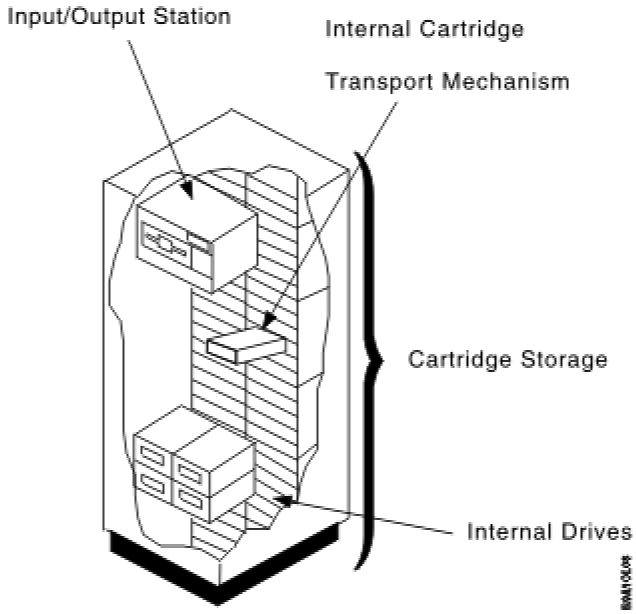


Figure 11. Real Optical Library

In real optical libraries, an optical volume in an optical library can be mounted only on an optical disk drive within the same optical library.

Note: For a grouped write request, OAM selects a volume in a library that is included in the target Object or Object Backup storage group. If a volume is in a library that is not included in the targeted Object or Object Backup storage group, the volume is not considered for write requests that are associated with that Object or Object Backup storage group. Any objects already written on the volume can still be read.

Pseudo optical library concept

In addition to real optical libraries, OAM also supports the concept of pseudo optical libraries. The concept of pseudo libraries has changed from previous releases; however, pseudo libraries defined in previous releases are still supported.

Recommendation: Convert to the new pseudo library concept to allow more flexibility and customization within your storage environment. This conversion prepares you for the eventual elimination of the device type association restrictions required with previous pseudo libraries.

A pseudo library is a collection of optical volumes sitting on shelves and operator-accessible drives that these volumes can be mounted on. A pseudo library works like a real optical library, except humans instead of robots are moving the cartridges from the shelves to the drives. The controller for the 3995-C*xx* library is a PC tower. The external drives are mounted in drive bays in the controller. You can attach one or more optical libraries to the controller.

The operator-accessible drives are physically attached to a real library controller, but logically belong to a pseudo library. For example, you could have three 3995-C3A controllers, each with six operator-accessible drives. You could map those eighteen operator-accessible drives in several ways:

- Assign all eighteen operator-accessible drives to a single pseudo library.
- Split the operator-accessible drives into three pseudo libraries, each containing six drives.
- Assign twelve operator-accessible drives to one pseudo library and the other six drives to the second pseudo library.

An example of a configuration that is not valid would be to map fifteen drives to pseudo library #1 and map three drives to pseudo library #2. All the drives on any given controller must belong to the same pseudo library.

Defining optical volumes in pseudo libraries

The installation defines these optical volumes, which are not necessarily of the same device and media types. Pseudo libraries are defined without a device type. When operator-accessible drives are defined, they can be assigned to a pseudo library chosen by the person defining the devices. These devices can be grouped in a manner that best fits the needs of the installation (for example, physical location, device and media affinity, backup and primary objects stored together, and so forth). There is no limit on the number of pseudo libraries that can be defined within an active SMS configuration.

Pseudo libraries defined prior to DFSMS/MVS 1.5 are still supported. These pseudo libraries were defined as a collection of one or more operator-accessible drives of the same device type, and one or more shelf-resident optical volumes of a like media type. OAM continues to honor these old pseudo library definitions. Support for both concepts allows installations wishing to convert their environments to the new pseudo library concept over a period of time the ability to use their previously defined pseudo libraries during the transition period.

Your installation needs to determine the pseudo library to which an optical volume is to be associated. This determination is made either when the volume is ejected from a real library, or when a volume is labeled on an operator-accessible drive.

During OAM initialization, if a volume record is encountered with an associated library name that is not known to the active configuration (ACDS), a message is issued indicating that the volume is ignored and that the library must be defined to the configuration before the volume will be recognized. For a shelf-resident volume, the library name associated with a volume is that of the pseudo library.

The default pseudo library definitions created by OAM are temporary definitions that exist only while the OAM address space is active. They are not part of the active SMS configuration (ACDS). If no pseudo library is defined within the OAM configuration database, OAM defines a pseudo library for each supported drive type. The reserved optical library names for these OAM-defined pseudo libraries for the shelf-resident volumes are as follows:

- PCTREUSE for 3995-131 drives and single-density rewritable volumes
- PCTWORM for 3995-132 drives and single-density WORM volumes
- P3995133 for 3995-133 drives and double-density volumes
- P3995SW3 for 3995-SW3 drives and quad-density volumes
- P3995SW4 for 3995-SW4 drives and 8*x*-density volumes

Attention: The installation must make certain that drives in the pseudo library are available to perform the requested task so that OAM can direct requests to a compatible drive for the task. Otherwise, the request fails, and an error message indicates that there were no available drives for the request. A given drive might be able to read requests but not write requests for a given media.

Related reading: For more information, see "Associating pseudo libraries" on page 297 and "Labeling an optical disk on a 3995 operator-accessible drive" on page 299. For more information on media compatibility and capability for optical drives, see Table 3 on page 36.

Associating ejected optical volumes with pseudo libraries

Once an optical volume is ejected from a real optical library, it becomes shelf-resident and is associated with a pseudo optical library determined by the installation. For more information concerning associating ejected volumes with pseudo libraries, see "Associating pseudo libraries" on page 297.

Mounting a shelf-resident optical volume on an operator-accessible drive

When a read request for a 3995 shelf-resident volume that belongs to a default pseudo library (with device type association) is received by OAM, any operator-accessible drive belonging to a pseudo library with an associated device type that is read-compatible with the volume is eligible to mount the volume.

When a read request for a 3995 shelf-resident volume that belongs to a pseudo library with no device type association is received, any operator-accessible drive (within the set of drives assigned to that pseudo library) with a device type that is read-compatible with the volume is eligible for the request.

If the request is a write request, drive selection is based on the drives in libraries associated with the storage group. A drive must belong to a library that is associated with the storage group in order to be considered. If a volume belonging to a pseudo library with a device type association is selected, only drives that are write-compatible and belong to a pseudo library that also has a device type association are eligible for the request. If the volume selected belongs to a pseudo library that has no device type association, then only drives that are write compatible with the selected volume and belong to the same pseudo library as the volume are eligible for the request. You can isolate volumes and operator-accessible drives by physical location instead of by device type. An installation can choose to have a large pseudo library if everything is in the same location, or have several pseudo libraries in various locations as long as there are associated drives that can satisfy the request.

When a shelf-resident optical volume is mounted on an operator-accessible drive in an OAMplex, the volume is managed and controlled by the instance of OAM to which the drive belongs. Any requests for the volume are then sent to the OAM where the volume is mounted, thus eliminating the need to demount and remount the volume.

Failing read/write requests for pseudo libraries

If a shelf-resident optical volume is associated with a pseudo optical library that does not contain any operator-accessible optical disk drives, then requests to write data to that volume or requests to read data from that volume fail because there is no optical disk drive on which to mount the volume. OAM never asks for a specific shelf-resident optical disk volume (by volume serial number) to be entered into a specific optical disk library for reading data from or writing data to the volume.

Tape storage

Tape volumes provide a low cost storage medium for storing primary and or backup copies of objects. Storing objects on tape volumes in conjunction with disk (DB2 tables or file system) and optical media provides flexibility and efficiency within the storage management facility. All devices within the installation can be used in concert with each other, complementing the performance objectives of the objects that reside on each medium. Objects can be migrated from disk to tape to optical disk or any combination of these three media, providing the most cost-effective method for meeting your data storage objectives.

Table 4 on page 45 describes hardware configurations that you can use separately or in specific combinations to create or modify your tape storage environment for OAM. The notes that correspond to the reference numbers in the table are listed at the end of the table.

The following drive types are supported in an IBM tape library environment:

3480
Identified in JCL statements as UNIT=3480. Only supported in the MTL.

3490
Identified on JCL statements as UNIT=3480X. Supported in the 3495 ATLDS and in the MTL.

3490E
Identified on JCL statements as UNIT=3490. Supported in the 3495 and 3494 ATLDS and in the MTL.

3590-1
Identified on JCL statements as UNIT=3590-1. Supported in the 3495 and 3494 ATLDS and in the MTL as a 3590 Model B Tape Subsystem.

3590-E
Identified on JCL statements as UNIT=3590-1. Supported in the 3494 ATLDS and in the MTL in 3590-1 emulation mode as a 3590 Model E Tape Subsystem.

3590-H
Identified on JCL statements as UNIT=3590-1. Supported in the 3494 ATLDS and in the MTL in 3590-1 emulation mode as a 3590 Model H Tape Subsystem.

3592-J
: Identified on JCL statements as UNIT=3590-1. Supported in the 3494 and 3584 ATLDS and in the MTL in 3590-1 emulation mode as a 3592 Model J Tape Subsystem.

3592-2
: Identified on JCL statements as UNIT=3590-1. Supported in the 3494 and 3584 ATLDS and in the MTL in 3590-1 emulation mode as a 3592 Model E05 Tape Subsystem.

3592-2E
: Identified on JCL statements as UNIT=3590-1. Supported in the 3494 and 3584 ATLDS and in the MTL in 3590-1 emulation mode as a 3592 Model E05 Tape Subsystem with encryption capability.

3592-3E
: Identified on JCL statements as UNIT=3590-1. Supported in the 3494 and 3584 ATLDS and in the MTL in 3590-1 emulation mode as a 3592 Model E06 Tape Subsystem with encryption capability.

3592-4E
: Identified on JCL statements as UNIT=3590-1. Supported in the 3584 ATLDS and in the MTL in 3590-1 emulation mode as a 3592 Model E07 Tape Subsystem with encryption capability.

Note: The devices that are listed above are also supported in the stand-alone (non-IBM tape library) environment.

Table 4. Tape storage configurations

Library model and subsystem device type	Library attachment	Media supported		Recording technology	Noncompacted data capacity
3494 L10 (VTS) Models B10, B16, B18, B20					
3490E	Yes	MEDIA1/2	(R/W)	36	400MB, 800MB
3957 V0x, VEx (VTS)					
3490E	Yes	MEDIA1/2	(R/W)	36	400MB, 800MB
3958 DE2 (VTS)					
3592-J (3590-1 emulation)	Yes	MEDIA5	(R/W)	EFMT1	100GB
3495 L20, L30, L40, L50					
3490	Yes	MEDIA1	(R/W)	18	200MB
3490E	Yes	MEDIA1 MEDIA1/2	(R) (R/W)	18 36	200MB 400MB, 800MB
3590-1 [2]	Yes	MEDIA3/4	(R/W)	128	10 GB, 20 GB
3494 L10					
3490E	Yes	MEDIA1 MEDIA1/2	(R) (R/W)	18 36	200MB 400MB, 800MB
3590-1 [2]	Yes	MEDIA3/4	(R/W)	128	10 GB, 20 GB
3590-E [3] (3590-1 emulation)	Yes	MEDIA3/4 MEDIA3/4	(R) (R/W)	128 256	10 GB, 20 GB 20 GB, 40 GB
3590-H [4] (3590-1 emulation)	Yes	MEDIA3/4 MEDIA3/4 MEDIA3/4	(R) (R) (R/W)	128 256 384	10 GB, 20 GB 20 GB, 40 GB 30 GB, 60 GB
3592-J [5] (3590-1 emulation)	Yes	MEDIA5/6 MEDIA7/8	(R/W) (R/W)	EFMT1 EFMT1	300 GB 60 GB

Table 4. Tape storage configurations (continued)

Library model and subsystem device type	Library attachment	Media supported		Recording technology	Noncompacted data capacity
3592-2 6 (3590-1 emulation)	Yes	MEDIA5/6 MEDIA5/6 MEDIA7/8 MEDIA7/8 MEDIA9/10	(R/W) (R/W) (R/W) (R/W) (R/W)	EFMT1 EFMT2 EFMT1 EFMT2 EFMT2	300GB 500GB 60GB 100GB 700GB
3592-2E 7 (3590-1 emulation)	Yes	MEDIA5/6 MEDIA5/6 MEDIA7/8 MEDIA7/8 MEDIA9/10	(R/W) (R/W) (R/W) (R/W) (R/W)	EFMT1 EFMT2/EEFMT2 EFMT1 EFMT2/EEFMT2 EFMT2/EEFMT2	300GB 500GB 60GB 100GB 700GB
3592-3E 8 (3590-1 emulation)	Yes	MEDIA5/6 MEDIA5/6 MEDIA5/6 MEDIA7/8 MEDIA7/8 MEDIA7/8 MEDIA9/10 MEDIA9/10	(R) (R/W) (R/W) (R) (R/W) (R/W) (R/W) (R/W)	EFMT1 EFMT2/EEFMT2 EFMT3/EEFMT3 EFMT1 EFMT2/EEFMT2 EFMT3/EEFMT3 EFMT2/EEFMT2 EFMT3/EEFMT3	300GB 500GB 640GB 60GB 100GB 128GB 700GB 1000GB
3584 L22					
3592-J 5 (3590-1 emulation)	Yes	MEDIA5/6 MEDIA7/8	(R/W) (R/W)	EFMT1 EFMT1	300 GB 60 GB
3592-2 6 (3590-1 emulation)	Yes	MEDIA5/6 MEDIA5/6 MEDIA7/8 MEDIA7/8 MEDIA9/10	(R/W) (R/W) (R/W) (R/W) (R/W)	EFMT1 EFMT2 EFMT1 EFMT2 EFMT2	300GB 500GB 60GB 100GB 700GB
3592-2E 7 (3590-1 emulation)	Yes	MEDIA5/6 MEDIA5/6 MEDIA7/8 MEDIA7/8 MEDIA9/10	(R/W) (R/W) (R/W) (R/W) (R/W)	EFMT1 EFMT2/EEFMT2 EFMT1 EFMT2/EEFMT2 EFMT2/EEFMT2	300GB 500GB 60GB 100GB 700GB
3592-3E 8 (3590-1 emulation)	Yes	MEDIA5/6 MEDIA5/6 MEDIA5/6 MEDIA7/8 MEDIA7/8 MEDIA7/8 MEDIA9/10 MEDIA9/10	(R) (R/W) (R/W) (R) (R/W) (R/W) (R/W) (R/W)	EFMT1 EFMT2/EEFMT2 EFMT3/EEFMT3 EFMT1 EFMT2/EEFMT2 EFMT3/EEFMT3 EFMT2/EEFMT2 EFMT3/EEFMT3	300GB 500GB 640GB 60GB 100GB 128GB 700GB 1000GB
3592-4E 9 (3590-1 emulation)	Yes	MEDIA5/6 MEDIA5/6 MEDIA5/6 MEDIA7/8 MEDIA7/8 MEDIA7/8 MEDIA9/10 MEDIA9/10 MEDIA9/10 MEDIA11/12 MEDIA13	(R) (R) (R) (R) (R) (R) (R) (R/W) (R/W) (R/W) (R/W)	EFMT1 EFMT2/EEFMT2 EFMT3/EEFMT3 EFMT1 EFMT2/EEFMT2 EFMT3/EEFMT3 EFMT2/EEFMT2 EFMT3/EEFMT3 EFMT4/EEFMT4 EFMT4/EEFMT4 EFMT4/EEFMT4	300GB 500GB 640GB 60GB 100GB 128GB 700GB 1000GB 1600GB 4000GB 500GB
Manual Tape Library (MTL)					
3480	N/A	MEDIA1	(R/W)	18	200MB
3490	N/A	MEDIA1	(R/W)	18	200MB

Table 4. Tape storage configurations (continued)

Library model and subsystem device type	Library attachment	Media supported		Recording technology	Noncompacted data capacity
3490E	N/A	MEDIA1 MEDIA1/2	(R) (R/W)	18 36	200MB 400MB, 800MB
3590-1[2]	N/A	MEDIA3/4	(R/W)	128	10 GB, 20 GB
3590-E[3] (3590-1 emulation)	N/A	MEDIA3/4 MEDIA3/4	(R) (R/W)	128 256	10 GB, 20 GB 20 GB, 40 GB
3590-H[4] (3590-1 emulation)	N/A	MEDIA3/4 MEDIA3/4 MEDIA3/4	(R) (R) (R/W)	128 256 384	10 GB, 20 GB 20 GB, 40 GB 30 GB, 60 GB
3592-J[5] (3590-1 emulation)	N/A	MEDIA5/6 MEDIA7/8	(R/W) (R/W)	EFMT1 EFMT1	300 GB 60 GB
3592-2[6] (3590-1 emulation)	N/A	MEDIA5/6 MEDIA5/6 MEDIA7/8 MEDIA7/8 MEDIA9/10	(R/W) (R/W) (R/W) (R/W) (R/W)	EFMT1 EFMT2 EFMT1 EFMT2 EFMT2	300GB 500GB 60GB 100GB 700GB
3592-2E[7] (3590-1 emulation)	N/A	MEDIA5/6 MEDIA5/6 MEDIA7/8 MEDIA7/8 MEDIA9/10	(R/W) (R/W) (R/W) (R/W) (R/W)	EFMT1 EFMT2/EEFMT2 EFMT1 EFMT2/EEFMT2 EFMT2/EEFMT2	300GB 500GB 60GB 100GB 700GB
3592-3E[8] (3590-1 emulation)	N/A	MEDIA5/6 MEDIA5/6 MEDIA5/6 MEDIA7/8 MEDIA7/8 MEDIA7/8 MEDIA9/10 MEDIA9/10	(R) (R/W) (R/W) (R) (R/W) (R/W) (R/W) (R/W)	EFMT1 EFMT2/EEFMT2 EFMT3/EEFMT3 EFMT1 EFMT2/EEFMT2 EFMT3/EEFMT3 EFMT2/EEFMT2 EFMT3/EEFMT3	300GB 500GB 640GB 60GB 100GB 128GB 700GB 1000GB
3592-4E[9] (3590-1 emulation)	Yes	MEDIA5/6 MEDIA5/6 MEDIA5/6 MEDIA7/8 MEDIA7/8 MEDIA7/8 MEDIA9/10 MEDIA9/10 MEDIA9/10 MEDIA11/12 MEDIA13	(R) (R) (R) (R) (R) (R) (R) (R/W) (R/W) (R/W) (R/W)	EFMT1 EFMT2/EEFMT2 EFMT3/EEFMT3 EFMT1 EFMT2/EEFMT2 EFMT3/EEFMT3 EFMT2/EEFMT2 EFMT3/EEFMT3 EFMT4/EEFMT4 EFMT4/EEFMT4 EFMT4/EEFMT4	300GB 500GB 640GB 60GB 100GB 128GB 700GB 1000GB 1600GB 4000GB 500GB
Stand Alone					
3480	N/A	MEDIA1	(R/W)	18	200MB
3490	N/A	MEDIA1	(R/W)	18	200MB
3490E	N/A	MEDIA1 MEDIA1/2	(R) (R/W)	18 36	200MB 400MB, 800MB
3590-1[2]	N/A	MEDIA3/4	(R/W)	128	10 GB, 20 GB
3590-E[3] (3490E or 3590-1 emulation)	N/A	MEDIA3/4 MEDIA3/4	(R) (R/W)	128 256	10 GB, 20 GB 20 GB, 40 GB

Table 4. Tape storage configurations (continued)					
Library model and subsystem device type	**Library attachment**	**Media supported**		**Recording technology**	**Noncompacted data capacity**
3590-H 4 (3490E or 3590-1 emulation)	N/A	MEDIA3/4 MEDIA3/4 MEDIA3/4	(R) (R) (R/W)	128 256 384	10 GB, 20 GB 20 GB, 30 GB 30 GB, 60 GB
3592-J 5 (3590-1 emulation)	N/A	MEDIA5/6 MEDIA7/8	(R/W) (R/W)	EFMT1 EFMT1	300 GB 60 GB
3592-J 5 (3490-E emulation)	N/A	MEDIA5	(R/W)	EFMT1	300 GB
3592-2 6 (3590-1 emulation)	N/A	MEDIA5/6 MEDIA5/6 MEDIA7/8 MEDIA7/8 MEDIA9/10	(R/W) (R/W) (R/W) (R/W) (R/W)	EFMT1 EFMT2 EFMT1 EFMT2 EFMT2	300GB 500GB 60GB 100GB 700GB
3592-2E 7 (3590-1 emulation)	N/A	MEDIA5/6 MEDIA5/6 MEDIA7/8 MEDIA7/8 MEDIA9/10	(R/W) (R/W) (R/W) (R/W) (R/W)	EFMT1 EFMT2/EEFMT2 EFMT1 EFMT2/EEFMT2 EFMT2/EEFMT2	300GB 500GB 60GB 100GB 700GB
3592-3E 8 (3590-1 emulation)	N/A	MEDIA5/6 MEDIA5/6 MEDIA5/6 MEDIA7/8 MEDIA7/8 MEDIA7/8 MEDIA9/10 MEDIA9/10	(R) (R/W) (R/W) (R) (R/W) (R/W) (R/W) (R/W)	EFMT1 EFMT2/EEFMT2 EFMT3/EEFMT3 EFMT1 EFMT2/EEFMT2 EFMT3/EEFMT3 EFMT2/EEFMT2 EFMT3/EEFMT3	300GB 500GB 640GB 60GB 100GB 128GB 700GB 1000GB
3592-4E 9 (3590-1 emulation)	Yes	MEDIA5/6 MEDIA5/6 MEDIA5/6 MEDIA7/8 MEDIA7/8 MEDIA7/8 MEDIA9/10 MEDIA9/10 MEDIA9/10 MEDIA11/12 MEDIA13	(R) (R) (R) (R) (R) (R) (R) (R/W) (R/W) (R/W) (R/W)	EFMT1 EFMT2/EEFMT2 EFMT3/EEFMT3 EFMT1 EFMT2/EEFMT2 EFMT3/EEFMT3 EFMT2/EEFMT2 EFMT3/EEFMT3 EFMT4/EEFMT4 EFMT4/EEFMT4 EFMT4/EEFMT4	300GB 500GB 640GB 60GB 100GB 128GB 700GB 1000GB 1600GB 4000GB 500GB

Table 4. Tape storage configurations (continued)

Library model and subsystem device type	Library attachment	Media supported	Recording technology	Noncompacted data capacity

Note:

1. MB = 1&finespace;048&finespace;576 bytes
   GB &finespace;= 1&finespace;073&finespace;741&finespace;824&finespace; bytes
   (R) &finespace;= Read only
   (R/W) = Read and write
   MEDIA1 = IBM Cartridge System Tape
   MEDIA2 = IBM Enhanced Capacity Cartridge System Tape
   MEDIA3 = IBM High Performance Cartridge Tape
   MEDIA4 = IBM Extended High Performance Cartridge Tape
   MEDIA5 = IBM Enterprise Tape Cartridge
   MEDIA6 = IBM Enterprise WORM Tape Cartridge
   MEDIA7 = IBM Enterprise Economy Tape Cartridge
   MEDIA8 = IBM Enterprise Economy WORM Tape Cartridge
   MEDIA9 = IBM Enterprise Extended Tape Cartridge
   MEDIA10 = IBM Enterprise Extended WORM Tape Cartridge
   MEDIA11 = IBM Enterprise Advanced Tape Cartridge
   MEDIA12 = IBM Enterprise Advanced WORM Tape Cartridge
   MEDIA13 = IBM Enterprise Advanced Economy Tape Cartridge
2. 3590-1 represents the 3590 Model B Tape Subsystem and is a system-defined esoteric.
3. 3590-E represents the 3590 Model E Tape Subsystem and is not a system-defined esoteric. It is supported in a library as a 3590 Model E tape subsystem rather than what it is emulating.
4. 3590-H represents the 3590 Model H Tape Subsystem and is not a system-defined esoteric. It is supported in a library as a 3590 Model H tape subsystem rather than what it is emulating.
5. 3592-J represents the 3592 Model J Tape Subsystem and is not a system-defined esoteric. It is supported in a library as a 3592 Model J tape subsystem rather than what it is emulating.
6. 3592-2 represents the 3592 Model E05 Tape Subsystem and is not a system-defined esoteric. It is supported in a library as a 3592 Model E05 tape subsystem rather than what it is emulating.
7. 3592-2E represents the 3592 Model E05 Tape Subsystem with encryption capability and is not a system-defined esoteric. It is supported in a library as a 3592 Model E05 tape subsystem with encryption capability rather than what it is emulating.
8. 3592-3E represents the 3592 Model E06 Tape Subsystem with encryption capability and is not a system-defined esoteric. It is supported in a library as a 3592 Model E06 tape subsystem with encryption capability rather than what it is emulating.
9. 3592-4E represents the 3592 Model E07 Tape Subsystem with encryption capability and is not a system-defined esoteric. It is supported in a library as a 3592 Model E07 tape subsystem with encryption capability rather than what it is emulating.
10. The library models indicated can be configured with any combination of correlating tape subsystem devices. These configurations may vary in the number of drives, slots, and media type supported in the libraries.

Related reading: For information about cartridge storage feature options and cartridge capacities for these tape devices, see the following documents:

- *TotalStorage Automated Tape Library (3494) Introduction and Planning Guide*
- *3480 Models A11/A22 and B11/B22 Introduction*
- *3490 A and B Models Introduction*
- *3490 Models C10, C11, C1A, C22 and C2A Introduction*
- *3590 Introduction and Planning Guide*
- *3592 Introduction and Planning Guide*
- *z/OS DFSMS Software Support for IBM TotalStorage Tape System 3590 Models E1x/H1x*
- *z/OS DFSMS Software Support for IBM System Storage TS1140, TS1130, and TS1120 Tape Drives (3592)*

Tape capacity and compaction

Use Table 4 on page 45 to determine the estimated capacity of an IBM cartridge tape when OAM is storing either objects or the backup copies of objects on IBM cartridge tape written on IBM tape subsystems.

The capacity of an IBM cartridge tape written by OAM and containing the primary or backup copies of OAM objects can be affected by a variety of factors, such as:

- The size of the object being stored.
- Whether the data OAM is storing is already hardware or software compacted.
- The tape compaction specified for the Object or Object Backup storage group using the SETOAM TAPECOMPACTION statement in the CBROAM*xx* PARMLIB member.
- The tape volume percent-full specified for the Object or Object Backup storage group using the SETOAM TAPEPERCENTFULL statement in the CBROAM*xx* PARMLIB member.
- The tape full threshold specified for the Object or Object Backup storage group using the SETOAM TAPEFULLTHRESHOLD statement in the CBROAM*xx* PARMLIB member.

If the object being stored is relatively small (16 KB or less), then the capacity of the tape cartridge can be substantially reduced. Likewise, if the size of the object being stored on tape cartridges is large, the capacity of the tape cartridge can be increased and better used. The smaller the object size, the more buffer space is required to separate the objects.

For example, if the data on an Enterprise Tape Cartridge 3592 compacts at a 3 to 1 ratio, the tape (on a 3592 Model J) can store as much as 900 GB (using EFMT1).

Recommendation: The tape compaction capability provides hardware compaction in the tape control unit and can increase the effective capacity of the tape media. Enable the compaction feature when OAM is writing primary copies or backup copies of objects to tape.

If the data that OAM is storing already is compacted, you should not expect any increase in the effective capacity of a tape cartridge due to the use of compaction. This situation is true for image data (such as ImagePlus® algorithm suited for image data). In addition, if the application invoking OAM for storing data is compacting the data, such as with the Item Access Facility (IAF) program, you should not expect an increase in the effective capacity of a tape cartridge using compaction.

OAM provides the capability for each Object or Object Backup storage group to specify what percent full OAM is to fill each tape cartridge belonging to the storage group. This option is specified with the TAPEPERCENTFULL keyword on the SETOAM statement in the CBROAM*xx* PARMLIB member.

Recommendation: If you want OAM to fill the tape cartridges to a certain percentage of their estimated capacity, reduce the approximate capacities listed in Table 4 on page 45. If your installation specifies the tape volumes should be filled to 90% of the estimated capacity, reduce the approximate capacities listed in the prior table by 10%.

Tape capacity and performance scaling

Performance scaling, also known as capacity scaling, is a function that allows you to contain data in a specified fraction of the tape, yielding faster locate and read times.

The 3592 Model J provides performance scaling by using the first 60 GB of MEDIA5 physical tape or capacity scaling by using the full 300 GB of physical tape. Both the base and encryption-capable 3592 Model E05 and 3592 Model E06 provide performance scaling, allowing you to optimize performance for MEDIA5 and MEDIA9 cartridges. The 3592 Model E07 provides the performance scaling option on MEDIA9 and MEDIA11 cartridges. Performance scaling for the 3592 Model E05, 3592 Model E06, and 3592 Model E07 limits the data written to the first 20% (the optimally scaled performance capacity) of the cartridge. Use the ISMF Data Class application to set the performance scaling attribute for tape allocations in an IBM tape library or in the stand-alone (non-SMS managed) tape environment. The performance scaling attribute allows you to select optimal performance for certain types of jobs and applications. The default setting is to use the full capacity of the tape.

In the OAM object tape environment, you can define a new SETOAM DATACLASS parameter that specifies a data class that uses the performance scaling attribute for a 3592 tape device. For instance, you might choose to use performance scaling for your primary object data (one or more Object storage groups), and use the full tape capacity for your backup object data.

Performance segmentation considerations

In addition to performance scaling, performance segmentation is a function that allows you to divide the tape into longitudinal segments. Using this optional data class specification, it is possible to segment the tape into two segments: one as a fast access segment to be filled first, and the other as additional capacity to be filled after the first segment is filled.

The 3592 Model J tape subsystem supports the performance segmentation option on the IBM Enterprise Tape Cartridge (MEDIA5). The 3592-2 and 3592-2E, 3592-2E, and 3592-3E models support the performance segmentation option on the IBM Enterprise Tape Cartridge (MEDIA5) and the IBM Enterprise Extended Tape Cartridge (MEDIA9). The 3592-4E model supports the performance segmentation option on the IBM Enterprise Extended Tape Cartridge (MEDIA9) and on the IBM Enterprise Advanced Tape Cartridge (MEDIA11). Where applicable, both the encryption and the non-encryption formats are supported. When using the performance segmentation option, the overall capacity of the cartridge is limited to 86.6% of the total capacity. The fast access segment occupies the first 20% of the cartridge, followed by the slower access segment. So, for example, using EFMT2 or EEFMT2, a MEDIA5 cartridge written on a 3592 Model E05 has a capacity of 500 GB. If the cartridge is performance segmented, the MEDIA5 cartridge is segmented into a 100 GB fast access segment and a 333 GB slower access segment (for a total capacity of 433 GB). By default, the MEDIA5, MEDIA9, or MEDIA11 cartridge is used to its full capacity. When written from loadpoint, the segmented tape cartridge is reformatted according to the assigned data class.

Because a segmented cartridge only has one physical partition and one EOV indicator, data can only be written to the slower access segment after the fast access segment has been filled. If an application wants to manage what data is placed in which segment, the application needs to manually track and fill the fast access segment before it can place less frequently accessed data in the slower access segment.

Note: A cartridge can be defined for performance scaling or performance segmentation, but not both.

KB tracking

For each tape volume, the following data-related columns are tracked in the DB2 TAPEVOL table in kilobytes:

- The capacity of the tape volume
- The free space on the tape volume
- The number of logical bytes written to a tape volume
- The number of physical bytes written to a tape volume
- The number of logical bytes deleted from a tape volume

With larger capacity tapes, the possibility exists that these fields may overflow their signed 4-byte value. To handle this situation, five overflow-related columns (one for each of the columns above) now exist in the DB2 TAPEVOL table to account for the number of overflow KBs (in 2 GB increments). The overflow field will indicate the number of times the main 4-byte field has overflowed. The following example illustrates the logic that is implemented using the number of logical bytes written:

- A tape volume has 2147983647 KB logically written to it. This value is more than a signed 4-byte field will hold in KBs (which is 2147483647).
- Dividing the KB's written by 2 GB (2147483648) will result in the number of overflow KB's (in 2 GB increments) as well as the remainder resulting from the division. In this example, the overflow value will be 1 and the remainder from the division will be 499999:

  ```
  2147983647 / 2147483648 = 1 with a remainder of 499,999
  ```

- The main column in the DB2 TAPEVOL table that represents the amount of data logically written to the tape volume in KB will be 499999.
- To determine the amount of data on the tape volume (in KB), multiply the overflow value that is associated with the amount of data logically written to the tape volume by (2 GB) which is 2147983648 and then add the result of the multiplication to the value in the main field that represents the amount of KB logically written to the tape volume.
- The new overflow column that represents the number of 2 GB's written to the tape volume will be 1.

Tape encryption support

Data encryption is an important tool for protecting against the possible misuse of confidential information, which could occur if tapes are lost or stolen. The 3592 Model E05, Model E06, and Model E07 support tape encryption with the actual encryption and decryption of the data occurring outboard in the tape drive itself. For further discussion of encryption-enablement and any MES capabilities in 3992 Models E05 and E06, refer to *IBM System Storage TS1130 Tape Drive and TS1120 Tape Drive and Controller Introduction and Planning Guide 3592* Models J1A, E05, E06, EU6, J70, and C06 and *IBM System Storage TS1130 Tape Drive and TS1120 Tape Drive and Controller Operator Guide 3592* Models J1A, E05, E06, EU6, J70, and C06. For further discussion of encryption-enablement and any MES capabilities in the 3592 Model E07, refer to *IBM System Storage TS1130 Tape Drive and TS1120 Tape Drive and Controller Introduction and Planning Guide 3592* Models J1A, E05, E06, E07, J70, C06, and C07 and *IBM System Storage TS1130 Tape Drive and TS1120 Tape Drive and Controller Operator Guide 3592* Models J1A, E05, E06, E07, J70, C06, and C07.

With the DFSMS tape subsystem encryption support, you can specify data class to have data encrypted when it is stored on an encryption-capable tape drive. In addition to this, the key label-related information that is used to encrypt the data key (of a tape cartridge) can be specified through the DD statement (JCL, dynamic, allocation and TSO ALLOCATE), data class, or encryption key manager (EKM) defaults. When the encryption-capable tape drive needs a key to perform an encrypted write, a data key is generated by the EKM. The data key used to encrypt the data on a tape cartridge is itself encrypted (using the public key of a public/private key pair) with either one or both key encrypting keys (KEKs) stored in the key stores. The KEKs are maintained by the EKM through an existing key store and are pointed to by the appropriate KEK label, which is also referred to as the key label.

The communication path to the encryption key manager (EKM) is across TCP/IP with the choice to go either in-band or out-of-band for the key management flow. With out-of-band key management, the communication path to the encryption key manager is handled by the control unit going directly to the encryption key manager. With in-band key management, the communication path to the encryption key manager is handled across ESCON/FICON with an IOS proxy interface in z/OS then handling the key exchange (across TCP/IP) with the encryption key manager. The IOS proxy interface supports both a primary and a secondary encryption key manager.

An encryption-capable 3592 Model E05 records in the existing non-encryption enterprise format 1 (EFMT1) and enterprise format 2 (EFMT2) recording formats, and also records in the encryption specific recording format (enterprise encrypted format 2 (EEFMT2)). The EEFMT2 recording format is supported across the MEDIA5, MEDIA6, MEDIA7, MEDIA8, MEDIA9, and MEDIA10 media types. Although the 3592 Model E05 can record in a lower (EFMT1) and a higher (EFMT2) recording format, an encrypted version of the lower recording format (EFMT1) is not supported. Only the higher recording format (EFMT2) is supported with an encrypted version (EEFMT2). You can also use the Performance Scaling and Performance Segmentation data class options, applicable with MEDIA5 and MEDIA9, with the encryption format EEFMT2. The capacities of EMFT2 and EEFMT2 written tapes are the same.

The 3592 Model E06 records in non-encryption enterprise format 2 (EFMT2) and 3 (EFMT3), as well as encrypted enterprise format 2 (EEFMT2) and 3 (EEFMT3), but does not record in non-encryption enterprise format 1 (EFMT1). The encryption formats (EEFMT2 and EEFMT3) are supported across the MEDIA5, MEDIA6, MEDIA7, MEDIA8, MEDIA9, and MEDIA10 media types. You can also use the Performance Scaling and Performance Segmentation data class options, applicable with MEDIA5 and MEDIA9, with EEFMT2 or EEFMT3. The capacities of EMFT3 and EEFMT3 written tapes are the same.

The 3592 Model E07 records in non-encryption enterprise format 3 (EFMT3) and 4 (EFMT4), as well as encrypted enterprise format 3 (EEFMT3) and 4 (EEFMT4). EFMT3 and EEFMT3 can be written to media types MEDIA9 and MEDIA10 and EFMT4 and EEFMT4 can be read from and written to media types MEDIA9, MEDIA10, MEDIA11, MEDIA12 and MEDIA13. The 3592 Model E07 can also read the older recording formats EFMT1 (MEDIA5 through MEDIA8) and EFMT2, EEFMT2, EFMT3, and EEFMT3 (MEDIA5 through MEDIA10). You can also use the Performance Scaling and Performance Segmentation data class options, applicable with MEDIA9 and MEDIA11, with EEFMT3 or EEFMT4. The capacities of EMFT4 and EEFMT4 written tapes are the same.

When writing from the beginning of tape (file sequence 1, DISP=NEW), the encryption-capable 3592 Model E05 drive records in the non-encryption recording format (EFMT2) by default; this default is set by z/OS OPEN processing. Lower format EFMT1 and encryption format EEFMT2 must be explicitly requested through data class. The 3592 Model E06 drives records in the non-encryption recording format (EFMT3) by default. This default is set by z/OS OPEN processing. Lower formats EFMT2 and EEFMT2, as well as the encryption format EEFMT3, must be explicitly requested through data class. The 3592 Model E06 will not write in recording format EFMT1. The 3592 Model E07 drives records in the non-encryption recording format (EFMT4) by default. Lower formats EFMT3 and EEFMT3, as well as the encryption formats EEFMT4, must be explicitly requested through data class.

When writing from the beginning of the tape (file sequence 1, DISP=OLD), since this processing does not go through the data class ACS routine, OPEN processing determines if the previous usage of the tape was encrypted and if encrypted, OPEN will explicitly set the EEFMT2 format (3592 Model E05), the EEFMT3 format (3592 Model E06), or the EEFMT4 format (3592 Model E07) with the volume's existing key management-related information being used by the drive to encrypt the data.

For an encrypted tape cartridge, the cartridge stores not only the encrypted user data but also critical key management-related information which is needed to interact with the key manager when decrypting data on the cartridge. A mix of data written in encrypted and non-encrypted formats is not supported on the same tape cartridge; whether the data on a cartridge is written in encrypted format is determined during OPEN processing, when the first file sequence on the tape is written. If the first file written to a tape is in the encrypted format; all subsequent files written to that same tape cartridge will be written in the encrypted format. All files written to a cartridge in the encrypted format are encrypted using the same data key. The exception to this is the volume label structure for the first file sequence, which is encrypted using a key known to all encryption-capable 3592 drives, which means it is in the clear.

In the 3592 Model E05, Model E06, and Model E07 environment (system-managed or stand-alone), when writing from the beginning of tape (file sequence 1, DISP=NEW), to request encryption format, EEFMT2, EEFMT3, or EEFMT4 is specified in data class. OPEN processing passes key management-related information (such as the key labels) to the drive for subsequent communication with the key manager.

For more information regarding the DFSMS encryption support, the encryption key manager (EKM) and the IOS proxy interface to the encryption key manager, refer to *z/OS DFSMS Software Support for IBM System Storage TS1140, TS1130, and TS1120 Tape Drives (3592)*.

Tape volumes

Each tape volume is identified by a unique volume serial number. The volume serial number must conform to z/OS volume serial number naming conventions as documented in the *z/OS MVS JCL Reference*. The serial number on the tape volume cannot match the serial number assigned to any other SMS tape, disk, or optical volume within the entire storage environment of the customer. The serial number must be unique within the installation.

Tape volumes can be used either in automated or manual tape libraries, or with stand-alone tape drives.

Tape volumes reside in a protective housing known as a *tape cartridge*. Table 5 on page 53shows the types of tape media that can be housed within a tape cartridge and used on IBM tape drives, and their associated Object Tape Volume Table MEDIATYP values.

Table 5. Tape media and MEDIATYP values

Media name	**Object Tape Volume Table MEDIATYP value**
IBM Cartridge System Tape (MEDIA1)	02
IBM Enhanced Capacity Cartridge System Tape (MEDIA2)	04
IBM High Performance Cartridge Tape (MEDIA3)	05
IBM Extended High Performance Cartridge Tape (MEDIA4)	06
IBM Enterprise Tape Cartridge (MEDIA5)	07
IBM Enterprise WORM Tape Cartridge (MEDIA6)	08

Table 5. Tape media and MEDIATYP values (continued)	
Media name	**Object Tape Volume Table MEDIATYP value**
IBM Enterprise Economy Tape Cartridge (MEDIA7)	09
IBM Enterprise Economy WORM Tape Cartridge (MEDIA8)	10
IBM Enterprise Extended Cartridge (MEDIA9)	12
IBM Enterprise Extended WORM Cartridge (MEDIA10)	14
IBM Enterprise Advanced Tape Cartridge (MEDIA11)	16
IBM Enterprise Advanced WORM Tape Cartridge (MEDIA12)	17
IBM Enterprise Advanced Economy Tape Cartridge (MEDIA13)	18

When objects are stored on tape volumes through an OSREQ STORE request, they are assigned to a specific Object storage group. The OAM Sublevel (OSL) parameter of the SMS storage class associated with that object is used to direct the store to either a tape sublevel 1 volume or a tape sublevel 2 volume.

- For tape sublevel 1, OAM selects the appropriate tape cartridge type based on the DATACLASS parameter (if applicable), or the TAPEUNITNAME parameter that is specified for the storage group on the SETOAM statement. If the tape volume is allocated for a stand-alone tape drive, the TAPEUNITNAME determines the device type to be used, therefore the characteristics of the tape cartridge must be consistent with the capabilities of the tape drive.
- For tape sublevel 2, similarly, OAM selects the appropriate tape cartridge type based on the L2DATACLASS parameter (if applicable), or the L2TAPEUNITNAME parameter that is specified for the storage group on the SETOAM statement. If the tape volume is allocated for a stand-alone tape drive, the L2TAPEUNITNAME determines the device type to be used, so the characteristics of the tape cartridge must be consistent with the capabilities of the tape drive.

Larger logical volume size support in the Virtual Tape Server

By default, the IBM 3494 Virtual Tape Server and the IBM TS7700 Virtualization Engine supports two logical volume sizes 400 MB and 800 MB, which correspond to the supported 3490 media types: cartridge system tape (MEDIA1) and enhanced capacity cartridge system tape (MEDIA2). Using outboard policy management support, the default volume size can be overridden at the library through a data class policy specification. If a maximum volume size is specified in the assigned data class, that volume size will override the default volume size for the volume when it is first mounted. A logical volume's maximum volume size can then change when it is mounted as a scratch volume again. However, for a TS7700 Virtualization Engine at Release 1.6 or higher, the logical volume's maximum size can change when the volume is written from load point, for either a scratch or private (specific) request. Prior to Release 1.6, the TS7700 retained the capacity of the logical volume (on reuse), unless the volume was mounted as a scratch volume. Application configuration-related changes may also be needed to fully utilize the new logical volume sizes. In the case with OAM's Object Tape Support, the TAPECAPACITY parameter in the SETOAM statement of the CBROAMxx PARMLIB member is used to specify the larger logical volumes sizes. For a 1000 MB logical volume, the capacity specified should be 1,000,000 KBs (1000 X 1000). Then for a 2000 MB logical volume, the capacity specified should be 2,000,000 KBs (2000 x 1000) and for a 4000 MB logical volume, the capacity specified should be 4,000,000 KBs (4000 x 1000). Refer to "SETOAM statements for object tape storage" on page 109 for a description of the TAPECAPACITY keyword associated with the SETOAM statement. With OA24966, and starting with Release 1.4 of the TS7700 Virtualization Engine (and only with the TS7700 Virtualization Engine), support was added in OAM to obtain the size of the logical volume from the library. With this added support, specification of the

TAPECAPACITY keyword may no longer be needed to utilize the capacity of the larger logical volumes, if the needed software support is installed and the IBM virtual tape libraries being used for OAM's object support are all TS7700 Virtualization Engines at Release 1.4 or above. For additional information on outboard policy management refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Note: For virtual tape libraries that are not at Release 1.6 (or higher) of the TS7700 Virtualization Engine, the virtual tape library will retain the capacity of the logical volume (on reuse), unless the volume is mounted as a scratch volume. When using the larger logical volume size support in the 3494 VTS or in prior release levels of the TS7700, the recommendation is to use TAPERECYCLEMODE(MVSSCRATCH) when recycling an object tape volume.

3592 media considerations

Storage groups using 3592 drives should consist of either WORM tape volumes or rewritable tape volumes. Additionally, it may not be desirable to mix extended, standard and economy length media types in the same storage group. In the IBM automated tape library environment, the SETOAM DATACLASS parameter (at the storage group or global level) can be used to specify a desired media type. By specifying a DATACLASS media interchange for the storage group, MEDIA5, MEDIA6, MEDIA7, MEDIA8, MEDIA9, MEDIA10, MEDIA11, MEDIA12, and MEDIA13 (applicable 3592 media types) can be segregated and prevent unintentional writing to the wrong media type. In the stand-alone environment, see "SETOAM keyword definitions for global level" on page 112 for assisting the tape management system in determining which media type to select for a mount request based on the storage group name being appended to the OAM tape data set names.

Related reading: For information concerning mounting, demounting, entering, and ejecting tape volumes into tape libraries or information regarding tape cartridges (requirements, capacities, and planning for their usage), see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Logical worm considerations

In addition to the 3592 physical WORM tape support, Release 1.6 of the TS7700 Virtualization Engine provides support for logical WORM (for both the TS7740 and the TS7720). Refer to APAR OA28632 (z/OS V1R9 and above) for more information. If the TS7700 is a multi-cluster grid, all clusters in the grid must be at a release level that supports logical WORM for the TS7700 to be WORM enabled. To request a logical WORM volume, an outboard data class policy must be defined (at the library) requesting logical WORM. This data class must then be assigned through the ACS routines, for a logical WORM request. A logical volume will become WORM when it is first mounted and written from load point. A volume will not take on the WORM attribute if user data had already been written to the volume. As with the 3592 physical WORM tape support, any attempt by the host to modify a previously written user data record of a logical WORM volume will be failed by the TS7700. With the 3592 WORM tape support, a volume is designated as WORM through usage of special WORM media types (MEDIA6, MEDIA8, MEDIA10, or MEDIA12). A logical volume (MEDIA1 or MEDIA2) will be designated as WORM, not by the usage of a special WORM media type, but when it is first mounted and written from load point. Refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* for more information.

Tape volume types

You can use four types of tape volumes to store objects:

- A group volume that is associated with an Object storage group
- A backup volume that is associated with an Object Backup storage group
- An MVS scratch volume that, when added dynamically, appears as belonging to an Object or Object Backup storage group when another volume is needed by OAM to satisfy a request
- An OAM scratch volume that is associated with the *SCRTCH* Object storage group. It is available to be used by any Object or Object Backup storage group that uses the same unit name and data class as this volume.

Group and backup volumes

Group and backup volumes are volumes that are already assigned to an Object or Object Backup storage group. If you issue a request to write objects to a tape volume that is already assigned to an Object or Object Backup storage group, the tape volume that OAM selects must have sufficient space available to satisfy the request. If no group or backup tape volumes with sufficient space are available to satisfy the request, OAM attempts to select an OAM scratch tape volume with a compatible unit name and data class to accommodate the write request. If no OAM scratch tape volume is available, OAM allocates an MVS scratch tape volume to accommodate the write request.

Scratch volumes

When an MVS scratch tape is added to an Object or Object Backup storage group, it is assigned the TAPEUNITNAME of the SETOAM statement that is associated with the storage group. Even though a tape unit name is specified for the group, the ACS routines (for ALLOC) can override this tape unit name specification by assigning the allocation to a Tape storage group that directs the allocation into an ATLDS or a MTL.

Example: The SETOAM statement can have a TAPEUNITNAME of 3480 associated with it. When the ACS routine runs for ALLOC, however, it overrides the information on the SETOAM statement and allocates the scratch tape to reside in an automated or manual tape library. In this case, the TAPEUNITNAME is automatically overwritten with the exact device type that was used to first mount the tape volume when it was first added to the storage group. The volume is allocated to a compatible tape device after it is ejected from the tape library dataserver.

Format of the object data on the tape media

OAM records object data on tape volumes using the BSAM OPEN, WRITE, CHECK, NOTE, POINT, SYNCDEV, and CLOSE macros to process the data recorded.

If the tape volume is a primary volume that belongs to an Object storage group and contains the primary copy of the objects, the data set name of the physical sequential data set is **OAM.PRIMARY.DATA**. Because the same data set name is created on multiple OAM tape primary volumes, the data set is not cataloged.

If the tape volume is a backup volume that belongs to an Object Backup storage group, and it contains the first backup copies of objects, the data set name of the physical sequential data set is **OAM.BACKUP.DATA**. If the tape volume is a backup volume that belongs to an Object Backup storage group, and it contains the second backup copies of objects, the data set name of the physical sequential data set is **OAM.BACKUP2.DATA**. Because OAM creates the same data set names on multiple OAM tape backup volumes, it does not catalog the data sets.

Attention:

1. If the DSNWITHSGNAME global keyword is specified on the SETOAM statement in the CBROAMxx PARMLIB member, the data set names will have the storage group name appended to the dataset names: OAM.PRIMARY.DATA.sgname, OAM.BACKUP.DATA.sgname, OAM.BACKUP2.DATA.sgname.
2. Allowing the ACS routines to assign or change the data class assignment of a tape volume is not recommended. The data class for tape volumes is determined by the SETOAM statement of the CBROAM*xx* PARMLIB member at MVS scratch tape allocation. The SETOAM statement provides this information either at the storage group level or at the OAM global level and best suits the requirements for the tape volume being allocated. Allowing the ACS routines to alter this specification could create unexpected consequences (for example, no compaction of the data when the SETOAM statement specified compaction). Your installation must ensure that the ACS routines do not alter the data class construct for OAM tape volumes.

[NOT Programming Interface Information] Each user object is recorded as one or more records within the data set. The maximum number of user object bytes within a single record is 32 628. No record contains data from more than one user object. Each record containing object data is self-describing and starts with a 128-byte prefix. The 128-byte prefix contains the following information:

- Prefix identifier

- Prefix version number
- Prefix length
- Collection name
- Object name
- Offset of first byte of user data contained in this record
- Length field containing number of bytes of user object data in this record
- Reserved space

When an OSREQ STORE macro is issued to store an object on tape, OAM physically writes the object data to the tape media, before the OSREQ STORE macro returns control to the application program. [End NOT Programming Interface Information]

Restriction: A single object never spans tape volumes.

Tape drives

In addition to optical disk drives, OAM also can store the primary copy, or the backup copy, or both, of objects on tape volumes that can be mounted on these tape drives. OAM provides support for various IBM tape subsystems (stand-alone tape drives), the automated tape library dataserver (ATLDS), and the manual tape library (MTL).

Unlike optical drives, tape drives are not defined to the system through ISMF. The system allocates the tape drives to use to satisfy read and write requests of objects. The system relies on information from the ACS routines, and the location of the volume to be mounted to determine what device should be allocated to handle the request. If the volume is a library-resident volume (residing in an ATLDS or MTL), the system chooses a device to satisfy the request. If the volume resides outside of an ATLDS or MTL, the system allocates a stand-alone drive. The drive selected for use with a stand-alone tape depends on the TAPEUNITNAME associated with that tape in the TAPEVOL table row. For an MVS scratch tape (which has no TAPEVOL table row), the TAPEUNITNAME associated with the storage group to which the tape is assigned determines the type of stand-alone device which is allocated. See Table 4 on page 45 for detailed information on all supported models.

Related reading: For more information concerning tape hardware configurations, and OAM's role with the tape library dataservers and stand-alone tape drives, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Using dynamic allocation for tape drives

OAM uses the z/OS dynamic allocation macro (DYNALLOC) for all tape drive allocations. Tape drives remain allocated while OAM is using them, but are then dynamically deallocated when OAM no longer needs them. Thus, the tape drives are obtained and released as OAM needs them; the tape drives are not dedicated to OAM. The tape drives are allocated to the OAM address space and not the application address space (for example, CICS, IMS MPP, or IMS BMP) invoking the OAM application programming interface (the OSREQ macro).

The allocated device could be a stand-alone tape drive or a drive inside of an automated or manual tape library. The device allocation depends on:

- The contents of the ACS routines
- Whether this mount is for an existing OAM tape that contains objects or a mount for an MVS scratch tape
- The location of the volume to be mounted

Although tape drives are not permanently allocated to the OAM address space, tape drives must be available to the OAM address space when they are needed to handle a request to store or retrieve an object on a tape volume.

The maximum number of tape drives capable of being concurrently allocated to the OAM address space is controlled by the sum of the MAXTAPESTORETASKS and the MAXTAPERETRIEVETASKS keywords at the OAM global level on the SETOAM statement.

Failing dynamic allocation and z/OS allocation recovery

If the initial dynamic allocation of a tape drive fails, OAM retries the dynamic allocation every 10 seconds for a full minute. If after one minute OAM does not successfully allocate the required device, OAM issues message CBR6425I indicating to the operator that OAM has not allocated a tape drive. The CBR6425I message lists the object name, collection name, storage group name, and tape volume name (SCRTCH if for an MVS scratch allocation) for this tape allocation request.

OAM continues to retry dynamic allocation every 10 seconds, for another four minutes or until a suitable tape drive is allocated, whichever comes first. During this period of time (up to five minutes) that OAM is trying to allocate a tape drive, z/OS allocation recovery processing is disabled and OAM is retrying the dynamic allocation.

If OAM does not successfully allocate a suitable tape drive at the end of five minutes, then OAM reissues message CBR6425I along with message CBR6400D. The CBR6400D message lists the storage group name and tape volume name for this tape drive allocation request and asks the operator if OAM should continue to retry with a NOWAIT option, continue to retry with a WAIT option, or cancel the request to dynamically allocate a tape drive. If the operator replies **R** (meaning retry with WAIT) to the CBR6400D message, OAM again issues the dynamic allocation macro, but with z/OS allocation recovery processing enabled. If the allocation request cannot be satisfied immediately, z/OS allocation recovery issues message IEF238D and no other dynamic allocations, dynamic deallocations, OPENs, or CLOSEs can occur in the OAM address space until this allocation completes or is canceled.

If the operator replies **N** (meaning retry with NOWAIT) to the CBR6400D message, OAM repeats the retry process from the beginning. OAM issues the dynamic allocation every 10 seconds for one minute. If after one minute OAM does not successfully allocate the required device, OAM issues the CBR6425I message. This message indicates to the operator that OAM has not allocated a tape drive. OAM continues to retry dynamic allocation every 10 seconds for another four minutes or until a suitable tape drive is allocated, whichever comes first. During this period (up to five minutes) that OAM is trying to allocate a tape drive, z/OS allocation recovery processing is disabled while OAM is retrying the dynamic allocation.

If the operator replies C (meaning cancel) to the CBR6400D message, OAM fails the tape drive allocation and its associated OAM request. Any other reply to the CBR6400D message causes OAM to reissue the CBR6425I and CBR6400D messages.

The actions performed during z/OS allocation recovery processing are affected by the options specified in the ALLOC*xx* member in PARMLIB. If an eligible device is not made available to OAM, the dynamic allocation request fails and the associated store or retrieve request for the object also fails. For more information concerning the ALLOC*xx* member of PARMLIB and the installation defaults for handling allocation requests, see *z/OS MVS Initialization and Tuning Reference*.

Note: This processing applies to stand-alone devices as well as devices inside of automated or manual tape libraries.

Retrying or canceling a volume mount

If OAM is waiting for the mount of a volume after the appropriate device has been allocated and the five minute default or the time specified on the MOUNTWAITTIME parameter of the SETOAM statement has elapsed with no mount occurring, message CBR6405D is issued to the operator to ask if the mount should be retried or canceled. If the operator replies **R**, the mount message is left on the console until the installation-specified amount of time (MOUNTWAITTIME) has again elapsed or the mount has been completed. This process continues until the requested volume is mounted or until the operator replies C. Should the operator cancel the request, one or more of the following actions occur:

- Message CBR2003I, stating that the tape volume that was requested to be mounted was not found, and is marked lost, is issued.

  Note: This message is not issued for an MVS scratch tape mount that is used to satisfy the request. This message is only issued for tape volumes that have rows in the TAPEVOL table.

- OAM marks the volume as lost so that future and current read requests for this volume fail with a nonzero return and reason code. For more information, see "Displaying volumes that have LOSTFLAG set" on page 352.

- If the request that required the mount was a write request with a corresponding row in the TAPEVOL table, OAM marks the volume as lost. OAM tape volume selection attempts to find another volume for the request. If the cancellation was for an MVS scratch tape, OAM fails the request.

Retrieving objects on devices compatible with the tape data format

Whenever an object exists on tape, OAM can retrieve the object only when a device compatible with the format of the data written on the tape volume is available at the time of the retrieve request. To retrieve any objects from tape, you must initialize OAM with a valid CBROAM*xx* specification.

If an Object or Object Backup storage group that used to have its objects written on tape is now having objects written to optical media, those objects can also be read back using a tape device compatible with the format of the data written on the tape volume available at the time of the retrieval request. To read data back from tapes previously written in a group that is no longer writing data to tape, there *does not* have to be a SETOAM statement for that group in the CBROAM*xx* PARMLIB member processed.

OAM has a default of one system read and one system write task; each group has a default of one read task and one write task. The installation should ensure that there is a compatible tape device available for allocation at the time the retrieval request is received. If there is no device available for allocation, z/OS allocation recovery issues allocation recovery messages, requesting that an offline or inaccessible device be made available. If this is not possible, the retrieval request for the pending mount fails.

Related reading: For more information, see "Using dynamic allocation for tape drives" on page 57.

Tape libraries

Tape libraries consist of a set of tape volumes and the set of tape drives on which those tape volumes can be mounted. A tape library can consist of one or more tape subsystems. These drives are configured into automated or manual tape library dataservers that contain library-resident tape volumes. The storage administrator defines tape libraries to SMS using ISMF library management definition panels. A tape library can contain tapes from multiple storage groups and a storage group can span up to eight libraries (ATLDSs, MTLs, or a combination of these).

Related reading: For more information on defining tape libraries, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Object tape and optical volume management

OAM expires optical and tape volumes that belong to Object and Object Backup storage groups. You can recycle or delete OAM object tape and optical volumes after all active data has been moved or recovered.

You can specify whether OAM is to leave the expired object tape volume in its storage group, reassign the volume as an OAM scratch volume, or remove the volume from OAM control.

You can specify whether OAM is to leave expired optical volumes in their storage group or to reassign them to OAM scratch status. The rewritable optical volume is available for reuse. OAM expires a WORM optical volume only if all objects have been expired and both sides of the optical platter are full.

The function provides the following benefits:

- Reduces significantly the amount of private storage that the OAM address space uses at larger installations.
- Frees up resources by expiring tape or optical cartridges when all of the data on the tapes has expired.
- Frees up resources by purging and recycling tape and optical volumes.

Related reading:

- For more information about using SETOAM TAPERECYCLEMODE to disposition expired object tape volumes, see "SETOAM statements for object tape storage" on page 109.
- For more information about using SETOPT OPTICALREINITMODE to disposition expired optical volumes, see "SETOPT statements for options" on page 129.

- For more information about expiring volumes, see "Expiring tape and optical volumes" on page 230.

Recycling tape and optical volumes

You can use the TAPE RECYCLE command to recycle 1 to 40 tape volumes, or use the MOVEVOL command with the RECYCLE option to recycle single tape or optical volumes. After all the objects have been moved off of the tape or optical volume, OAM processes the volume in a similar way as if it has expired. When you recycle a tape volume, you can leave it in its storage group, return the volume to OAM scratch status, or return the volume to MVS scratch status. You can reuse the tape or optical volume immediately.

When a volume is released from DFSMSdfp, DFSMSrmm also releases the volume.

Related reading: For more information about recycling volumes, see the following material:

- "Reusing recycled tape and optical volumes" on page 228
- "Moving objects and recycling the source volume" on page 281
- "Using the OAM Object Tape Volume Return to MVS Scratch exit routine" on page 233

- *z/OS DFSMSrmm Managing and Using Removable Media*

Deleting tape and optical volumes

You can use the MOVEVOL command with the DELETE option to delete tape and optical volumes. When the Move Volume process completes and no objects remain on the source tape or optical volume, OAM removes the volume from the OAM inventory. OAM does not reclaim space on rewritable optical platters that have been deleted.

Related reading: For more information about deleting volumes, see the following topics:

- "Deleting recycled tape and optical volumes from OAM" on page 229
- "Moving objects and deleting the source volume" on page 282

Deleting recovered tape and optical volumes

You can use the volume RECOVERY command with the DELETE option to delete recovered tape and optical volumes. When the recovery process completes and no objects remain on the source tape or optical volume, OAM removes the volume from the OAM inventory.

Related reading: For more information about deleting recovered volumes, see "Deleting a recovered tape or optical volume" on page 275.

Chapter 2. Planning for OAM installation

In many ways, planning is the most important phase of the OAM implementation and administration cycle. Time spent in planning is fully repaid in time, effort, and money saved by a well-implemented installation and a smooth transition to full system integration. This topic identifies key areas that must be addressed during planning. Rather than repeat large amounts of information available elsewhere in the DFSMS and storage management libraries, this topic focuses specifically on object-related issues and provides references to other resources. A case study, included at the end of this topic, illustrates how planning concepts can be applied in a typical situation.

This topic is organized into various sections, one for each phase of the planning process, and one for the case study. The following subtasks are covered:

- "Analyzing your business environment" on page 62
- "Analyzing your processing environment" on page 68
- "Estimating resource requirements" on page 71
- "Preparing the physical environment" on page 83
- "Preparing for installation and customization" on page 83
- "Planning to program applications for OAM" on page 84
- "Planning to administer OAM" on page 84
- "Preparing to operate OAM" on page 84
- "OAM planning case study" on page 85
- "Setting up the planning team" on page 61
- "Planning to use a file system" on page 80

Setting up the planning team

To most effectively implement OAM, your planning team should include individuals with significant technical expertise in the following areas:

- **DFSMS-Related Products.** OAM is a component of DFSMS and interacts closely with other members of the DFSMSdfp family of products. Knowledge of system-managed storage (SMS) and the interactive storage management facility (ISMF) is essential for successful planning and implementation. Additionally, familiarity with DFSMShsm and the other DFSMS components provides a meaningful context for understanding OAM.
- **DB2.** OAM uses DB2 databases to store objects and internal information (such as object indexes). Implementing OAM for objects is likely to have a significant impact on your installation's DB2 space requirements. If OAM is to be set up in a Parallel Sysplex, DB2 data sharing installation and knowledge is also required.
- **File systems.** If OAM is to use a file system, the planning team must evaluate the file systems provided by Network File System (NFS) and zFS.
- **Customer Information Control System (CICS).** If OAM is invoked by CICS transactions, the planning team must evaluate the effect of their interaction.
- **Hardware Configuration Definition (HCD).** HCD is used to define devices to the hardware configuration.
- **Information Management System (IMS).** If OAM is invoked by IMS transactions, the planning team must evaluate the effect of their interaction.
- **Time Sharing Option/Extended (TSO/E).** If OAM is invoked by TSO/E transactions, the planning team must evaluate the effect of their interaction.

- **Cross-system Coupling Facility (XCF).** If OAM is to be established in a Parallel Sysplex, the planning team must evaluate the impact to the coupling facility requirements and resources.

As with any major installation, the OAM planning effort should also involve people with project management experience and representatives of the end-user areas that are affected by the implementation.

Analyzing your business environment

Your processing environment reflects the unique goals, procedures, and structure of your business; therefore, you need to analyze your business environment so that you can implement OAM for objects successfully. The technical installation can then evolve logically from the functions and requirements you have defined.

The first task in the analysis process is to characterize the objects that are processed. Among the most useful classification categories are:

- **Size.** Are objects small, medium, or large? What are the criteria for these categories in your installation?
- **Activity.** How often are objects retrieved? How often are new objects stored? Is one object accessed many times or are many objects accessed one time each? What is the required response time for accessing an object?
- **Volume.** How many objects of each size will be created? How many objects must be processed every hour or every day?
- **Life cycle.** Is the activity level stable or does it change in response to a business cycle (such as monthly billing)? Are such changes random or periodic? How frequently do these changes occur? Are objects backed up? How long do you plan to retain objects? How do you plan to handle expired objects? Do you plan to delete objects automatically?
- **OAMplex.** If you are planning to run an OAMplex, how many systems (OAMs) will be in your OAMplex? Which systems will have the hardware physically attached to assist in determining where OSMC should be run? How will the storage group and library disbursement be handled between systems?
- **OAM configuration.** You must also consider whether you will implement a classic OAM configuration or a multiple OAM configuration. For example, do you have a need for multiple OAM object address spaces, perhaps to separate a production implementation from a test implementation or perhaps for multiple separate OAM production instances? Each object instance is associated with a unique DB2. This identification is done indirectly by using the DB2 subsystem identification that is associated with each storage group and also with each OAM object instance. Each OAM object instance is associated with a unique DB2 subsystem and with a unique set of storage groups.

As a result of this analysis, you can:

- Determine criteria for grouping objects.
- Establish performance objectives.
  - Determine the best system for OSMC processing per storage group.
  - Determine hardware distribution based on demand and location.
- Identify storage management cycles.

This analysis, in turn, leads you to create storage groups, storage classes, data classes, and management classes through which OAM and SMS can implement your storage management policy. The ultimate goal is to develop a set of Storage Management Subsystem (SMS) constructs that you can use to accurately describe and respond to the complex reality of your business environment.

Grouping objects

During the process of characterizing your installation's objects, you probably discovered that the objects can be grouped in various ways. OAM uses the following techniques to group objects physically and logically:

- Object storage group assignment represents the physical storage, managed by OSMC according to your storage management policy.
- Collection assignment represents a logical relationship between objects.

Every object belongs to a collection; every collection belongs to an Object storage group. Each Object storage group can contain one or more collections; however, a collection can never span multiple Object storage groups.

Object storage groups

The Object storage group makes it possible to manage a set of storage devices as a single object storage area. Each Object storage group encompasses several types of storage devices in an object storage hierarchy. (See "Using Object, Object Backup, and Tape storage groups" on page 17.)

You can organize storage into physically separate groups, such as:

- Business needs
- Accountability
- Security
- Application isolation
- Device characteristics
- Connectivity

Collections

A collection typically contains objects that are used by the same application or are of a similar type. Collections are useful for dealing with sets of objects that are too large to be handled as a single object, but too small to warrant a separate Object storage group. For example, all objects in a collection can have the same default initial storage class and management class attributes.

You can organize objects into collections for a variety of reasons. For example, if objects related to a corporate division are kept in one Object storage group, it might be desirable to subdivide that Object storage group into collections of departmental data.

Establishing performance objectives

Different response times are required for different sets of objects and some objects are accessed more frequently than others. OAM uses the storage class to specify object performance objectives and availability requirements to SMS. Every object in the object storage hierarchy must have an associated storage class. The fact that every object has an associated storage class makes every object, by definition, SMS-managed.

Your business needs provide the service-level criteria on which storage classes are built. Table 6 on page 63 shows how you can specify performance objectives for different storage classes depending on the service levels required:

Table 6. Storage class service levels for a variety of business needs

Business need	Service level
Daily operation	Fast response; frequent access
Online customer inquiries	Fast response; occasional access
Quarterly batch processing	Medium response; periodic access
Legal retention requirements	Slow response; very infrequent access

A storage class does not represent any physical storage. OAM analyzes the storage class parameters and tries to meet the performance objectives by placing the object on a device that best meets those objectives. Using storage classes to force use of a specific device type can defeat the purpose of system-managed storage and cause serious inefficiencies.

Example: Using a storage class that causes objects to be written directly to optical media without being staged through disk can degrade system performance. It also can significantly increase the number of optical disks needed per day because of the inefficient storing of optical volume table of contents (VTOC) information.

Consider separating the storage classes that are used to control objects for one application from the storage classes that are used for other applications. If it becomes necessary to change the performance objectives for objects used by an application, its associated storage classes can be changed without affecting the other applications.

Related reading: For a detailed discussion of storage classes and how to plan for them, see *z/OS DFSMSdfp Storage Administration*.

Identifying management cycles

Every business is subject to operational cycles that influence work flow. These cycles often have a direct effect on performance and availability requirements. The management class, with storage classes and ACS routines, makes it possible for SMS to respond to these cycles as it manages object storage. Every object in the object storage hierarchy must have an associated management class. (See Figure 12 on page 65 for a representation of this process.)

As you analyze your business environment, consider the potential effects of these cycles on your work load and, therefore, your object access requirements:

- Accounting
- Reporting
- Manufacturing
- Marketing
- Backup
- Retention
- Physical location

Remember to factor into your analysis the frequency of each cycle (such as daily, weekly, monthly, quarterly, or annually).

To fully exploit the management class construct, it is necessary to understand class transitions and storage management cycles. A *class transition* is a change in an object's management class or storage class when an event occurs that brings about a change in an object's service level or management criteria. Class transition criteria are specified in management class definitions. When a recalled object is restored to removable media, and its time on disk has exceeded the MC definition, the object will be processed according to the MC definition in effect.

Example: A management class might specify that 180 days from an object's creation date, the ACS routines should be invoked to determine if a class transition is needed.

Understanding storage management cycles

A class transition occurs during a storage management cycle. A *storage management cycle* is an invocation of the OAM Storage Management Component (OSMC) for an Object or Object Backup storage group. The storage management cycle ensures that every object that is scheduled for processing is placed in the correct level of the object storage hierarchy (as specified by its storage class), is deleted or backed up (as specified by its management class), and, if necessary, is flagged for action during a subsequent storage management cycle.

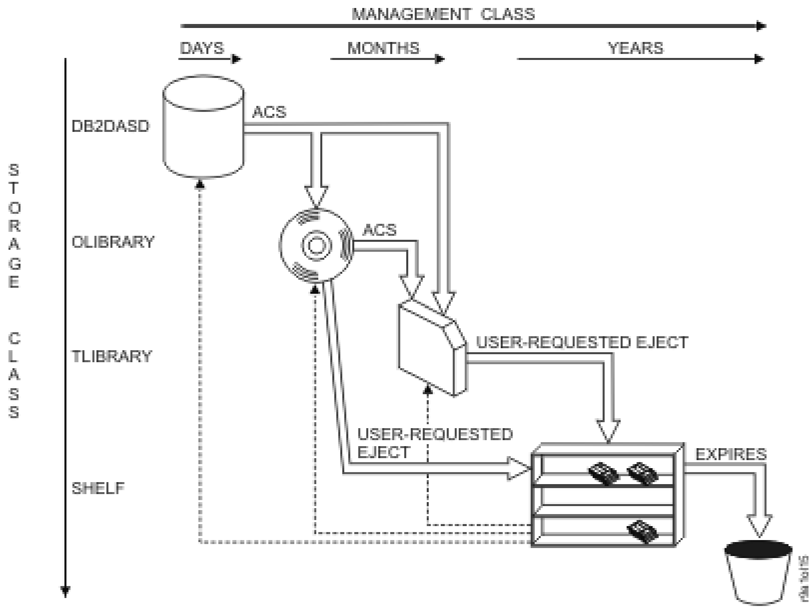


Figure 12. Example of Using Class Transitions and ACS Routines to Change Management Classes

Storage management cycles can run automatically once each day, starting at a period of time that you specify (for example, during third shift). If you are in an OAMplex and using automatic startup, you should also specify a system name; otherwise, multiple systems will try to process the same storage group at the same time. As it executes, OSMC checks to see if an object is scheduled for processing. An object is scheduled for processing for any one of the following reasons:

- It was stored since the last storage management cycle.
- It was retrieved and the UPD=N parameter is not specified on the OAM1 statement in the IEFSSN*xx* member of PARMLIB.
- Its storage class, management class, or expiration date has been changed since the last storage management cycle.
- It was marked for a class transition by a previous storage management cycle.
- It was marked for expiration by a previous storage management cycle.
- It was previously recalled to a disk sublevel for user-defined number of days, and that number of days has elapsed. OSMC will restore the object back to removable media, and process per object's management class.

Checking for object deletion

When OSMC encounters an object in the Object storage group that is scheduled for processing, it first checks to see if the object has reached its scheduled expiration. If it has, OSMC invokes the auto-delete (CBRHADUX) installation exit to determine if the object is allowed to be expired. If the auto-delete exit approves of the deletion, then the expiration is honored and all copies of the object are deleted. If the auto-delete exit denies the deletion, then the expiration is not honored and no copies of the object are deleted.

If the object is being deleted from the file system, the physical deletion is deferred and the object is added to the File System Delete Table for later physical deletion. The file system must remain mounted in the z/OS Unix file system hierarchy, as the physical deletion could be performed at any time that OAM is running, regardless of the OSMC functions that are currently started.

If the object is being deleted from an optical rewritable volume, the deleted space and deleted counts are updated, and the object name, collection name, volume serial number, and sector location are added to the deleted objects table for later physical deletion. If the object is deleted from a tape volume, the tape volume record is updated with the deleted kilobytes.

Each time OAM receives a request to delete an object from a tape volume, OAM updates the number of logical kilobytes deleted from that tape volume by adding the size of the object which was just deleted to the existing logical kilobytes deleted value for that tape.

Related reading: For more information on deleting objects, see the following topics:

- "Deleting tape and optical volumes" on page 60
- "Auto-delete installation exit (CBRHADUX)" on page 573

Recommendation: If you are using DFSMSrmm to manage OAM objects on tape, the following vital record specifications, shown in TSO/E format, might be appropriate:

```
RMM ADDVRS DSNAME('OAM.PRIMARY.DATA') COUNT(99999) LOCATION(HOME)

RMM ADDVRS DSNAME('OAM.BACKUP.DATA') COUNT(99999) LOCATION(HOME)

RMM ADDVRS DSNAME('OAM.BACKUP2.DATA') COUNT(99999) LOCATION(HOME)

RMM ADDVRS DSNAME('OPEN') JOBNAME(oamtaskname) COUNT(99999) LOCATION(HOME)

RMM ADDVRS DSNAME('ABEND') JOBNAME(oamtaskname) COUNT(99999) LOCATION(HOME)
```

These DSNAME examples are data set names without DSNWITHSGNAME specified in the SETOAM statement. If DSNWITHSGNAME were specified, the storage group name would be appended as the data set name's low level qualifier. The JOBNAME value is the name of the job and started task for the OAM address space that opens the tape data sets.

Determining class transition and backup requirements

If the object has not been marked for expiration, OSMC determines if a class transition is needed. When a class transition is indicated, OSMC invokes the ACS routines. These routines evaluate the object's class assignments and change them, if necessary. Next, OSMC performs any actions that an object's class assignments indicate:

- OSMC checks the storage class to determine if the object should be placed at a different level of the object storage hierarchy.
- OSMC checks the management class to set the *next* date on which management action (that is, expiration or class transition) is needed.
- OSMC makes up to two backup copies of the object if the management class indicates that one or two backup copies are required, and the requested number of backup copies does not currently exist.

This situation can result from any of the following scenarios:

- This storage management cycle is the first one for a new object.
- An application change changes an object's management class to one that requires backups.
- A class transition changes an object's management class to one that requires backups.
- An application change occurs so that the object's management class requires two backup copies where currently only one backup copy exists.
- A class transition occurs so that the object's management class requires two backup copies where currently only one backup copy exists.

For objects with a primary copy on disk, tape, or optical, OSMC performs the following actions:

- OSMC makes backup copies according to the management class that is assigned to the object. Backup copies can be directed to an optical disk, tape volume, or both depending on the definitions that are associated with the Object backup storage groups specified in the CBROAM*xx* member of PARMLIB.

  If the management class requires a single backup copy, OSMC directs the backup copy to the associated Object Backup storage group. This storage group can be located on either an optical disk or a tape volume. If the management class indicates that two backup copies of the object are required and the SETOSMC statement associates two Object Backup storage groups with the Object storage group where the primary object copy resides, the first backup copy is written to one Object Backup storage group and the second backup copy is written to the other Object Backup storage group. These Object Backup storage groups can reside on the same or different media types (optical or tape).

For backup copies to be made to tape volumes, you must specify SETOAM statements with the STORAGEGROUP and TAPEUNITNAME parameters for the Object Backup storage groups.

- The object is presented to the ACS routines in the CTRANS environment to allow for class transition.
- If SETOSMC CLEAROLDLOC has been specified for objects transitioning from removable media to the DB2 sublevel or file system sublevel, the old location information is cleared during this OSMC cycle.
- The next scheduled processing for the object is determined. If the object has not expired yet, OSMC processing for the object is next scheduled for when the object is to expire (based on the OSREQ STORE/CHANGE specifications for the RETPD and the management class assigned to the object).
- On expiration of the object, OSMC deletes all copies of the object (primary and all the backups). Deletion removes information from the object directory but might or might not result in physical deletion, depending on the type of media. OSMC does not physically delete object copies residing on tape and WORM media. For objects that reside on tape volumes, the number of logical kilobytes that are deleted from the volumes is incremented for each object deleted. Objects that reside on a file system will be physically deleted, but there is a time delay before such deletion.

Related reading: For more information on expiring objects, see "Objects not selected for expiration processing by OSMC" on page 239.

Developing appropriate management classes

Before you begin: Like storage classes, management classes can be developed to meet a variety of needs. Develop as many management classes as necessary to use the class transitions and storage management cycles that are required by your business. To avoid processing inefficiencies or unexpected results, or both, careful analysis of the end results of class transition is critical for a successful implementation.

Perform the following steps to develop your management classes:

- Analyze your applications to plan for large groups of objects that have the same management requirements. You can assign these objects to one management class and store them together in one collection.
- Identify objects that are good candidates for early deletion or class transition.
- Separate objects that do not need to be backed up from those that do.
- Identify objects or collections that require immediate backup.
- Identify objects that require a delayed class transition.
- Identify objects with medium to low response-time requirements, so that they can be moved to tape or optical storage as soon as possible.
- Use ISMF to define your management classes by selecting option 3, Management Class, from the ISMF Primary Option Menu for storage administrators.

Related reading: For more information on defining management classes, see *z/OS DFSMSdfp Storage Administration*.

OSMC processing management in an OAMplex

You can decide whether to start OSMC processing automatically or manually for a specific Object or Object Backup storage group. You can specify the name of the OSMC processing system name in the ISMF panel for defining/altering storage groups.

Automatic OSMC processing of storage groups

You can start OSMC processing automatically for Object and Object Backup storage groups using the OSMC cycle window in the ISMF Storage Group Define panel. If you specify the OSMC system name in the Storage Group Define panel, OSMC processes the storage groups on the specified OSMC system. If you omit the OSMC system name, OSMC processes all storage groups in the OAMplex. For more information, see "Understanding storage management cycles" on page 64.

You also can use the SETOSMC CYCLEWINDOW keyword to specify either START/ONLY or START/STOP mode for the OSMC cycle. START/STOP mode defines the start and end times for processing the storage group. START/ONLY mode defines the start time for processing the storage group. For more information on using the SETOSMC CYCLEWINDOW keyword, see "SETOSMC statements for use in the OSMC environment" on page 138. **Specifying the processing system name is important because omitting it can result in significant resource contention in DB2 and OSMC.**

Example: You have three systems in an OAMplex and all systems started processing storage group SGROUP01 at the same time. Because all three systems are in contention for the same resources associated with SGROUP01, it could result in errors when one system is waiting for resources that another system holds. The most efficient way to start OSMC processing is to use the OSMC processing system name to have each system process different storage groups at the same time.

Manual OSMC processing of storage groups

You can use the F OAM,START,OSMC command to start OSMC processing for Object storage groups only. If you issue an F OAM,START,OSMC command, OSMC starts all Object storage groups whose OSMC processing system name is blank or matches the system name on which the command was issued. You can use the F OAM,S,STORGRP command to start OSMC processing for either Object or Object Backup storage groups. The F OAM,START,STORGRP,*storagegroup* command is always honored on the system where the command is issued, even if another system name is specified in the processing system name.

Recommendation: Try to localize the OSMC processing to the system where the hardware is physically online for that storage group to reduce the amount of cross-system processing that is required. Also, if multiple systems are running different storage groups at a time, the impact to DB2 should be analyzed (especially if normal activity to the storage group is occurring at the same time).

Concluding the business analysis phase

As you have seen, OAM and SMS use a variety of conceptual structures through which you can describe your business environment and specify a storage management policy. Furthermore, each of these structures offers a significant amount of flexibility.

Unless your business environment is an unusually simple one, you should expect the analysis process to require several iterations. There are likely to be several equally viable ways to define your Object storage groups and classes. Unfortunately, there are no magic algorithms for choosing which approach to implement. That decision can be made only by one who knows the most about your business: *you*.

At this point in the planning process, you should have a rough idea of how your objects are to be organized into Object storage groups, collections, storage classes, and management classes. The next step is to analyze your processing environment.

Analyzing your processing environment

Installing a new product is rarely an isolated event. Planners must evaluate how the existing environment will be affected by the new product, as well as how the new product must be customized to integrate with the existing components. This section presents guidelines for analyzing the hardware and software that make up your processing environment.

Hardware

You would use OAM in a mainframe environment. To take advantage of the full range of OAM capabilities, the environment should provide substantial amounts of internal and external storage. In addition to standard direct access storage device (DASD) devices, OAM also uses optical disk drives and tape devices inside of and outside of ATLDSs and MTLs within an object storage environment. OAM also supports zFS on DASD devices, as well as NFS file systems exported from NFS servers and mounted in the z/OS UNIX storage hierarchy. OAM does not have any hardware prerequisites; however, you can expand internal and external storage capacities to accommodate an increased work load.

Device considerations for larger data objects

When expanding the maximum data object size beyond 50MB (using the IEFSSN*xx* keyword, MOS=*nnn*), remember that OAM does not span an object across multiple volumes. Ensure that the removable storage media used to store, backup, or transition a larger data object has sufficient capacity. See Updating the IEFSSNxx PARMLIB member for more information.

Auxiliary storage and real storage considerations

To process objects greater than 256MB on the file system sublevel or on tape, the OAM address space uses 64–bit addressing and virtual storage above the 2G bar. OAM usage of virtual storage above the 2G bar will result in significantly increased storage utilization on your system.

Attention: This will require additional capacity planning to back this virtual storage with a robust auxiliary paging subsystem backed by DASD and sufficient real storage. Your planning must be for peak OAM workloads including concurrent application initiated activity and concurrent OSMC processing. See "OSMC processing management in an OAMplex" on page 67 for the potential OSMC processing you must consider.

For more information on setting virtual storage limits and backing virtual storage with sufficient real and auxiliary storage see the migration actions in *z/OS Migration*.

Grouping tape devices

You can group together tape devices and define them as one group to the system. For instance, a group of 3490 tape drives in the same room can be grouped together and defined as 3490GRP. These tape device groups are known as *esoterics*. Once an esoteric is specified on the SETOAM statement for a group, you need to ensure the existence of that esoteric while an OAM tape exists that specifies that esoteric in the TAPEUNITNAME field of the OAM tape volume (TAPEVOL) table.

Attention: Do not change the contents of that esoteric to introduce incompatible tape device types. Should the esoteric name be deleted or changed, the volumes associated with the esoteric name cannot be allocated. Because the TAPEUNITNAME cannot be resolved, the tape that is required for the request is not mounted and the allocation request fails.

For scratch allocations for an ATLDS or MTL, the esoteric TAPEUNITNAME associated with a storage group is overridden with the exact device type for the device that is allocated for the MVS scratch tape mount.

Considering storage configurations

Use of optical or tape storage is *not* required for OAM. In fact, as you first begin to work with objects, it can be desirable to implement a pilot application that uses only disk. If and when optical or tape storage is included in your storage management scheme, one or a combination of the hardware configurations described in Table 3 on page 36 and Table 4 on page 45 can be used as a standard configuration.

Related reading: For more information and examples concerning standard tape library hardware configurations, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Deferring hardware ordering until after work load analysis

The task of ordering hardware should be deferred until after the OAM work load is analyzed and resource requirements are estimated (see "Estimating resource requirements" on page 71). Depending on the volume of objects to be processed and stored, you can order additional DASD (for DB2 and any zFS requirements), NFS file servers (if NFS is used), optical storage devices, or tape devices. An Object or Object Backup storage group can specify no more than eight optical libraries. Be sure to request appropriate documentation when you place the hardware order.

Related reading: For more information about OAM hardware-related issues, see Appendix A, "Sample optical hardware configurations," on page 385, and *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Software

Software analysis must address two distinct types of programs:

- System software—system control programs, transaction control programs, security packages, communication programs, database management systems, storage management systems, and similar global system products
- Custom applications—locally-written application programs, customized exits, site-specific macros, and other software unique to your installation

Additionally, OAM has software prerequisites that you must install prior to implementing OAM.

System software

OAM is a component of DFSMSdfp and uses DB2 extensively. CICS and IMS transactions, TSO/E programs, and MVS batch jobs can invoke the OSREQ application programming interface (API) to OAM.

Restrictions:

Consider the following restrictions when planning for the installation of OAM:

- In a classic OAM configuration, only one OAM address space can be active per z/OS system. In a multiple OAM configuration, there can be one tape library OAM address space and multiple object OAM address spaces.
- Only one DB2 subsystem application can be associated per OAM address space.
- Optical devices can only be directly accessed by the OAM address space where they are physically online; however, they can be indirectly accessed by other OAMs within an OAMplex.
- If in an OAMplex, the DB2 subsystems connected to the OAMs in the OAMplex must belong to the same DB2 data sharing group.

Related reading:

See *z/OS DFSMS OAM Application Programmer's Reference* for additional information on the OSREQ API.

To process objects greater than 256MB on the file system sublevel or on tape, the OAM address space uses 64–bit addressing and virtual storage above the 2G bar. Additional configuration could be required to specify a MEMLIMIT value that will provide virtual storage above the 2G bar to the OAM address space. For more information see "MEMLIMIT for OAM" on page 35.

Custom applications

By definition, custom applications are unique to your installation. The planning team is responsible for thoroughly investigating the installed software base to determine what, if any, custom application constraints apply for OAM implementation.

Software prerequisites

The following software is the minimum release level that must be installed to enable this release of OAM:

- Any currently supported release of DB2 for z/OS. See IBM DB2 for z/OS (www.ibm.com/analytics/us/en/technology/db2/db2-for-zos.html) for information on the currently supported versions of DB2 for z/OS.
- If the OSREQ macro is invoked from a CICS transaction, then IBM CICS Transaction Server 2.2 or higher is required:
- If the OSREQ macro is invoked from an IMS application, then one of the following is required
  - IBM IMS/VS Version 2
  - IBM IMS/ESA® Version 4 Database Manager

Software can be ordered as soon as you have determined which, if any, of the required components must be added or upgraded. Remember to request supporting documentation when you place the software order.

If you are adding significant new system software along with OAM, consider phasing the installations. Install and test one product at a time; then, when the system is stabilized, add another product. This approach simplifies error diagnosis by limiting the number of potential problem areas.

Related reading: For more information about the software requirements for DFSMS, see *z/OS Migration*.

Estimating resource requirements

To fully assess the potential impact of implementing OAM, you must translate the work load estimates from the business analysis phase into resource requirements. This section provides some guidelines for evaluating resource needs and formulas for estimating space requirements for DASD, tape, and optical storage.

After the estimating process is complete, you can evaluate the capacity of the current processing environment and prepare to increase it, if necessary.

Attention: The formulas, constants, and performance rates used in this section are only for estimating purposes. They do not constitute benchmarks or guarantees and are provided solely as guidelines to assist you in your planning effort. They should not be interpreted as estimates for operation under normal work loads.

General requirements

Estimating resource requirements is more of an art than a science. Although rules of thumb and generalized algorithms are useful, each installation's needs and environment are unique. To optimize the estimating effort, each planning team should allow a reasonable margin of error and be alert to the subtle interactions that can affect performance, throughput, and resource requirements. The following comments are included to stimulate your analysis during the estimation process:

- Object size and activity level influences virtually all aspects of resource utilization. The validity of the remaining resource estimates depends on the accuracy of these fundamental assumptions.
- When estimating object transfer rate and other throughput issues, remember that OAM is part of a complex system, influenced by many factors.
- Application design can have a significant effect on OAM efficiency. For example, careful use of buffering can reduce virtual storage requirements.
- When estimating the amount of time needed to complete a storage management cycle, consider the following:
  - All data movement resulting from storage class changes, whether caused by class transition or application request, is handled during the storage management cycle.
  - Movement from disk to optical or tape is typically faster than the reverse.
  - The frequency and volume of backup copies of objects that are made during the cycle can increase processing time.
  - The effective drive transfer rate is faster for a single large object than for multiple small objects.
- In addition to estimating obvious needs, such as DASD, tape, and optical storage space, consider possible requirements for resources that are less directly related to the OAM installation. For example, do you have enough of the following:
  - Trained support personnel, application developers, and end users?
  - User IDs with the correct access authority?
  - Tapes to process backups for the DB2 tables?
  - Shelf space for shelf-resident optical disks and library-resident tape volumes associated with manual tape libraries?

- Shelf space for shelf-resident tape volumes used with stand-alone tape devices?
- Customized transport classes (CTCs) for OAM XCF processing to decrease any possible impact on cross-system processing based on locality of hardware versus the system originating the request.
- DB2 threads and locks to process concurrent access to tables during an OSMC cycle?

DASD storage

Implementing OAM requires a significant amount of DASD storage. DASD space is required for the object databases and for the OAM administration database. This section provides techniques for calculating DASD space requirements for these databases.

In addition to the space requirements detailed here, consider DASD space needed for the OAM configuration database as well as OAM-related system and application programs. See "DASD resources" on page 87 for an example of how these computations are applied.

System paging

The various subcomponents of OAM make significant use of virtual storage resulting in a high auxiliary storage requirement. In most instances, an installation uses one or more dedicated DASD volumes for paging so that there is sufficient auxiliary storage. If after system tuning low paging activity occurs, you can allocate other low-activity data sets to a paging volume.

Attention: A minimum of 210 MB (290 cylinders on a 3390 DASD) of auxiliary storage is required for OAM local paging. Other applications that use OAM might require additional auxiliary storage.

OAM exploits 64–bit virtual storage when objects greater than 256MB are written to or read from the file system sublevel or tape. If your installation will implement objects greater than 256MB on the file system sublevel or tape, you must plan for the significantly increased storage utilization on your system. For more information, see "Auxiliary storage and real storage considerations" on page 69.

Object databases

Each Object storage group has one object database. Each object database requires eight separate VSAM linear data sets. These data sets are for the object directory and its three indexes, the small object table and its index, and the large object table and its index. These data sets are explicitly allocated to give you control over data set placement and size. This control allows you to take full advantage of your configuration to optimize system and DASD I/O performance. (See Figure 68 on page 484 for a diagram of an object database.)

Before attempting to estimate your DASD requirements, you must have completed the business analysis process by establishing the following:

- Object grouping for all objects to be handled by OAM
- Object sizes within each Object storage group
- Number of objects to be stored in DB2 databases for each Object storage group

Tip: Object Backup storage groups do not use the object database. Backup information for each object is kept in the object database for the primary Object storage group.

Data set name qualifiers

Table 7 on page 72 provides descriptions and data set name qualifiers for each of the required data sets in the database.

Table 7. Data set name qualifiers and descriptions for each

Qualifier	Description
OSMDTS	Object directory table; information about the object
OSMLBTS	LOB base table; LOB specific information about the object.

Table 7. Data set name qualifiers and descriptions for each (continued)	
Qualifier	**Description**
OTLOBX1	LOB base table index 1; index (collection ID/object name) into the LOB base table
OSMLATS	LOB auxiliary table; storage area for objects designated as LOB
OTLOBAX1	LOB auxiliary table index1; index (ROWID) into the LOB auxiliary table to navigate to LOB data
OBJDIRX1	Object directory index 1; cluster index (odcreats) into the object directory table
OBJDIRX2	Object directory index 2; index (pending action date) into the object directory table
OBJDIRX3	Object directory index 3; index (collection name/object name) into the object directory table
OSMOTS04	4 KB small object table; storage area for objects with a length less than or equal to 3980 bytes (small objects)
OBJT04X1	4 KB object table index 1; index (collection ID/object name) into the 4 KB object table
OSMOTS32	32 KB large object table; storage area for objects with a length greater than 3980 bytes and less than or equal to 256 MB (268,435,456 bytes).
OBJT32X1	32 KB object table index 1; index (collection ID/object name/segment name) into the 32 KB object table
Note: Kilobytes = 1 024 bytes.	

Variables

Table 8 on page 73 describes the variables used in the DASD formulas:

Table 8. Variables used in DASD formulas	
Variable	**Description**
nt	Total number of objects stored within an Object storage group, which includes objects stored in the small object table, objects stored in the large object table, and objects stored directly on optical or tape storage
ns	Maximum number of objects stored in the small object table
nl	Maximum number of objects stored in the large object table
nlob	Maximum number of objects stored in the LOB storage structure
aos	Average object size
b	Number of 4-kilobyte pages per track on the device (see Table 9 on page 73)
cb	Number of 32-kilobyte blocks per cylinder on the device (see Table 9 on page 73)

Constants

Table 9 on page 73 summarizes the constants that are related to device type.

Table 9. DASD device characteristics		
Device Type	**3390**	**3380**
Pages per track (b)	12	10
Blocks per cylinder (cb)	22	19
Tracks per cylinder	15	15

Formulas

Table 10 on page 74 provides calculations for estimating DASD requirements for the eight object data sets.

These formulas do not include any significant free space. Calculate these formulas using the absolute maximum number of objects anticipated, plus whatever additional free space that you require.

The results of the formulas are expressed in tracks on DASD, which can be converted to cylinders using the information in Table 9 on page 73. The exception is the large object tables, (OSMOTS32 and OSMLATS), where the result of the calculation is expressed in cylinders.

The results of all these calculations must be rounded up to the next higher integer.

Table 10. Example of calculations for determining DASD requirements

Object directory table or index	**Calculation**
Directory table (OSMDTS)	*nt* ÷ (23 x *b*)
Directory index 1 (OBJDIRX1)	*nt* ÷ (260 x *b*)
Directory index 2 (OBJDIRX2)	*nt* ÷ (166 x *b*)
Directory index 3 (OBJDIRX3)	*nt* ÷ (68 x *b*)
4 KB small object table (OSMOTS04)	If *aos* > 1900 bytes: (*ns* ÷ *b*) x 1.1 *If *aos* < 1900 bytes: (*ns* ÷ *b*) ÷ "floor" x 1.1 where "floor" is (4074 ÷ (*aos* + 61))
4 KB object table index 1 (OBJT04X1)	*ns* ÷ (68 x *b*)
32 KB large object table (OSMOTS32)	***nl* ÷ ((32 746 ÷ (*aos* + 63)) x *cb*) x 1.1
32 KB object table index 1 (OBJT32X1)	*nl* ÷ (65 x *b*)
LOB base table (OSMLBTS)	*nlob* ÷ (52 x *b*) x 1.1
LOB base index 1 (OTLOBX1)	*nlob* ÷ (68 x *b*)
LOB auxiliary table (OSMLATS)	*nlob* ÷ ((32 746 ÷ *aos*) x *cb*) x 1.1
LOB auxiliary index 1 (OTLOBAX1)	*nlob* ÷ (195 x *b*)

Note: *Where "floor" means round to the next smaller integer before dividing by *ns* ÷ *b*. All objects stored in this table must be less than or equal to 3980 bytes in length. For example:

"floor" = (4074 ÷ (1500 + 61)) = 2.6 (or = 2 when rounded down to the next lowest integer)

Example: (34 100 ÷ 12) ÷ 2 = 14 209 x 1.1 = 15 630

**If your average object size (*aos*) is less than 32 746 bytes in length, the value (*aos* + 63) must be rounded to the next higher multiple of 4 KB before using it in a formula. For example, 5 KB rounds up to 8 KB, 13 KB rounds up to 16 KB, and so on. Find the quotient of 32 746 ÷ (*aos* + 63) first. If the quotient is greater than 1, ignore any fractional remainder; if the quotient is less than 1, use the remainder as the result. Multiply the result by the value for *cb*, which yields the value of the divisor for the formula. Drop any fractional remainder from the divisor before dividing into the dividend *nl*. This quotient must be rounded to the next higher integer before being increased by the multiplier. The final result must be rounded to the next higher integer as well.

OAM configuration database

The OAM configuration database (CBROAM) contains configuration information related to the target destinations for objects including tape volumes, optical libraries, drives, slots, and volumes. CBROAM also identifies objects to be ultimately deleted by OAM from optical volumes and the file system. It is a DB2 database and consists of the following tables:

Library
: Contains one row for each optical library. The DB2 name of this table is `OLIBRARY`. There is a unique index on the library name.

Drive
: Contains one row for each optical drive, whether operator-accessible or library-resident. The DB2 name of this table is `DRIVE`. There are two indexes defined on the table; one is unique and one is not.

Slot
: Contains one row for each of the slots in a 9246 optical library. The DB2 name of this table is `SLOT`. There is a unique index defined on the slot name in combination with the library name.

  Note: The 9246 optical library is no longer supported.

Volume
: Contains one row for each optical disk volume. The DB2 name of this table is `VOLUME`. There is a unique index on the volume serial number.

Deleted-Objects
: Contains one row for each object waiting to be deleted from 3995 rewritable optical media. The DB2 name of this table is `DELOBJT`. There are two indexes defined on the table; one is unique and one is not.

Tape Volume
: Contains one row for each tape volume used by OAM for object storage. The DB2 name of this table is `TAPEVOL`. There is a unique index on the volume serial number.

File System Delete
: Contains one row for each object waiting to be deleted from the file system. The DB2 name of this table is `FSDELETE`.

Table 11 on page 75 provides DASD space recommendations for storage of the CBROAM tables.

Table 11. CBROAM space recommendations

Description	DB2 name	Primary space 3390 tracks	Secondary space 3390 tracks
Library Table	OCLIBTSP LNAMINDX	1 1	1 1
Drive Table	OCDRVTSP DNAMINDX DRIDINDX	5 1 1	1 1 1
Slot Table	OCSLTTSP SLIBINDX	1 1	1 1
Volume Table	OCVOLTSP VSERINDX	20 2	10 1

Table 11. CBROAM space recommendations (continued)			
Description	**DB2 name**	**Primary space 3390 tracks**	**Secondary space 3390 tracks**
Deleted-Objects Table	OCDELTSP DVOLINDX DELOINDX	100 10 100	10 5 10
Tape Volume Table	OCTVLTSP TVOLINDX	10 2	1 1
File System Delete Table	OCFSDTSP FSDTINDX	100 10	10 5

These recommendations allow:

- 216 optical library definitions
- 1 320 drive definitions
- 1 404 slot definitions for as many as twenty-two 9246 library definitions
- 5 040 volume definitions
- 39 600 objects waiting for deletion from rewritable optical media
- 5 280 tape volumes to be used for storing objects
- 34 900 objects waiting for deletion from the file system

If you do not use a particular level or sublevel of the OAM storage hierarchy, then you can use minimal space values for the tables related to that level or sublevel. For example, if the file system sublevel is not implemented in your installation, then use 1 for the primary and secondary space values for the File System Delete Table and the associated index.

If your installation requires more entry space or if you are using DASD storage that allows fewer than twelve pages per track, the above space recommendations might need to be increased.

OAM administration database

An additional database is needed for object management. This database is identified by the data set name qualifier OAMADMIN. Although specific calculations could be made for exact tracks needed based on the number of storage class names, management class names, and collection names used by your installation, experience has shown that the recommendations in Table 12 on page 76 should be adequate.

Table 12. OAMADMIN space recommendations			
Description	**Data set qualifier**	**Primary space 3390 tracks**	**Secondary space 3390 tracks**
Management Class ID Table	MCIND CBRMGTX CBRMGTY	1 1 1	1 1 1
Storage Class ID Table	SCIND CBRSTOX CBRSTOY	1 1 1	1 1 1

Table 12. OAMADMIN space recommendations (continued)			
Description	**Data set qualifier**	**Primary space 3390 tracks**	**Secondary space 3390 tracks**
Collection Name Table	COLIND CBRCLTX1 CBRCLTX2 CBRCLTX3	2 1 1 1	2 1 1 1

These recommendations allow the maximum 32 767 storage class names, the maximum 32 767 management class names and over 60 000 collection names.

Object storage on removable media

If your installation is going to use optical, tape, or both types of storage, you must estimate the amount of optical disk and tape cartridge storage that you need for your OAM implementation. Factors to consider should include the number of the following:

- Optical libraries required per day
- Optical disks, tape cartridges, or both required per year
- Shelf-resident optical disks, tape cartridges, or both
- Optical, tape, or both types of libraries
- Operator-accessible optical disk drives
- Tape stand-alone drives, ATLDSs, and MTLs, or a combination of these devices

This section provides techniques for calculating optical and tape space requirements based on these considerations. See “Optical resources” on page 91 for an example of how these computations are applied.

Constants

Use the information in Table 3 on page 36 and Table 4 on page 45 regarding storage slot and cartridge capacities as constants for DASD resource calculations for your optical and tape configurations.

Formulas

The following formulas can be used to calculate storage (optical, tape, or both) estimates.

Note: The term "cartridge" in the following formulas refers to optical disk cartridges and tape cartridges. If you are only using one type of medium within your installation, simply calculate according to the needs of your storage management policy. If you are using both optical and tape storage, consider cartridges needed for both media when calculating the formulas in Table 13 on page 77.

Table 13. Formulas for calculating storage requirements	
Consideration factor	**Calculation**
Cartridges per day	# cartridges required per day equals: # megabytes written per day ÷ # of megabytes per cartridge *where*: # of megabytes written per day equals: objects created per day *x* object size in megabytes.
Cartridges per year	# cartridges required per year equals: total # of cartridges per day *x* workdays per year

Table 13. Formulas for calculating storage requirements (continued)	
Consideration factor	**Calculation**
Shelf-resident cartridges	# shelf-resident cartridges equals: # cartridges required per year *x* retention period in years
Storage group adjustment	storage group adjustment equals: # of storage groups *x* # of active drives Therefore, total number of cartridges required per day equals: # cartridges required per day + storage group adjustment
Libraries for library-resident cartridges	# libraries required to hold library-resident cartridges equals: # of days library resident *x* (# cartridges required per day ÷ # slots in the library)
Note: # indicates a total number *x* indicates multiplication ÷ indicates division + indicates addition	

Cartridges per day

The number of objects created per day should include only those objects stored on optical, tape, or both media types. If several objects of different sizes are to be written to optical, tape, or both media types, calculate the number of megabytes written per day for each object size and sum the results to get the total number of megabytes written per day. See Table 13 on page 77 for more information on this calculation.

Attention: You need to determine what value should be assigned to the time periods within your calculations. For example, the term *day* could be a calendar day or a workday depending on the requirements of your business. A week can be either a seven-day calendar week or a five-day work week (or in some environments, this time frame can even be less). A year might include all the days of the year (including weekends and holidays) or it might only include the regular workdays for your installation. Include these factors in your calculations. Remember that OSMC does not recognize the difference between workdays, weekends, or holidays. Take this factor into account in your calculations for resource planning.

Each cartridge can contain objects from only one Object storage group. Therefore, on the first day that optical, tape, or both types of storage are used, you will need at least one cartridge for each Object storage group. If you plan to create backup copies of objects, remember to calculate the number of cartridges needed for each Object Backup storage group (use the *cartridges per day* formula in Table 13 on page 77). On the first day that your installation creates backup copies, you will need at least one cartridge of the appropriate media type for each Object Backup storage group.

Adjusting for storage groups and active drives

If multiple drives are to be used for writing objects from one or more Object or Object Backup storage groups, increase the number of cartridges required per day by following the calculation for *storage group adjustment* in Table 13 on page 77.

Effectively using optical volume space

The usage of optical volume space is affected by the size of the objects and how the objects are written: chained by the storage management cycle or unchained by direct write to optical media. Table 14 on page 79 provides an estimate of optical volume usage for media used with a operator-accessible optical drive.

Table 14. Effective optical volume usage for ibm optical disk media		
	Effective utilization	
Object size in KB	Data written by storage management cycle (chained)	Data written directly to optical volume (unchained)
40 000	100%	93%
20 000	100%	47%
10 000	100%	23%
5 000	100%	12%
4 000	100%	9%
3 000	84%	7%
2 000	56%	5%
1 000	28%	2%

Attention: In subsequent calculations, ensure that you use a value for the total number of cartridges *required-per-day* that is large enough to reflect your planned use of storage groups, multiple drives, and direct write to optical and or tape. The validity of those calculations depends on the accuracy of your total number of cartridges *required-per-day* estimate.

Related reading: For 3995 capacity information, see *3995 Introduction and Planning Guide*.

Cartridges per year

Use the calculations in Table 13 on page 77 for *cartridges-per-year* to determine the number of cartridges needed to satisfy your yearly medium requirements for your storage environment.

Shelf-resident cartridges

Use the calculations in Table 13 on page 77 for *shelf-resident cartridges* to determine the number of shelf-resident cartridges needed to satisfy your yearly medium requirements for your pseudo library environments.

Determining library requirements

The number of libraries required for an OAM implementation is influenced by several factors:

- Number of libraries required to hold library-resident cartridges.
- Number of libraries required to satisfy the maximum retrieval rate of objects on optical, tape, or both types of storage.
- Number of libraries required for the storage management cycle to complete within the allotted processing period.

The largest of the three numbers represents the number of libraries that you should plan to install. The following guidelines concerning libraries can help you evaluate your library needs.

Libraries for library-resident cartridges

To ensure that you correctly estimate the appropriate amount of libraries (optical, tape, or both) to hold all of your library-resident cartridges, see the calculation under *libraries for library-resident cartridges* in Table 13 on page 77.

Libraries for maximum retrieval rate

Each 3995 optical library can handle up to 200 mounts per hour and still provide an acceptable response time. If mount activity exceeds this rate, you might experience long delays on retrieval because of queued requests. To correct this situation, consider either installing an additional optical library or keeping more objects on DASD or tape.

Libraries for storage management cycle processing

Your installation should include in its regular schedule a period of time during which the storage management cycle can run. For example, you might execute the storage management cycle every day during third shift. During this time, OAM moves objects between optical disk volumes, tape volumes, and DASD. If this processing period is short, it might be necessary to install additional libraries to prevent contention caused by the following situations:

- Several Object storage groups are processed concurrently (controlled by the MAXS parameter).
- Multiple drives are used concurrently for a given Object storage group.
  - Optical drive usage is controlled by the DRIVE STARTUP THRESHOLD storage group parameter in ISMF.
  - Tape drive usage is controlled by the use of the TAPEDRIVESTARTUP (threshold in MB) keyword on the SETOAM statement for each Object storage group.

Related reading: For information about using objects effectively for IBM 3995 media, see the *3995 Introduction and Planning Guide*.

XCF resource estimation

To best use the resources of the cross-system coupling facility (XCF), you should first use the default transport classes and run Resource Measurement Facility (RMF) reports with XCF usage to determine if customization is needed. If you try to establish a configuration where the hardware is on the same system where the highest needs are for that library (storage group level, OSMC processing level, or user grouping), the cross-system overhead is reduced. OAM processing in an OAMplex increases XCF resource overhead with small messages used to communicate changes in the configuration during normal processing, and larger messages for object reads or writes that require cross-system processing.

Related reading: For more information, see "Using appropriate transport classes within XCF" on page 212 and *z/OS RMF User's Guide*.

Concluding resource estimation

After your resource estimations have been calculated, compare those projections with the resources you have available. Determine the additional resources that you need to use OAM, and develop a schedule for obtaining those resources. You can order some items prior to OAM installation; other items might not be needed until later, as your use of objects increases. Make sure your project plan includes time to order, install, and test essential resources before OAM is installed.

Planning to use a file system

Careful and complete planning is critical for a successful implementation of the file system capability within OAM. There are a number of considerations that must be coordinated between the Unix System Services environment and the OAM implementation.

A directory location within the Unix file system hierarchy can be associated with each object storage group in which the OAM file system support will be implemented. IBM recommends that a unique directory location be specified for each object storage group and that this directory location contains a mounted file system specifically for the storage group (similar to the unique DB2 object storage tables for the DB2 sublevel of the OAM storage hierarchy). This provides for object data isolation, may improve performance, facilitates file system backup and other maintenance, and allows for future growth. Each installation is unique and your planning must consider the maximum number of objects that can be contained within each storage group (which depends upon, and could be limited by, the underlying file system). The following considerations apply to the directory location under the assumption that a file system specific to the storage group is mounted at the directory location as recommended by IBM:

- The directory location specified must be exclusively for OAM usage. OAM will create subdirectories and files at this location, which should be treated as a "black box". No deleting, renaming, movement, or any other manipulation of these directories and files or their attributes should be performed. Additionally, no other user data should be stored in the Unix file system hierarchy at or beneath the specified directory location.
- The directory location (L2DIR) specified may contain symbolic links. This, however, will introduce a dependency upon the continued existence of the symbolic links for successful OAM operation. The contents of any symbolic links must be understood, recorded, and archived, in case the symbolic links are ever intentionally or accidentally removed or altered. OAM will create directories and files within the specified directory location and these OAM-created directory names and file names cannot contain any symbolic links and, if symbolic links are present, they will cause an error.
- A strategy must be developed to ensure that the OAM address space is stopped before Unix System Services is shut down. Your local procedures should be updated to incorporate this important activity. Failure to stop the OAM address space before shutting down Unix System Services can have unpredictable results, including abends and hangs, and can adversely affect OAM operation.

  Note: In a multiple OAM configuration, it is not necessary to stop the tape library OAM address space. Only object OAM address spaces must be stopped.
- A strategy must be developed to ensure that the file system is mounted whenever OAM is in operation to ensure that OAM can successfully write, read, or delete files within the specified directory location.
- For file system space planning, assume that the OAM file name itself can be up to 255 characters. This will consume larger amounts of disk space for each file's directory entry and should be taken into account when allocating space for file systems. Also take into account any other file system overhead for the file system you are using. For NFS, you must investigate the NFS server configuration to ensure that sufficient directory space is available for the OAM file name entries.
- In general, file system performance can degrade significantly when hundreds of thousands of files are stored in a single directory. For OAM file system planning, it is recommended that you plan to use multiple object storage groups to limit the number of objects stored in the file system sublevel for any given object storage group. Note also that to mitigate performance degradation, OAM may create up to 2053 individual subdirectories in any given directory, so ensure that this creation of at least 2053 subdirectories is supported by the file system you plan to use with OAM.
- A strategy must be developed to ensure that the file system will be backed up regularly and coordinated with the backup of other OAM data and metadata (such as the DB2 object storage tables and DB2 object directory tables). See *z/OS UNIX System Services Planning* for backup techniques, which include use of the Tivoli Storage Manager product, the DFSMShsm feature of z/OS, and the DFSMSdss feature of z/OS. You also will need to include the CBROAM*xx* PARMLIB member in your backup strategy, as it contains essential file system location information required for proper OAM operation.

Security configuration for the file system

To allow the OAM address space to access directories and files in the Unix file system hierarchy, the Security Server (RACF) or equivalent security product must be configured to provide both a Unix System Services group (with an associated group ID) and user (with an associated user ID) for the OAM started procedure. The following description assumes usage of the Security Server (RACF) and the examples assume the definition of the STARTED class which is the preferred method for assigning identities to started procedures such as the procedure that you use to start the OAM address space. If you currently have a group and user defined for the OAM started procedure, you will need to review your configuration to ensure it meets the criteria identified below, but minimally the group and user may need to be altered; the description below will outline the steps required to configure the Security Server (RACF) for the OAM started procedure.

The group should:

- Be unique and used only by OAM
- Consist of a single user used only by OAM, where this single user is the owner of the group
- Have an associated Unix System Services Group ID (GID) defined in the OMVS segment that is not assigned to any other group.

The user:

- Should result in a definition with the PROTECTED attribute (accomplished by specifying the NOPASSWORD attribute).
- Should have an associated Unix System Services User ID (UID) defined in the OMVS segment that is nonzero and not assigned to any other user.
- Can optionally include limits for OAM usage of Unix System Services in the OMVS segment.
- Must be connected to the unique group for OAM usage, be the only user connected to the group, and be the owner of the group.

The intent is that the user that is defined cannot be used to login to Unix System Services. Configuration for OAM within the Unix file system hierarchy will be performed by a superuser and after that point the OAM address space (by the OAM specific group and user) is the only entity that should have access to the directories and files containing OAM object data (other than a non-OAM backup mechanism, which typically relies upon superuser privileges to access directories and files and therefore membership in the OAM specific group is not required).

Complete the following steps:

1. Create or update the group specifically for OAM usage
2. Create or update the user specifically for OAM usage (optionally specify Unix System Services process level limits for PROCUSERMAX and FILEPROCMAX).
3. Connect the user to the group (if not already connected).
4. Create or update a resource profile to associate the name of the OAM started procedure with the OAM specific group and user

With this Security Server (RACF) configuration in place, when an operator starts OAM, the OAM procedure name is used to find a matching resource profile and the resource profile then provides the group and user to be associated with the OAM address space. This group and user (and associated group ID and user ID), as well as any optional process level Unix System Services limits defined for the user, will be applied to the OAM usage of Unix System Services. The group and user associated with the OAM address space will be displayed in message IEF695I.

Although a process level limit also exists to specify the CPU time that a process can use (MAXCPUTIME), this limit does not apply to the OAM address space. Instead, the OAM address space is defined by default in MVS as well as in the Program Properties Table (PPT) by member SCHED*xx* of PARMLIB to be a system task that will not time out so no CPU time limit will be applied to the OAM usage of Unix System Services

Table 15 on page 82 provides examples of the commands that can be used to complete the Security Server (RACF) configuration.

Table 15. Commands for completing the security configuration

Create	**Update**
Group	
ADDGROUP *oamgrp* OMVS(GID(*gid*))	ALTGROUP *oamgrp* OMVS(GID(*gid*))
User	
ADDUSER *oam* DFLTGRP(*oamgrp*) OWNER(*oamgrp*) NAME('OAM Address Space') NOPASSWORD OMVS(UID(*uid*))	ALTUSER *oam* OMVS(UID(*uid*))
or	or

Table 15. Commands for completing the security configuration (continued)

Create	Update
ADDUSER *oam* DFLTGRP(*oamgrp*) OWNER(*oamgrp*) NAME('OAM Address Space') NOPASSWORD OMVS(UID(*uid*) FILEPROCMAX(*fff*) PROCUSERMAX(*ppp*))	ALTUSER *oam* OMVS(UID(*uid*) FILEPROCMAX(*fff*) PROCUSERMAX(*ppp*))
Connecting User to Group (only if DFLTGRP was not originally specified on ADDUSER)	
CONNECT *oam* GROUP(*oamgrp*)	
Resource Profile for the OAM started task (assumes user name is "OAM")	
RDEFINE STARTED OAM*.OAM* STDATA(USER(=MEMBER) GROUP(*oamgrp*))	RALTER STARTED OAM*.OAM* STDATA(USER(=MEMBER) GROUP(*oamgrp*))
Activate Sharing of the Resource Profile	
SETROPTS RACLIST(STARTED) REFRESH	

Note: Although the Security Server (RACF) group and user can be different on each system within an OAMplex, the associated group ID in the OMVS segment must be the same on all systems in the OAMplex and the associated user ID in the OMVS segment must be the same on all systems in the OAMplex.

Preparing the physical environment

OAM itself does not require changes to the physical environment. However, if you are using optical storage subsystems or tape library subsystems for the first time, you can prepare for their installation.

Related Reading

- For detailed information concerning optical storage subsystems, see the *LAN Channel Station Installation and Test* and the *3995 Introduction and Planning Guide*.
- For information regarding the tape library dataservers, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Preparing for installation and customization

Once the conceptual groundwork for OAM has been completed, you are ready to install OAM and to translate the theoretical model into a functioning system.

Before you begin: Ensure that all hardware and software prerequisites have been met. For more information, see "Analyzing your processing environment" on page 68.

For the detailed procedure for installing OAM, see "High-level installation and migration checklists" on page 97.

To simplify the installation process, a library of sample jobs and other useful data sets (SAMPLIB) is shipped with the product.

Related reading: Appendix B, "Sample library members," on page 421 contains listings or prologs of many of the SAMPLIB members.

Planning to program applications for OAM

The *z/OS DFSMS OAM Application Programmer's Reference* describes the OSREQ macro, the programming interface provided by OAM. This document contains detailed information about programming applications which use OAM.

As you consider OAM programming applications, keep in mind the following items:

- Application design plays a significant role in OAM performance. For example, careful use of buffering can reduce virtual storage requirements.
- Application programs are responsible for synchronizing OAM-related DB2 databases (for example, using SYNCPOINT under CICS).
- The auto-delete installation exit can be programmed specifically for an application, as described in "Auto-delete installation exit (CBRHADUX)" on page 573.
- ACS routines must be programmed.
- Maintenance of the auto-delete installation exit and ACS routines is generally the responsibility of the storage administration team. However, a particular application might require modification to use the exit and the ACS routines.
- When objects are stored directly to the file system sublevel from an application program, the application must perform the DB2 "commit" within 24 hours of storing the object. Failure to do this will ultimately result in loss of object data stored in the file system.

Planning to administer OAM

Product implementation only *begins* with installation; your planning must also include preparation for ongoing administration of the product. The focal point of OAM administration is the storage administration team, which is responsible for the following tasks:

- Monitoring and maintaining the SMS configuration through ISMF
- Monitoring and maintaining DB2 databases
- Tuning OAM
- Establishing recovery procedures
- Destroying expired data

The success of an OAM implementation depends significantly on the quality of the support staff. The storage administration team should receive intensive training before OAM installation and encouragement to keep abreast of current technology through continuing education. Your IBM marketing representative can help you identify appropriate publications and training opportunities.

Related reading: Chapter 4, "Administering OAM," on page 183, provides tools and techniques for performing these functions.

Preparing to operate OAM

Daily operation of the OAM system is the responsibility of the operations staff. Operator tasks are explained in Chapter 5, "Operating OAM and OTIS address spaces and OSMC functions," on page 255. To ensure successful OAM operation, the operations staff should work closely with the storage administration team and the application team to coordinate support efforts. Be sure to update your installation's operating procedures manual to include OAM-related tasks.

Operators need to have in-depth knowledge about the hardware used by OAM, especially optical and tape storage devices. They should be encouraged to keep abreast of current technology through continuing education. Your IBM marketing representative can help you identify appropriate publications and training opportunities.

Related reading: For information concerning operator tasks related to the tape library dataservers, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

OAM planning case study

A large company is in the process of planning for OAM implementation. This case study documents part of their planning effort.

Object characterization

During the business analysis phase, the planning team members determined that they will be processing two types of objects, which they refer to as summary objects and detail objects. Summary objects are small; detail objects are considerably larger. On an average workday, about 10 000 summary objects and 10 000 detail objects are created.

Establishing performance objectives and availability requirements

The planning team analyzed data usage patterns to establish performance objectives and availability requirements for their two object types. Summary objects are used for 30 calendar days, and response time must be fast during that period; however, at the end of 30 calendar days, the objects are no longer used and might be deleted. Detail objects have a more complex life cycle. During the first seven calendar days after creation, detail objects are accessed frequently, and fast response is needed. After that time, retrieval frequency drops, as does the response-time requirement. Detail objects are rarely used after 180 calendar days, but the company is required by law to keep them on file for five years (1 825 calendar days).

Table 16 on page 85 summarizes the characteristics of the company's objects.

Table 16. Object characteristics

Characteristics	Detail objects	Summary objects
Object size in bytes	64 000	3 000
Number created per workday	10 000	10 000
Number of workdays that fast retrieval is required (less than 1-second response time)	7	30
Number of calendar days that medium retrieval is required (less than 20-second response time)	180	—
Number of calendar days from creation after which the object is rarely accessed	180	—
Number of calendar days from creation after which the object can be deleted (5 years)	1 825	30
Maximum retrieval rate required (objects per hour) for fast retrieval	1 000	2 000
Maximum retrieval rate required (objects per hour) for medium retrieval	100	—
Maximum retrieval rate required (objects per hour) for slow retrieval	10	—
Number of backup copies required	1	0

Establishing collections and constructs

The planning team established two collections, one for each type of object. They developed storage classes (SC) to reflect the three service levels indicated by the business analysis. They created management classes (MC) to correspond to the transition points in the objects' life cycles (that is, when they expire or when their performance objectives change). Then they determined which storage and management classes would be the default initial class assignments for each collection. Finally, specifications were drafted for the ACS routines that control an object's storage group, storage class, and management class assignments.

Also, during the resource estimation phase, the planning team concluded that the 10 000 detailed objects that are being stored daily through the workweek must be backed up for disaster recovery purposes. The data contained within the detailed objects is critical business data that must be recovered in the event of a disaster.

SC Name
: **Description**

FASTPERF
: SC for objects with high-performance requirements (less than 1-second response time preferred). This storage class was designed to be the default initial storage class assignment for both summary and detail objects.

MEDPERF
: SC for objects with medium-performance requirements (less than 20-second response time preferred). This storage class was designed primarily for detail objects for which high performance is no longer necessary.

LOWPERF
: SC for objects with low-performance requirements (more than 20-second response time acceptable). This storage class was designed primarily for detail objects that are rarely retrieved.

MC Name
: **Description**

EXP30
: Thirty calendar days from the date of creation, objects with this management class should be processed by the storage management cycle. The expiration attributes in the definition for this management class indicate that objects in this class can be deleted after 30 calendar days. EXP30 was developed primarily for summary objects and is their default initial management class assignment.

TRAN7
: Seven calendar days from the date of creation, objects with this management class should be processed by the storage management cycle. This class was developed primarily for detail objects. After seven calendar days, the performance objective for detail objects changes from fast to medium, and a new storage class assignment is needed. TRAN7 is the default initial management class assignment for detail objects.

TRAN180
: On the 180th calendar day from the date of creation, objects with this management class should be processed by the storage management cycle. This class was developed primarily for detail objects. After 180 calendar days, the performance objective for detail objects changes from medium to low and a new storage class assignment is needed. TRAN180 should be assigned to detail objects that have a storage class assignment of MEDPERF.

EXP1825
: Five years (1825 calendar days) from the date of creation, objects with this management class should be processed by the storage management cycle. The expiration attributes in the definition for this management class indicate that objects in this class can be deleted after five years. EXP1825 was developed for detail objects and should be assigned to all detail objects that have a storage class assignment of LOWPERF.

Tip: Management class definitions include parameters other than those shown in this case study (for example, backup requirements and retention period for an object since last use). Review all of the parameters before you attempt to classify the objects in your applications.

Coding ACS routines to implement class transitions

The company's technical staff was able to code ACS routines that use these classes, along with the installation's object-naming conventions, to implement class transitions. During every storage management cycle, OSMC checks each object that is scheduled for storage management processing to see if it is due for expiration. If it is, OSMC deletes it; otherwise, the management class assignment is used to determine if class transition is needed. For class transition, the ACS routines are invoked.

The general logic of the storage management cycle and the ACS routines is as follows (for objects created on day *X*):

1. For definitions of collections:
   - If the collection name is detail and the object name is Null, then SC=FASTPERF and MC=TRAN7.
   - If the collection name is summary and the object name is Null, then SC=FASTPERF and MC=EXP30.

   These default values are stored in collection name entry in the DB2 Collection Name Table when the first object is stored to that collection. When neither storage class nor management class is specified on a request to store an object into one of these collections, the object is assigned the default classes associated with that collection.
2. At *X* + 7 calendar days, objects in MC=TRAN7 are processed by the storage management cycle. MC=TRAN7 does not specify that these objects should be deleted, so the ACS routines are invoked. If an object's name begins with *D* (the naming convention for detail objects) and has SC=FASTPERF, the ACS routines change the class assignments to SC=MEDPERF and MC=TRAN180. This change in storage class can cause the object to be relocated in the object storage hierarchy. For example, the object might move from DASD to optical disk to a tape volume, or any combination therein.
3. At *X* + 30 calendar days, objects in MC=EXP30 are processed by the storage management cycle. The expiration attributes in the definition for MC=EXP30 specify that objects in this class should be deleted at *X* + 30 calendar days, so that the objects are deleted by OSMC.
4. At *X* + 180 calendar days, objects in MC=TRAN180 are processed by the storage management cycle. MC=TRAN180 does not specify that these objects should be deleted, so that the ACS routines are invoked. If an object's name begins with *D* (the naming convention for detail objects) and has SC=MEDPERF, the ACS routines change the class assignments to SC=LOWPERF and MC=EXP1825. The change in storage class are recorded in the object directory.

   You can conduct an analysis using DB2, SPUFI, or Query Management Facility (QMF) to determine which volumes in the configuration presently located within real libraries contain only objects with a storage class intended for the shelf or pseudo library. Those volumes found can be ejected from the library and placed in their assigned shelf location. The DB2 analysis will query the object directory table for each storage group required. Query the storage class table to determine the storage class identifier with which to qualify queries.
5. At *X* + 1825 calendar days, objects in MC=EXP1825 are processed by the storage management cycle. The expiration attributes in the definition for MC=EXP1825 specify that objects in this class should be deleted at *X* + 1825 calendar days, so that the objects are deleted by OSMC.

Resource estimation

Having completed the business analysis phase, the planning team used their object size and activity estimates to evaluate the DASD, tape, and optical storage resources that would be needed for OAM.

DASD resources

During the resource estimation phase, the planning team used the formulas in Table 8 on page 73 to determine their DASD storage needs.

Calculating DASD storage for an object storage database

The following example calculates the DASD storage needed for one object storage database. These calculations would be repeated for each object storage database within OAM. The specific values for the example calculations are based on the following conditions:

- During each workday 10 000 objects, each 3 000 bytes long, are stored in database GROUP00. A second case is also shown in the example calculation for objects averaging 1 500 bytes long.
- During each workday 10 000 objects, each 64 000 bytes long, are stored in database GROUP00. A second case is also shown in the example calculation for objects averaging 9 000 bytes long.
- The 3 000-byte objects that have been in the database for 30 calendar days are deleted from the database.
- The 64 000-byte objects that have been in the database for seven calendar days are moved from the database to optical storage, where they will remain for five years (1 825 calendar days) before being deleted.
- New objects are stored in the database before any existing objects are deleted or moved.
- To allow for new objects exceeding the predicted maximum number, an extra 10% space contingency is added.
- The database for object storage resides on a 3390-type, DASD device.

Calculating the number of objects stored on DASD, optical, and tape

The planning team began by calculating the number of objects that will be stored on DASD, tape, and optical storage.

1. The maximum number of small objects stored on DASD is determined as the number of summary objects created daily (10 000) plus the number of summary objects already resident in the database (10 000 x 30 days) plus the 10% contingency. The maximum number of small objects stored on DASD is identified in the formulas as the variable *ns*.

   Calculate the value of *ns* as:

   ns = (10 000 + (10 000 x 30)) x 1.1 = 341 000
2. The maximum number of large objects stored on DASD is determined as the number of detail objects created daily (10 000) plus the number of detail objects already resident in the database (10 000 x 7 days) + the 10% contingency. The maximum number of large objects stored on DASD is identified in the formulas as the variable *nl*.

   Calculate the value of *nl* as:

   nl = (10 000 + (10 000 x 7)) x 1.1 = 88 000
3. The number of objects that are stored within this Object storage group, but not within the small or large object tables, is determined as the number of summary objects retained within this Object storage group minus the number of summary objects on DASD, plus the number of detail objects retained within this Object storage group, minus the number of detail objects on DASD. Neither calculation can have a result less than zero.

   a. The number of summary objects retained within this Object storage group but not stored in the small object table is calculated as the number of summary objects stored each day, times the maximum number of days the objects are retained, minus the maximum number of summary objects stored on DASD (value of variable *ns* above).

      Calculate the number of small objects on optical and tape storage as:

      os = ((10 000 x 30) x 1.1) - *ns* = 330 000 - 341 000 = 0

   b. The number of detail objects retained within this Object storage group but not stored in the large object table is calculated as the number of detail objects stored each day, times the maximum number of days the objects are retained, minus the maximum number of detail objects stored on DASD (value of variable *nl* above).

      Calculate the number of large objects on tape and optical storage as:

ol = ((10 000 x 1307) x 1.1) - *nl* = 14 377 000 - 88 000 = 14 289 000

c. The total number of objects retained on tape and optical storage on any given day is the sum of the number of summary objects on tape and optical (*os* above) plus the number of detail objects on tape and optical (*ol* above).

   Calculate the total number of objects retained on tape and optical storage as:

   ot = *os* + *ol* = 0 + 14 289 000 = 14 289 000

4. The total number of objects that need to be referenced in the GROUP00 database is the sum of the maximum number of small objects stored on DASD (*ns* above), plus the maximum number of large objects stored on DASD (*nl* above), plus the total number of objects retained on tape and optical (*ot* above). This value is used in the formulas as the variable *nt*.

   Calculate the value of *nt* as:

   nt = *ns* + *nl* + *ot*

   nt = 341 000 + 88 000 + 14 289 000 = 14 718 000

Calculate the storage needed for the object storage database data sets (see Table 17 on page 89).

Table 17. Storage calculations for the object storage database data sets

Table, index, or data set	Calculations
Object directory data set `GROUP00.OSMDTS`	tracks = *nt* ÷ (23 x *b*) example: 14 718 000 ÷ (23 x 12) = 53 326 cylinders = 53 326 ÷ 15 = 3555 This results in 53 326 tracks or 3555 cylinders
Object directory index 1 `GROUP00.OBJDIRX1`	tracks = *nt* ÷ (260 x *b*) example: 14 718 000 ÷ (260 x 12) = 4717 cylinders = 4717 ÷ 15 = 315 This results in 4717 tracks or 315 cylinders
Object directory index 2 `GROUP00.OBJDIRX2`	tracks = *nt* ÷ (166 x *b*) example: 14 718 000 ÷ (166 x 12) = 7389 cylinders = 7389 ÷ 15 = 493 This results in 7389 tracks or 493 cylinders
Object directory index 3 `GROUP00.OBJDIRX3`	tracks = *nt* ÷ (68 x *b*) example: 147 180 000 ÷ (68 x 12) = 18 037 cylinders = 18 037 ÷ 15 = 1203 This results in 18 037 tracks or 1203 cylinders
Small object table index `GROUP00.OBJT04X1`	tracks = *ns* ÷ (68 x *b*) example: 341 000 ÷ (68 x 12) = 418 cylinders = 418 ÷ 15 = 28 This results in 418 tracks or 28 cylinders
Small object table (object size 3000 bytes) `GROUP00.OSMOTS04`	tracks = (*ns* ÷ *b*) x 1.1 example: (341 000 ÷ 12) x 1.1 = 31 259 cylinders = 31 259 ÷ 15 = 2084 This results in 31 259 tracks or 2084 cylinders
Small object table (object size 1500 bytes) `GROUP00.OSMOTS04`	tracks = (*ns* ÷ *b*) ÷ "floor" (4074 ÷ (*aos* + 61)) x 1.1 "floor" = (4074 ÷ (1500 + 61)) = 2.6 (= 2 when rounded to down to the next lowest integer) example: (34 100 ÷ 12) ÷ 2 = 14 209 x 1.1 = 15 630 cylinders = 15 630 ÷ 15 = 1042 This results in 15 630 tracks or 1042 cylinders

Table 17. Storage calculations for the object storage database data sets (continued)	
Table, index, or data set	**Calculations**
Large object table index `GROUP00.OBJT32X1`	tracks = *nl* ÷ (65 x *b*) example: 88 000 ÷ (65 x 12) = 112 cylinders = 112 ÷ 15 = 8 This results in 112 tracks or 8 cylinders
Large object table `GROUP00.OSMOTS32`	cylinders = *nl* ÷ (32 746 ÷ (*aos* + 63) x 22) x 1.1 first calculating: (32 746 ÷ (64 000 + 63)) = approximately 0.511 example: 88 000 ÷ (0.511 x 22) = 7828 x 1.1 = 8611 This results in 8611 cylinders
Large object table (object size 9000 bytes) `GROUP00.OSMOTS32`	(32 746 ÷ (*aos* + 63)) results in: round 9063 up to next 4K boundary = 12000 32 746 ÷ 12 = 2.73 (= 2 when rounded down to the next lowest integer) example: 88 000 ÷ (2 x 22) = 2000 x 1.1 = 2200 This results in 2200 cylinders
LOB base table GROUP00.OSMLBTS	tracks = *nlob* ÷ (52 x *b*) x 1.1 example: 50 000 ÷ (52 x 12) x 1.1 = 88 cylinders = 88 ÷ 15 = 6 This results in 88 tracks or 6 cylinders
LOB base index GROUP00.OTLOBX1	tracks = *nlob* ÷ (68 x *b*) example: 50 000 ÷ (68 x 12) = 108 cylinders = 62 ÷ 15 = 5 This results in 62 tracks or 5 cylinders
LOB auxiliary table GROUP00.OSMLATS	cylinders = *nlob* ÷ ((32 746 ÷ *aos*) x *cb*) x 1.1 example: 50 000 ÷ ((32 746 ÷ 64000) x 22) x 1.1 = 4886 This results in 4886 cylinders
LOB auxiliary index GROUP00.OTLOBAX1	tracks = *nlob* ÷ (195 x *b*) example: 50 000 ÷ (195 x 12) = 22 cylinders = 22 ÷ 15 = 2 This results in 22 tracks or 2 cylinders

Table 18 on page 90 summarizes the storage space calculations.

Table 18. Sample storage space calculation results		
Database name qualifiers	**Tracks needed**	**Cylinders needed**
GROUP00.OSMDTS	53 326	3 555
GROUP00.OBJDIRX1	4 717	315
GROUP00.OBJDIRX2	7 389	493
GROUP00.OBJDIRX3	18 037	1 203
GROUP00.OBJT04X1	418	28
GROUP00.OSMOTS04 (object size 3000 bytes)	31 259	2 084
GROUP00.OSMOTS04 (object size 1500 bytes)	15 630	1 042

Table 18. Sample storage space calculation results (continued)		
Database name qualifiers	**Tracks needed**	**Cylinders needed**
GROUP00.OBJT32X1	112	8
GROUP00.OSMOTS32	—	8 611
GROUP00.OSMOTS32 (object size 9000 bytes)	—	2 200
GROUP00.OSMLBTS	88	6
GROUP00.OTLOBX1	62	5
GROUP00.OSMLATS	—	4 886
GROUP00.OTLOBAX1	22	2

It is important to remember that the object directory table has an entry for *every* object within that Object storage group. This directory includes objects in the 4 KB object table, in the 32 KB object table and in the LOB storage structure, plus objects in the file system sublevel and on tape and optical storage. The 4 KB object table, the 32 KB object table, and the LOB storage structure have only the objects that are stored on in the DB2 sublevel. When an object is stored directly in the file system or on tape or optical storage, an entry is created in the object directory, but the object itself is not stored in either the 4 KB or the 32 KB object tables or the LOB storage structure. For all objects stored directly on tape or optical storage, the calculations for small and large objects on tape or optical storage (see item "3" on page 88) will provide the values needed for the object directory space.

LOB=*x* specifies whether or not OAM is to exploit DB2 LOB support for large objects that exceed 32 KB (32,640 bytes).

LOB=A
: specifies that objects for all storage groups that exceed 32 KB will be stored in a LOB storage structure when stored to DB2. LOB=A indicates to OAM that the installation has created LOB storage structures and associated V_OSM_LOB_BASE_TBL views for ALL object storage groups defined in the ACDS. This will result in optimal performance when storing large (greater than 32 KB) objects to DB2 since OAM will not query DB2 to see if the LOB base table view exists. If the LOB base table view does not exist then the large object store will fail.

LOB=P
: indicates to OAM that the installation has created LOB storage structures and associated V_OSM_LOB_BASE_TBL views for a PARTIAL list of object storage groups defined in the ACDS.

If the object being stored is greater than or equal to 32KB, then LOB=P requires OAM to query DB2 to see if the LOB base table view exists for a given object storage group. If the LOB base table view does exist for a given object storage group, large objects are stored in the associated LOB storage structure. If the LOB base table view does not exist, large objects are stored in the 32KB data table.

Note: A LOB storage structure must be used for objects greater than 256M.

LOB=N
: specifies that objects that exceed 32 KB and less than or equal to 256 MB (268,435,456 bytes) are to be stored in a 32 KB data table when stored to DB2. If specified, Store Sequence processing (STOREBEG, STOREPRT, STOREEND) to DB2 will fail. This is the default option.

Optical resources

During the resource estimation phase, the planning team used the formulas in "Formulas" on page 77 to determine their optical storage needs. They used the object size and volume estimates that were developed during the business analysis phase (see Table 16 on page 85). The following assumptions were used for the calculation:

- Only the 10 000 detail objects would eventually be written to optical disk.
- The optical residence period is 180 calendar days from creation date minus the seven calendar days on DASD.

Table 19 on page 92 represents the optical resources calculated by the planning team for each of the media types.

Note: The free space available on double-density, quad-density, and 8*x*-density WORM platters might not match the formatted capacity. The hardware holds a certain percentage of sectors in reserve based on the media type.

Table 19. Optical resource requirements for library resident data

Media	**SD WORM[2]**	**SD REWR[2]**	**DD WORM[2]**	**DD REWR[2]**	**QD WORM**	**QD REWR**	**8x WORM**	**8x REWR**
MB of data to write per workday	625	625	625	625	625	625	625	625
Cartridge capacity for user data in KB	297 983	314 569	620 934 637 041[4]	637 041	1 122 469[1,4] 1 211 012[2] 1 273 011[2,4]	1 122 468[1] 1 273 011[2]	1 986 966[1,4] 2 319 786[2,4] 2 256 268[3] 2 526 904[3,4]	1 986 965[1] 2 319 786[2]
Disks used per workday	1.059	1.078	0.530	0.539	0.248	0.261	0.136[2] 0.125[3]	0.144[2] 0.132[3]
Calendar days the data is needed to be library resident	173	173	173	173	173	173	173	173
Cartridges needed to retain library resident data	184	187	92	94	43	45	24[2] 22[3]	25[2] 23[3]
Slots in the library	144	144	144	144	52 104 156 258	52 104 156 258	52 104 156 258	52 104 156 258
Libraries needed to hold library resident data	2	2	1	1	1	1	1	1

Note:

MB = megabyte (1 048 576 bytes)
GB = gigabyte (1 073 741 824) bytes
SD = single-density
DD = double-density
QD = quad-density
8x = 8*x*-density
REWR = rewritable media
WORM = write-once, read-many

[1]For 512-sector media
[2]For 1024-sector media
[3]For 2048-sector media
[4]For continuous composite WORM (CCW) media.
CCW media values are the same as for REWR media of the same density.

Estimating the number of libraries required for maximum retrieval rate

The next factor to consider in estimating libraries is the number of libraries needed to satisfy the maximum retrieval rate for objects on optical storage. The application work load (as estimated in Table 16 on page 85) has a maximum retrieval rate of 100 objects per hour. Even if each retrieval results in mounting a different disk, one library is sufficient to satisfy this requirement. If you are making backup copies of objects, you need to plan for additional optical disks to accommodate them. The megabytes of data written per day is related to objects assigned to a management class that specifies an auto backup value of *yes*.

Tape resources

To estimate the number of tape cartridges needed per workday, the planning team determines the amount of data to be backed up per workday. The amount of OAM data being backed up per workday is 572.2 MB (10 000 objects per day x 60 000 bytes per object = 600 000 000 bytes, which equates to 585 938 KB, or 572.2 MB).

The number of tape cartridges needed per workday is determined by the following factors:

- The amount of data written to tape per workday
- The type of tape cartridge
- The recording technology used on the tape cartridges
- The number of tape drives and tape volumes available on each system within an OAMplex, if writing backups to tape in a Parallel Sysplex. Also take into consideration the number of tape volumes needed if transitioning storage groups to tape.

Assuming the tape cartridges are filled to 100% of their estimated capacity, Table 20 on page 93 shows the number of tape cartridges of each type of recording format combination that would be needed daily.

Table 20. The number of tape cartridges needed per workday

Media type	MB written to tape per workday	Cartridge capacity for user data	Number of tape cartridges used per workday
Cartridge System Tape (18-track format)	572.2	200 MB	2.86
Cartridge System Tape (36-track format)	572.2	400 MB	1.43
Enhanced Capacity Cartridge System Tape (36-track format)	572.2	800 MB	.715
IBM High Performance Cartridge Tape (128-track format)	572.2	10 GB	.056
IBM Extended High Performance Cartridge Tape (128-track format)	572.2	20 GB	.028
IBM High Performance Cartridge Tape (256-track format)	572.2	20 GB	.028
IBM Extended High Performance Cartridge Tape (256-track format)	572.2	40 GB	.014
IBM High Performance Cartridge Tape (384-track format)	572.2	30 GB	.019
IBM Extended High Performance Cartridge Tape (384-track format)	572.2	60 GB	.009
IBM Enterprise Tape Cartridge (EFMT1 recording technology)	572.2	300 GB	.002

Media type	MB written to tape per workday	Cartridge capacity for user data	Number of tape cartridges used per workday
IBM Enterprise Extended Tape Cartridge (EFMT2 or EEFMT2 recording technology)	572.2	500 GB	.0011
IBM Enterprise Extended Tape Cartridge (EFMT3 or EEFMT3 recording technology)	572.2	1000 GB	.00056
IBM Enterprise Advanced Tape Cartridge (EFMT4 or EEFMT4 recording technology)	572.2	4000 GB	.00016

Table 20. The number of tape cartridges needed per workday (continued)

Chapter 3. Migrating, installing, and customizing OAM

This topic explains how to install OAM, customize it for your business and processing environments, and verify that the installation is complete and correct. First, decide on your installation scenario: Are you installing OAM for the first time, migrating to a new release of OAM, merging separate OAM systems into an OAMplex, moving an OAM system to another system, or adding OAM systems to an OAMplex? Use the following roadmap to guide you to the correct procedure for your installation scenario.

Subtask	Associated procedure (see . . .)
"Verifying hardware and software prerequisites" on page 95.	"Verifying hardware and software prerequisites" on page 95
"Preparing the processing environment" on page 96.	"Preparing the processing environment" on page 96
Installing OAM for the first time.	• Table 21 on page 96 • "High-level installation and migration checklists" on page 97
Migrating to a new version of OAM.	• Table 21 on page 96 • "High-level installation and migration checklists" on page 97
Moving OAM to another system.	"Procedures for moving OAM to another system" on page 179
Merging OAMs into an OAMplex.	"Merging OAMs into an OAMplex" on page 180
Adding OAM systems to an OAMplex.	"Adding OAM systems to an existing OAMplex" on page 180

In addition to these basic installation requirements, this topic also describes an optional auto-delete installation exit that you can use to further customize your installation.

To simplify the installation process, a library of sample jobs and other useful data sets (SAMPLIB) is shipped with the product. This topic includes instructions for using SAMPLIB; Appendix B, "Sample library members," on page 421 contains listings of many of the SAMPLIB members. Before running any SAMPLIB job, remember to change the JCL to reflect your installation's requirements (for example, accounting information and data set names).

Related reading: See *z/OS MVS JCL Reference* for additional information.

Note: Unless otherwise indicated, once you migrate to the current release, you do not need to do anything to migrate backward to the previous release level other than to run the BIND jobs. Releases prior to V2R3 do not support a multiple OAM configuration, so when migrating backward to a pre-V2R3 release, the configuration must be set to a classic configuration.

Verifying hardware and software prerequisites

Before installing OAM, you must first verify that the hardware and software requirements specified in "Hardware" on page 68 and "Software" on page 70 have been met. Ensure that all the prerequisites have been installed and thoroughly tested to verify that they operate correctly in your processing environment before proceeding with any other installation steps.

Preparing the processing environment

For OAM to communicate with the rest of the processing environment, you must alter system software to interface with OAM, which might entail these tasks:

- Changing DB2 installation parameters and modifying the computer facility resource manager (CFRM) policy if necessary
- Changing CICS installation parameters.
- Configure the z/OS Unix file system
- Modifying the installation exit to
  - Manage deleted objects
  - Notify when tape volumes are returned to scratch
- Changing system libraries.
- Creating DB2 databases for object tables and directories.
- Creating OAM configuration databases.
- Creating and binding DB2 packages.
- Creating OSR application plans, or creating LCS, ISMF, and OSR application plans.
- Creating OSMC application plans.
- Verifying DB2 installation.
- Defining user catalogs.
- IPLing the system.
- Specifying the SMS definitions and programs that OAM uses.

This topic presents procedures for each of these tasks.

Preparing for migration or installation

Upon completion of any migration prerequisites, you should proceed with the following installation and migration checklist, paying particular attention to the guidance that is directed at installations that have had OAM installed previously.

Important: The term *migration* means to upgrade from one version of DFSMS to this current version. This assumes that you were previously using OAM for storing objects and that you will continue to use OAM for object storage upon installation of this current version or release. The term *installation* indicates that you have not installed OAM in any of the previous releases of z/OS, and are therefore installing OAM for the first time with this version of the product.

Table 21 on page 96 outlines the steps required for a new installation or migrating up from a previous version of DFSMS. These steps are discussed in detail in this topic.

Table 21. Checklist for new installation or migration to a new release of OAM

Checklist steps	New installation (z/OS V2R3)	Migrate
Change DB2 installation parameters	P	V
Change Customer Information Control System (CICS) installation parameters	P	V
Configure the z/OS Unix file system	P	V
Modify the installation exit to: Manage deleted objects Notify when tape volume returned to scratch	P	V
Change system libraries	P	P

Table 21. Checklist for new installation or migration to a new release of OAM (continued)		
Checklist steps	**New installation (z/OS V2R3)**	**Migrate**
Create DB2 databases for object tables and directories	P	V
Create OAM configuration database (OCDB)	P	V
Run sample migration jobs as required	N	V
Create and binding DB2 packages	P	P
Create OSR application plans*	P*	P*
Create OSMC application plans	P	P
Create LCS, ISMF, and OSR application plans	P	P
DB2 installation verification	P	P
Initial program load (IPL) the system	P	P
Specify the SMS definitions and programs used by OAM	P	V
Perform the OAM installation verification program	P	P
Optionally, update OAM to support new tape devices	P	V
Run Grants to authorize users	P	V
***Perform only if NOT performing the "Creating the LCS, ISMF, and OSR application plans" on page 157 step.** **P = Perform** **V = Verify (perform if necessary)** **N = Do Not Perform**		

High-level installation and migration checklists

A number of steps are involved in the installation of OAM. A checklist to outline these steps and to assist in ensuring that all steps have been completed is provided. The detailed procedural steps follow this checklist.

Recommendation

Read this entire section first so you know what to expect in this installation. As you perform the installation, reread the appropriate segment for each step in the checklist prior to performing it. Use this checklist only as a guide, as it does not contain the detailed information that you need to perform each step correctly. See the given page numbers.

Requirements

Before proceeding with this installation checklist, you should have verified the prerequisite hardware and software requirements (see "Hardware" on page 68 and "Software" on page 70 for more information).

Related reading

If you already have OAM installed and are migrating to the current z/OS release, see *z/OS Migration* for the specific migration steps.

If OAM has been previously installed, you must carefully analyze the following steps for your particular installation. You will be given additional migration guidance for each step indicating that you need only review the step to ensure that you have completed it in your previous installation, or that you should perform the step regardless of whether you completed it in your previous installation, or that you should not perform the step. However, remember that each installation is unique, and you must carefully study these materials to ensure that you are taking the appropriate action for your installation environment.

1. "1 Changing DB2 installation parameters" on page 100
2. "2 Changing CICS installation parameters" on page 101

   Note: Perform these steps only if CICS is installed on your system.

   a. Update or create CICS PLT

b. Update CICS CSD

c. Update CICS SIT

d. Connect DB2 to CICS

e. Copy CBRICONN to DFHRPL

3. "3 Configure the z/OS Unix file system" on page 103

Notes:

- For a new installation, configure the file system. If you are migrating to a new release, you can reuse your existing file system.
- Perform this step only if you are using the file system for object support.

4. "4 Modifying the installation exit to manage deleted objects" on page 104

Notes:

- For a new installation, evaluate and implement the auto-delete installation exit. If you are migrating to a new release, you can reuse your existing auto-delete installation exit.
- Perform this step only if you are running OSMC for expiration processing.

5. "5 Changing system libraries" on page 104

a. Update IGDSMSxx PARMLIB member

Note: Do not perform this step if you are using only DB2 sublevel storage (no file system, optical volumes, or tape devices) and you are not running OSMC for expiration processing. This step is only necessary for a classic OAM configuration or to automatically start the tape library address space in a multiple OAM configuration.

b. Updating the IEFSSNxx PARMLIB member

c. Update SMFPRMxx PARMLIB member

d. Update PROGxx PARMLIB member

e. Update CONSOLxx PARMLIB member

f. Add the OAM object tape return to the MVS scratch volume exit routines to the PROGxx member

g. Create or Update CBROAMxx PARMLIB members.

Note: Perform this step only if you intend to start the OAM address space for object support. You need to start OAM in order to:

- store objects in the file system or on tape or optical devices
- delete objects from the file system and enable the file system delete task to delete objects from the FSDELETE table in DB2
- use OSMC
- join an OAMplex.

h. "5h Updating the PROCLIB" on page 146

1) Modify, if necessary, then run CBRIPROC SAMPLIB job.

2) Modify, if necessary, then run CBRAPROC SAMPLIB job.

Note: Perform this step only if you start the OAM address space.

3) Verify or create device numbers.

Note: Perform this step only if you are using optical devices.

6. "6 Creating DB2 databases for object tables and directories" on page 149

Note:

Remember to modify the SAMPLIB jobs for your installation JOB card requirements and DB2 subsystem name and to start DB2 before running the SAMPLIB jobs.

a. Add additional steps to the database creation jobs, if necessary:

(CBRIALC0 and CBRISQL0)

b. Modify the OAM data set allocation jobs:

(CBRIALC0, CBRIALCX, CBRIALCY)

Note: If DB2 data sets are being placed in an SMS storage group, you must properly prepare the environment (that is, create ACS routines and the source control data set, and so on).

c. Run the OAM data set allocation jobs:

(CBRIALC0, CBRIALCX, CBRIALCY)

d. Modify the OAM database definition jobs:

(CBRISQL0, CBRISQLX, CBRISQLY)

e. Run the OAM database definition jobs:

(CBRISQL0, CBRISQLX, CBRISQLY)

f. Modify OAM LOB data set allocation and definition job. (CBRILOB)

Note: Perform this step ONLY if you intend to enable LOB support.

g. Run the OAM LOB data set allocation and definition job. (CBRILOB)

Note: Perform this step ONLY if you intend to enable LOB support.

7. "7 Creating the OAM configuration database" on page 154

   a. Modify, if necessary, then run the CBRSAMPL SAMPLIB job (for first-time OAM installations).

   Note: Do not perform this step if you are using only DB2 sublevel storage (no file system, optical volumes, or tape devices) and you are not running OSMC for expiration processing. If you choose not to execute this step, then you need to modify the CBRPBIND job in SAMPLIB to validate the bind at run time (VALIDATE RUN) rather than validating the bind at bind time (VALIDATE BIND).

8. "8 Merging object tables and OCDB for an OAMplex" on page 154

   a. Run the CBRSMERG SAMPLIB job.

   Note: Do not perform this step at initial installation. Modify and run this step for migration purposes. This job should only be run if you plan to merge multiple OAM configuration databases (OCDB) into a single shared OCDB for an OAMplex.

   b. Run the CBRSG100 SAMPLIB job.

   Note: Do not perform this step at initial installation. Perform this step only when all of the following conditions exist: you are setting up an OAMplex; you currently have multiple OAMs running on separate MVS images in a sysplex; and you want to merge two or more separate OAMADMIN tables, object storage databases, or both.

9. "9 Creating and binding DB2 packages" on page 155

   a. Run the CBRPBIND SAMPLIB job.

10. "10 OSR application plans" on page 155

    a. Run the CBRIBIND SAMPLIB job.

    b. Run the CBRIGRNT SAMPLIB job.

    Note: Perform this step only if you are using only DB2 sublevel storage (no file system, optical volumes, or tape devices), and are not using OSMC for object expiration. If using a file system, optical devices, tape devices, or OSMC, then you need to perform "12 LCS, ISMF, and OSR application plans" on page 157 instead.

11. "11 OSMC application plans" on page 156

    a. Run the CBRHBIND SAMPLIB job.

    b. Run the CBRHGRNT SAMPLIB job.

Note: Do not perform this step if you are using only DB2 sublevel storage (no file system, optical volumes, or tape devices) and you are not running OSMC for expiration processing.

12. "12 LCS, ISMF, and OSR application plans" on page 157
    a. Run the CBRABIND SAMPLIB job.

       Note: Do not perform this step if you are using only DB2 sublevel storage (no file system, optical volumes, or tape devices) and you are not running OSMC for expiration processing.
    b. Run the CBRAGRNT SAMPLIB job.

       Note: Do not perform this step if you are using only DB2 sublevel storage (no file system, optical volumes, or tape devices) and you are not running OSMC for expiration processing.
13. "13 Verifying DB2 installation" on page 158
    a. Verify that all application BINDS have been run.
    b. Verify that all application plans have been authorized.
    c. Verify that all application plans have been created.
14. "14 IPL the system" on page 159
15. "15 Specifying the SMS definitions and programs used by OAM" on page 160
    a. Define the base SCDS.
    b. Define libraries and drives in the OAM configuration database.

       Note: You must perform this step at initial installation. During migration, you might optionally perform this step if you are adding or changing libraries or drives. This step is required only if you are using optical devices.
    c. Define Object and Object Backup storage groups.
    d. Define storage classes.
    e. Define data classes.
    f. Define management classes.
    g. Define and test ACS routines.
    h. Validate and activate the SMS configuration.
16. "16 Running the OAM Installation Verification Program for object support" on page 178, if you are storing objects.
17. "17 Adding new tape devices to the OAM object tape configuration" on page 179

1 Changing DB2 installation parameters

The following procedural steps provide details to assist you in the performance of the checklist steps from the "High-level installation and migration checklists" on page 97.

Evaluate and select appropriate values.

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

The following values are provided as guidance information in establishing a DB2 system for use with OAM. If you have other applications sharing a DB2 system with OAM, you should evaluate the following and select values appropriate for all applications.

Tip: These values are only suggestions and are given only for installation. You might need to change them for optimum performance.

LOB support

Modify user LOB Value Storage (LOBVALA) and System LOB Value Storage (LOBVALS) fields from the DB2 installation panel DSNTIP7, Sizes Panel 2, to establish proper limits for the amount of storage that can be used for storing LOB values.

Buffer pools and max connects

When you defined DB2 job DSNTIJUZ, you specified values for the following storage sizes installation parameters. Evaluate the values you specified and ensure that your selected values will provide optimum performance. Change the values as required.

```
MAX USERS         (NUMCONCR)  200
MAX TSO           (NUMCONTS)  100
MAX BATCH         (NUMCONBT)  100
MIN BP0 BUFFERS   (BUFMIN00)  200
MAX BP0 BUFFERS   (BUFMAX00)  300
MIN BP1 BUFFERS   (BUFMIN01)  200
MAX BP1 BUFFERS   (BUFMAX01)  300
MIN BP2 BUFFERS   (BUFMIN02)  100
MAX BP2 BUFFERS   (BUFMAX02)  200
MIN BP32K BUFFERS (BUFMIN32)   50
MAX BP32K BUFFERS (BUFMAX32)  100
```

Guideline: You can verify or modify the values using DB2 dialog installation or by DB2 Job DSNTIJUZ directly.

EDM pools

If the environmental descriptor management (EDM) function pool size is not large enough for the databases, DB2 errors are received. The following is suggested for EDM pool size:

```
EDM POOL      9 000    (20 storage groups)
EDM POOL     20 000   (100 storage groups)
```

Related reading: See the topic "Calculating EDM pool size" in IMS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPH2) for an explanation of EDM pool size.

DB2 group buffer pools

Add the buffer pool information from "Buffer pools and max connects" on page 101 or the buffer pools used for the OCDB, OAMADMIN database, object directories, and object storage databases to the CFRM policy. Perform this step only if you are setting up an OAMplex and data sharing environment.

Related reading: For more information on buffer pools, see IMS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPH2).

Date and time routines

OAM does not require that dates and times be in a particular format; however, OAM returns and displays dates and times in ISO format only. The following example shows the ISO date and time format returned by OAM:

```
YYYY-MM-DD
```

DB2 default encoding scheme

OAM packages for the product and application BINDS require the default encoding scheme to be EBCDIC. The ENSCHEME and APPENSCH defaults parameters are defined in DSNTIJUZ at installation time.

2 Changing CICS installation parameters

If you plan to run OAM under CICS, make the following changes to your CICS installation before using OAM:

2a *Update or create CICS PLT.*

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

- If you have a program list table (PLT) to specify programs to be executed in the post-initialization phase of CICS startup, add the following entry:

  ```
  DFHPLT TYPE=ENTRY,PROGRAM=CBRICONN
  ```

  This entry names the OSR CICS initialization load module (CBRICONN) and invokes CBRICONN when CICS is initialized.

- If you do not have a PLT to specify programs for execution in the post-initialization phase of CICS startup, use one of the following to generate your CICS PLT:

  ```
  DFHPLT TYPE=INITIAL,SUFFIX=xx
  DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
  DFHPLT TYPE=ENTRY,PROGRAM=CBRICONN
  DFHPLT TYPE=FINAL
  END
  ```

  Related reading: For additional information on DFHPLT, see CICS Transaction Server for z/OS (www.ibm.com/support/knowledgecenter/SSGMGV/welcome.html).

2b ***Update CICS CSD.***

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

If you are a new CICS Transaction Server (CICS TS) user, you must create and initialize a CICS System Definition (CSD) file. Define the OAM module CBRICONN and add the group to your CICS region group list.

```
DEFINE PROGRAM(CBRICONN) GROUP(xxx) LANGUAGE(ASSEMBLER) CEDF(NO)ADD GROUP(xxx) LIST(yyy)
```

For xxx, substitute the group name for your CSD OAM definitions (that is, OAM).For yyy, substitute the CICS region name for your OAM enabled CICS region (that is, CICSA).

Attention: CICS TS does not support macro definitions for the Processing Program Table (PPT) resource. If you are moving to CICS TS from an earlier release and you have macro level resource definitions for the PPT, you must migrate the resources to the CSD. For information on how to do this, see the topic "Getting started with resource definition" in CICS Transaction Server for z/OS (www.ibm.com/support/knowledgecenter/SSGMGV/welcome.html).

Related Reading: For more information on the CICS System Definition (CSD) file, see CICS Transaction Server for z/OS (www.ibm.com/support/knowledgecenter/SSGMGV/welcome.html).

2c ***Update CICS SIT.***

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

Add the following entry to the CICS system initialization table (SIT) to supply initialization parameters to CICS:

```
         PLTPI=vv,       *** POSTINITIALIZATION CONNECTION
```

where:

vv
: Specifies the suffix of the DFHPLT module

2d ***Connect DB2 to CICS.***

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

CICS TS provides Resource Definition Online (RDO) support for DB2 resources. Define the following objects using RDO.

```
DB2CONN to connect DB2 to CICS.
DB2ENTRY for plan CBRIDBS.
```

These objects describe the global attributes of the CICS DB2 connection and the relationship between CICS transactions and DB2 resources (including application plans and command processors). For information on how to create these objects using Resource Definition Online (RDO), see CICS Transaction Server for z/OS (www.ibm.com/support/knowledgecenter/SSGMGV/welcome.html).

Attention: CICS TS Version 1 Release 3 and above do not support the use of DSNCRCT macro definitions to create the Resource Control Table (RCT). If you are moving to CICS TS, use CICS RDO to support DB2 resources instead of using DSNCRCT macros.

2e ***Copy CBRICONN to DFHRPL.***

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

CBRICONN must be contained in a load library in the DFHRPL concatenation. CBRICONN is delivered in SYS1.LINKLIB. You might add SYS1.LINKLIB to your DFHRPL concatenation or copy CBRICONN into another load library in the concatenation. If you choose not to use SYS1.LINKLIB in your DFHRPL list, you must remember to upgrade the copy of CBRICONN every time the module in SYS1.LINKLIB is changed due to maintenance or a new release of OAM.

3 Configure the z/OS Unix file system

Before configuring a file system for use by OAM:

- It is essential that thorough planning has been completed
- The Security Server (RACF) configuration must be performed
- The physical file system (zFS aggregate or NFS export on an NFS file server) to be used by OAM must first be created

The physical file system must then be incorporated into the z/OS UNIX file system hierarchy and the location within the z/OS UNIX file system hierarchy to be used for a given storage group must be specified by additional OAM configuration. You may want to develop a planning worksheet to help with your OAM file system configuration.

You will then prepare the Unix file system hierarchy by completing the following steps for each object storage group in each OAM instance in which you will exploit the OAM file system support. These steps will be performed by a Unix System Services superuser, typically in a Unix shell:

1. Create a new directory in the Unix file system hierarchy that will function as a mount point for the physical file system (the path to this directory will later be associated with the OAM object storage group in the OAM configuration).
2. Mount the physical file system for the OAM object storage group at the newly created mount point. Develop a strategy to ensure that this file system continues to remain mounted for OAM usage.
3. Change the owner for the file system mounted at the mount point to the Security Server (RACF) user name for OAM
4. Change the group for the file system mounted at the mount point to the Security Server (RACF) group name for OAM
5. Change the permissions for the file system mounted at the mount point to only allow access to the Security Server (RACF) user name for OAM. Do this by using a value of '700' on the chmod command for this directory.
6. Create a new regular file within the mounted file system at the mount point. This file is referred to as a "sentinel" file by OAM and must be named `4oamonly`. The size and contents of the file are unimportant, however the presence of this file will be verified by OAM before performing file system operations to ensure that the file system is mounted.
7. Change the owner for the sentinel file to the Security Server (RACF) user name for OAM
8. Change the group for the sentinel file to the Security Server (RACF) group name for OAM
9. Change permissions for the sentinel file to only allow access to the Security Server (RACF) user name for OAM. Do this by using a value of '600' for the chmod command for this file.

10. In a later step, you will add a SETDISK statement to the CBROAM*xx* PARMLIB member that associates the Unix file system hierarchy location with the object storage group, as described in "5 Changing system libraries" on page 104.
11. In a later step, you will create or update any SMS storage classes to specify usage of the file system sublevel of the OAM storage hierarchy, as described in "15 Specifying the SMS definitions and programs used by OAM" on page 160.

Appendix C, "Example file system configuration for OAM usage," on page 481 contains an example of the required file system configuration steps for an object storage group.

4 Modifying the installation exit to manage deleted objects

For a new installation, evaluate and implement the auto-delete installation exit. If you are migrating to a new release, you can reuse the auto-delete installation exit.

Perform this step only if you are running OSMC for expiration processing.

One of the rules defined in the management class is the end of an object's life. OSMC can delete an object when its lifetime expires. An object can also expire through an explicit expiration date. If an object has an explicit expiration date, the explicit expiration date takes precedence over the management-class-defined lifetime. OSMC calls the auto-delete installation exit before it deletes any object. The auto-delete installation exit indicates by return code whether the object should be deleted. Also, the installation exit can record the deletion of an object so applications can be kept synchronized with the OAM object directory table. In an OAMplex, you should synchronize the instances of CBRHADUX across the OAMs to avoid one OAM deleting an object with the approval of the exit when there is another exit on another OAM that is set to deny the delete request.

Requirement: The sample auto-delete installation exit prevents objects from being deleted. If your previous installation of OAM relied on the sample auto-delete installation exit to allow objects to be automatically deleted, you *must* modify the code in this exit to continue automatic deletion. You also must modify the code in this exit to define or change your installation handling of deleted objects. For more information about the installation exit, see "Auto-delete installation exit (CBRHADUX)" on page 573.

5 Changing system libraries

After using SMP/E to install DFSMSdfp, change the system libraries using the following procedures. Some procedures are completed only if the installation uses optical storage. These procedures are identified in the text. Some procedures are completed only if the installation uses object tape storage. These procedures are also identified in the text. Unless otherwise noted, all other procedures must be completed.

Update PARMLIB.

5a ***Update IEFSSNxx PARMLIB member to initialize the OAM subsystems.***

Notes:

1. You can add an OAM subsystem without re-IPLing the system by using the SETSSI ADD,S=OAMx,I=CBRINIT command. In a multiple OAM configuration, additional SETSSI ADD statements can be issued. See *z/OS MVS System Commands* for information on using the SETSSI command.
2. Starting with z/OS V2R2, the SETSSI command can also be used to delete an existing subsystem. This command should not be used to delete the OAMx subsystem. Consider using RACF to ensure that the OAMx subsystem is not deleted.
   Starting with z/OS V2R3, the MODIFY OTIS,DELSUB,subsys command can instead be used to remove the specified subsystem.
3. In a classic OAM configuration, a single OAMx subsystem definition must be defined in the IEFSSN*xx* member of PARMLIB. The format of the subsystem definition is as follows:

```
SUBSYS SUBNAME(OAMx) INITRTN(CBRINIT) INITPARM('[TIME=x][,MSG=x][,OTIS=x][,UPD=x][,MOS=nnnn][,LOB=x][,QB=x][,DP=x]')
```

In a multiple OAM configuration, one or more OAM subsystem definitions can be defined in the IEFSSN*xx* member of PARMLIB. An OAM subsystem definition must exist for each OAM instance in the multiple OAM configuration. The format of the subsystem definition is as follows:

```
SUBSYS SUBNAME(OAMx) INITRTN(CBRINIT) INITPARM('D=xxxx[,TIME=x][,MSG=x][,OTIS=x][,UPD=x][,MOS=nnnn][,LOB=x][,QB=x]
[,DP=x]')
```

where:

TIME=*xxx*
: Specifies upon which time zone the timestamp value in the object directory in DB2 is to be based and also affects the time zone for CYCLE START and CYCLE END times defined within the storage group definition for each object and object backup storage groups:

- TIME=GMT specifies that the timestamp value in the object directory in DB2 is to be based on GMT.
- TIME=LOC specifies that the timestamp value in the object directory in DB2 is to be based on local time. This is the default.

Note: This keyword is only processed for the first OAM subsystem to initialize and that value is used for all OAM subsystems. If specified on any OAM subsystem other than the first one to initialize, it is ignored.

MSG=*x*
: Specifies the format for how the OAM message text appears:

- MSG=EM specifies that the message text is in mixed-case English. This is the default.
- MSG=EU specifies that the message text is in uppercase English.

Note: This keyword is only processed for the first OAM subsystem to initialize and that value is used for all OAM subsystems. If specified on any OAM subsystem other than the first one to initialize, it is ignored.

OTIS=*x*
: Specifies whether OTIS should wait for JES to completely initialize before OTIS is started:

- OTIS=Y specifies that OTIS will not start until JES is initialized.
- OTIS=N specifies that OTIS starts independently from JES. This is the default.

Note: This keyword is only processed for the first OAM subsystem to initialize and that value is used for all OAM subsystems. If specified on any OAM subsystem other than the first one to initialize, it is ignored.

UPD=*x*
: Specifies whether DB2 updates for the pending action date (ODPENDDT) and the last reference date (ODLREFDT) fields should be performed:

- UPD=Y specifies that the ODPENDDT and ODLREFDT fields should be updated on all OSREQ retrieves.

  Note: The ODLREFDT field is not updated for OSREQ CHANGE requests.

  This is the default.
- UPD=N specifies that the ODPENDDT and ODLREFDT fields should not be updated for any OSREQ RETRIEVE requests.

  Note: The ODLREFDT field is not updated for OSREQ CHANGE requests.

  `Restriction:` If you use UPD=N, you cannot base transition criteria on the time since last use parameter in the ISMF management class definition.
- UPD=C specifies that the ODPENDDT and ODLREFDT fields should be updated on all OSREQ RETRIEVE and on all OSREQ CHANGE requests.

  Note: Regardless of the setting for UPD=, the ODLREFDT field is not updated when:

  – OSREQ RETRIEVE results in RECALL being scheduled.

– OSREQ RETRIEVE of object currently in RECALL mode.

MOS=*nnnn*
Specifies the maximum object size limit in MB. Valid values are 50-2000. If this keyword is omitted, the maximum supported object size defaults to 50MB. The maximum object size is checked when objects are initially stored through the OSREQ programming interface and is not checked on subsequent retrievals, in case the keyword was omitted, or its value was changed.

LOB=*x*
Specifies whether OAM uses DB2 LOB support for large objects that exceed 32 KB (32640 bytes). LOB has the following options:

- LOB=A specifies that, for all storage groups, objects that exceed 32 KB are to be stored in a LOB storage structure when stored to DB2. LOB=A indicates to OAM that the installation has created LOB storage structures and associated V_OSM_LOB_BASE_TBL views for ALL object storage groups defined in the ACDS. This results in optimal performance when you want to store large objects(greater than 32 KB) to DB2, because OAM does not query DB2 to see if the LOB base table view exists. If the LOB base table view does not exist, the large object store fails.
- LOB=P indicates to OAM that the installation has created LOB storage structures and associated V_OSM_LOB_BASE_TBL views for a PARTIAL list of object storage groups defined in the ACDS.If the object being stored is greater than or equal to 32KB, then LOB=P requires OAM to query DB2 to see if the LOB base table view exists for a given object storage group. If the LOB base table view does exist for a given object storage group, large objects are stored in the associated LOB storage structure. If the LOB base table view does not exist, large objects are stored in the 32KB data table.

  Note: A LOB storage structure must be used for objects greater than 256M.
- LOB=N specifies that objects that exceed 32 KB and less than or equal to 256 MB (268,435,456 bytes) are to be stored in a 32 KB data table when stored to DB2. Stores will fail for objects that exceed 256 MB. This is the default option.

QB=*x*
Specifies whether an OSREQ QUERY request results in a call into the OAM address space to retrieve the backup retrieval order keys. This specification is at the global level and pertains to all OSREQ QUERY processing.

- QB=Y indicates that OSREQ QUERY requests result in a call into the OAM address space for each backup copy. The OSREQ QUERY returns a complete backup retrieval order key for each backup copy. If a backup copy does not exist, then the OAM address space is not called and the backup retrieval order key contains binary zeros. This is the default.
- QB=N indicates that OSREQ QUERY requests do not result in a call into the OAM address space for each backup copy. The backup retrieval order key contains binary zeros for each backup copy regardless if the backup copy exists or not.

DP=*x*
specifies the scope at which deletion-protection is enabled or disabled. If a given object storage group has deletion-protection that is enabled, then no objects can be deleted from that object storage group before the object's expiration date.

- DP=A indicates that deletion-protection is enabled for all object storage groups.
- DP=P indicates that deletion-protection is partially enabled. Specifically, deletion-protection is enabled only for object storage groups that have OAM Deletion Protection set to ENABLED in ISMF for the object storage group SMS construct.
- DP=N indicates that deletion-protection is not enabled for any object storage group. This is the default.

D=*x*
specifies the 4-character DB2 SSID or Group Attachment Name of the DB2 subsystem associated with this OAM subsystem in a multiple OAM configuration.

- If D= is not specified on the first OAM subsystem to initialize, OAM runs in a classic OAM configuration and no other OAM subsystems can be defined.

- If D= is specified on the first OAM subsystem to initialize, OAM runs in a multiple OAM configuration. Additional OAM subsystems up to the maximum allowed can be defined. All additional OAM subsystem definitions must also specify D=. For a Tape Library OAM subsystem, specify D=NONE.

Note: Each DB2 subsystem can only be associated with one OAM subsystem on each system. If either the SSID or Group Attachment Name for a DB2 subsystem is specified with D= on one OAM subsystem definition, no other OAM subsystem definitions on the same system should specify either the SSID or Group Attachment Name for that same DB2 subsystem.

- The TIME=, MSG= and OTIS= keywords are only processed for the first OAM subsystem (Tape Library or Object) to initialize. Those values are used for all OAM subsystems until the next IPL. Any specification of TIME=, MSG= or OTIS= on any other OAM subsystem definitions is ignored. Note that OAM subsystem definitions after the BEGINPARALLEL statement in the IEFSSN*xx* member can be processed in any order. To ensure that a particular OAM subsystem is initialized first, make it the first OAM subsystem definition and place it before any BEGINPARALLEL statement.
- An OAM Tape Library subsystem definition in a multiple OAM configuration utilizes only the D=, TIME=, MSG=, and OTIS= parameters. All other parameters are ignored.

OAMx is the name of the subsystem, and CBRINIT is the name of the OSR initialization module run at IPL time.

`Rule:` The SMS subsystem must be started before the OAMx subsystem is started to ensure that the SMS entry precedes the BEGINPARALLEL keyword in IEFSSN*xx*. The OAMx subsystem entry in the IEFSSN*xx* parmlib member can then follow the BEGINPARALLEL keyword. If the BEGINPARALLEL keyword is not being used, you must still ensure that the SMS subsystem entry precedes the OAMx subsystem entry. The OAMx entry defines the OAMx subsystem; you must add this entry even if you do not plan to start the OAM address space. To prevent a possible system abend, make certain that the subsystem name in the IEFSSN*xx* member is different from the name of the PROCLIB member that is used to start the OAM address space.

5b ***Update SMFPRMxx PARMLIB member.***

You might perform this step at initial installation. For migration, you might verify or perform this step if you determine that it has not yet been completed in your current environment.

1. Verify and update if necessary the SMFPRM*xx* PARMLIB member to ensure that it has been set up. For more information on this PARMLIB member, see “Changing SMF recording” on page 571.

5c ***Update PROGxx PARMLIB member.***

You might perform this step at initial installation. For migration, you might verify or perform this step if you determine that it has not yet been completed in your current environment.

5d ***Update CONSOLxx PARMLIB member.***

You might perform this step at initial installation. For migration, you might verify or perform this step if you determine that it has not yet been completed in your current environment.

The Action Message Retention Facility (AMRF) must be active for messages with descriptor code 3 (eventual action required) to be recalled by using the `D R,L,KEY=OAM` command after the messages have rolled off the z/OS console. Activate AMRF at IPL by setting keyword AMRF=Y in the CONSOL*xx* PARMLIB member.

5e ***Add the OAM object tape return to the MVS scratch volume exit routines to the PROGxx member.***

If you are using a tape management system other than DFSMSrmm, then you may need to add the module name of the installation exit routine provided by the tape management system with the OAM Object Tape Return to the MVS Scratch Volume dynamic exit point (CBRUXTVS_EXIT). For example, if your tape management system uses the DFSMShsm Tape Volume Exit (ARCTVEXT), you may specify the following EXIT statement in the PROGxx PARMLIB member:

- EXIT ADD EXITNAME(CBRUXTVS_EXIT) MODNAME(ARCTVEXT)
- STATE(ACTIVE)

You may associate multiple exit routines to the CBRUXTVS_EXIT dynamic exit point by specifying multiple EXIT ADD statements.

5f ***Create or Update CBROAMxx PARMLIB members.***

You must perform this step if you are storing objects in file system sublevels or on tape volumes, setting up an OAMplex, or customizing your OSMC or optical environments. Do ***not*** perform this step if you are using only DB2 sublevel storage (no optical volumes, tape devices, or file system sublevel), and if you do not start the OAM address space for processing objects.

This CBROAM*xx* member is invoked using the OAM=*xx* parameter in the OAM member of PROCLIB and is processed during OAM address space initialization. You must create or update the CBROAM*xx* member to use object tape storage, file system sublevels, or an OAMplex. You can specify optional parameters that OAM can use with tape storage, optical storage, OSMC, file system sublevels, and in an OAMplex. If you do not specify specific parameters, the defaults are used. You can include the CBROAM*xx* PARMLIB member in any data set that is specified in the concatenation list in SYS1.IPLPARM(LOAD*xx*).

Related reading: For information about using the CBROAM*xx* PARMLIB member in various environments, see the following topics:

- "SETDISK statements for file system sublevel" on page 108
- "SETOAM statements for object tape storage" on page 109
- "SETOPT statements for options" on page 129
- "SETOSMC statements for use in the OSMC environment" on page 138
- "SETTLIB statement for tape library settings" on page 13
- "OAMXCF statements in an OAMplex" on page 144
- "ONLYIF statements" on page 143

SETDISK statements for file system sublevel

Creating or updating the CBROAM*xx* PARMLIB member with SETDISK statements is required to use the file system sublevel within your OAM environment.

Use SETDISK to configure the file system sublevel of the disk level in the OAM storage hierarchy. For each object storage group in which a file system sublevel will be defined, a SETDISK statement must be specified to provide the file system type and the file system directory to be used for the storage group. SETDISK is specified only at the storage group level.

Note: The file system type and file system directory for the object storage group must be carefully selected as these are static values and cannot be changed. These values must remain available in the SETDISK statement so that OAM can successfully store and equally importantly to later retrieve objects from the file system at any point in the future for as long as the object is stored in the file system.

This syntax diagram shows the syntax for the SETDISK statement. For information regarding how to read syntax diagrams, see "How to Read Syntax Diagrams" on page xix.

SETDISK statement syntax

```
►► SETDISK ── STORAGEGROUP( name ── L2TYPE( ─┬─ ZFS ─┬─ ) ── L2DIR(location) ── ) ►◄
                                             └─ NFS ─┘
```

SETDISK parameters

STORAGEGROUP(*name*)

A required parameter that specifies disk related parameters for a specific Object storage group which is in the active configuration, and which was previously defined using the ISMF storage group application. This parameter on the SETDISK statement provides additional information beyond what was specified using ISMF for the Object storage group to which it pertains. For *name*, specify the name of an Object storage group. The disk level of the OAM storage hierarchy can only be used for the storage of the primary copy of an object, therefore an Object Backup storage group cannot be specified on the SETDISK statement.

L2TYPE

A required parameter that specifies the file system type for disk sublevel 2. There is no default value for L2TYPE. The following values, which must be specified in upper case as shown, are valid:

NFS

The file system directory to be used for this object storage group is within a Network File System (NFS).

ZFS

The file system directory to be used for this object storage group is within a z/OS File System (zFS).

Note: The hierarchical file system (HFS) is not supported by OAM.

L2DIR(*location*)

A required parameter that identifies the existing file system directory location within the Unix file system hierarchy to be used for this object storage group. There is no default value for *location*. The value for *location*:

- Must be from 1 to 30 characters in length
- Can contain the characters: a-z, A-Z, 0-9, '.' (period), '_' (underscore), '-' (hyphen), and '/' (forward slash)
- Is case sensitive
- Must be an absolute path beginning with / or a symbolic link. A symbolic link must resolve to a directory path with a maximum length of 100 characters
- May optionally end with a forward slash
- Is strongly recommended to be unique for each storage group on each OAM instance to facilitate future growth and file system maintenance by identifying the location where a unique file system dedicated to the storage group has been mounted.

The existence of the location is not verified during OAM initialization. Ensure that the file system containing the specified directory is mounted before OAM processing attempts to access the file system directory location.

The following is an example of a SETDISK statement for an object storage group named SG3, which specifies that the zFS file system to be used for this storage group has been mounted in the z/OS UNIX file system hierarchy at the directory location /myoam/sg3/:

```
SETDISK STORAGEGROUP(SG3 L2TYPE(ZFS) L2DIR(/myoam/sg3/))
```

The L2DIR and L2TYPE values can not be updated with the `F OAM,UPDATE` operator command. The L2DIR and L2TYPE values can be displayed with the `F OAM,DISPLAY,SETDISK` operator command.

SETOAM statements for object tape storage

A CBROAM*xx* PARMLIB member contains SETOAM statements. The CBROAM*xx* member is processed during OAM address space initialization to establish the tape-related values for the object tape storage. Creating or updating the CBROAM*xx* PARMLIB member with SETOAM statements is required to use object tapes within your environment. If the SETOAM statement does not assign TAPEUNITNAME values to the Object Backup storage groups, the backup copies of the objects are stored on optical media. If no SETOAM statements exist in the CBROAM*xx* member of PARMLIB, the objects are stored on optical media by default.

Because an installation might want to tailor its object tape storage for different initializations of OAM, multiple CBROAM*xx* members might be created. In addition, multiple SETOAM statements might be supplied in one CBROAM*xx* PARMLIB member.

Use the SETOAM statement to determine whether backup objects are stored on tape volumes when they are written to an Object Backup storage group. The CBROAM*xx* PARMLIB member might contain one or more SETOAM statements. If the same parameter is specified multiple times on the same SETOAM statement, the last occurrence of the parameter is accepted. If the same parameter is specified multiple times on different SETOAM statements, the last occurrence on the last statement is accepted. If any syntactical errors are encountered in processing the statements in the CBROAM*xx* member of PARMLIB, OAM issues a message, and the OAM address space fails to initialize.

Related reading: See "Using the UPDATE command to set SETOAM, SETOSMC, and SETOPT values" on page 366 for information on changing the SETOAM values dynamically, or on defining the values when the CBROAM*xx* PARMLIB member is not used at initialization.

Figure 13 on page 110 is an example of a CBROAM*xx* PARMLIB member that can be used as a sample for your installation. See the syntax diagrams that follow Figure 13 on page 110 for graphical depictions of the SETOAM statement. The descriptions of the keywords are found in the discussion of the SETOAM statement on "SETOAM keyword definitions for global level" on page 112.

```
SETOAM DATACLASS(INMTL)
      L2DATACLASS(VTSDC)
      DSNWITHSGNAME
      DEMOUNTWAITTIME(120)
      MAXTAPERETRIEVETASKS(2)
      MAXTAPESTORETASKS(2)
      MAXRECYCLETASKS(4)
      MOUNTWAITTIME(5)
      TAPECAPACITY(CST18 55555)
      TAPECAPACITY(CST36 88888)
      TAPECAPACITY(ECCST 99999999)
      TAPECAPACITY(ESOTERIC1 7654321)
      TAPECAPACITY(ESOTERIC2 77766655)
      TAPECAPACITY(ESOTERICn 2147483646)
      TAPEEXPIRATION(2035/165)
      TAPEFULLTHRESHOLD(4096)
      TAPEDISPATCHERDELAY(30)
      TAPERECYCLEMODE(GROUP)
      OAMSCRATCHSYNCH(ENABLED)
      ALLOCRETRYMINUTES(3)
      TAPESDB(LARGE)
      STORAGEGROUP(GROUP00
        DATACLASS(TAPEGRP)
        DEMOUNTWAITTIME(120)
        SGMAXTAPERETRIEVETASKS(2)
        SGMAXTAPESTORETASKS(1)
        TAPECOMPACTION
        TAPEDRIVESTARTUP(9999)
        TAPEEXPIRATION(2055/003)
        TAPEFULLTHRESHOLD(2048)
        TAPEPERCENTFULL(76)
        L2TAPEUNITNAME(3590-1)
        L2DATACLASS(ATL1)
        SGMAXRECYCLETASKS(1)
        TAPEUNITNAME(3490))
```

Figure 13. CBROAMxx PARMLIB member sample using SETOAM

The syntax diagrams below show the syntax for the SETOAM statement. The first diagram shows the syntax for OAM global level parameters. The second diagram shows the subparameters for the STORAGEGROUP keyword parameter. For information regarding how to read syntax diagrams, see "How to Read Syntax Diagrams" on page xix.

SETOAM
ALLOCRETRYMINUTES(*minutes*)
DATACLASS(*SMS data class*)
DEMOUNTWAITTIME(*seconds*)
DSNWITHSGNAME
L2DATACLASS(*name*)
MAXRECYCLETASKS(*nn*)
MAXTAPERETRIEVETASKS(*tasks*)
MAXTAPESTORETASKS(*tasks*)
MOUNTWAITTIME(*minutes*)
OAMSCRATCHSYNCH(*mode*)
PERCENTVALID(*nnn*)
STORAGEGROUP(*name* STORAGEGROUP subparameters)
TAPECAPACITY(*unitname kilobytes*)
TAPEDISPATCHERDELAY(*seconds*)
TAPEEXPIRATION(*YYYY/DDD*)
TAPEFULLTHRESHOLD(*kilobytes*)
TAPERECYCLEMODE(*expiration_mode*)
TAPESDB(SMALL | LARGE)

SETOAM statement syntax: OAM global level parameters

STORAGEGROUP subparameters

L2TAPEUNITNAME(*unitname*) TAPEUNITNAME(*unitname*)
DATACLASS(*SMS data class*)
DEMOUNTWAITTIME(*seconds*)
L2DATACLASS(*name*)
MAXTAPERETRIEVETASKS(*tasks*)
MAXTAPESTORETASKS(*tasks*)
SGMAXRECYCLETASKS(*nn*)
SGMAXTAPERETRIEVETASKS(*tasks*)
SGMAXTAPESTORETASKS(*tasks*)
TAPECOMPACTION | NOTAPECOMPACTION
TAPEDRIVESTARTUP(*threshold in megabytes*)
TAPEEXPIRATION(*YYYY/DDD*)
TAPEFULLTHRESHOLD(*kilobytes*)
TAPEPERCENTFULL(*percent*)

SETOAM keyword definitions for global level

ALLOCRETRYMINUTES(*minutes*)

An optional parameter that specifies the time, in minutes, that OAM retries dynamic allocation of a tape drive that failed because all online tape devices are allocated before issuing message CBR6400D.

If the Automated Tape Allocation Manager (ATAM) is in use, you might want to reduce the time that OAM makes retry attempts or to bypass retry processing entirely and issue CBR6400D immediately, so ATAM can start allocating a tape drive without having to wait for OAM retry processing. By automatically responding to allocation requests, ATAM reduces the need for operator intervention and enables more efficient use of tape drives. See *IBM Tivoli Automated Tape Allocation Manager for z/OS User's Guide*, SC23-9736, for more information on ATAM.

Valid values for *minutes* are 0, 1, 2, 3, 4, and 5. A value of 0 means no retry will be attempted and CBR6400D will be issued immediately. The default value is 5.

Note: This is a global specification only and cannot be specified at the storage group level.

DATACLASS(*name*)

An optional parameter that specifies the SMS data class to be used for Object and Object Backup storage groups that do not have their own DATACLASS specification on the STORAGEGROUP subparameter of the SETOAM statement. If you specify DATACLASS at the OAM global level, but not at the storage group level, this specification of DATACLASS applies to each of the storage groups with a corresponding SETOAM statement which do not explicitly specify a DATACLASS. Using the DATACLASS keyword on the SETOAM statement at the global level allows the installation to affect Tape Device Selection Information, and volume expiration date for those Object or Object Backup storage groups which do not have an explicit DATACLASS keyword in their STORAGEGROUP subparameter list. (See the discussion in "Understanding the data class construct" on page 32 for details on TDSI.) There is no global level OAM default DATACLASS.

In an IBM tape library environment, specify the SETOAM DATACLASS parameter using a data class to request the desired media type or recording technology. To request tape data encryption in the tape library environment and in the stand-alone environment, use DATACLASS to request an encryption recording format; you can also specify the encryption key labels and encoding mechanisms using data class. In addition, you can specify DATACLASS to take advantage of performance scaling or performance segmentation in an IBM tape library environment and also in the stand-alone environment. Use the performance options in the data class to improve the random retrieval rate of primary objects in Object storage groups. For objects written to Object Backup storage groups, you can specify a data class that does not specify performance scaling, then you can use the full capacity of the volume.

Note: Tape data encryption is supported on the 3592, starting with the 3592 Model E05. Performance scaling and performance segmentation is supported on all the 3592 models, starting with the 3592 Model J1A.

Recommendation: Do not allow ACS routines to assign or change the data class assignment of an OAM tape volume. The data class for OAM tape volumes is determined by the SETOAM statement of the CBROAM*xx* PARMLIB member at MVS scratch tape allocation. The SETOAM statement provides this information at the storage group level or at the OAM global level, and it best suits the requirements for the tape volume that is being allocated. Allowing ACS routines to alter this specification could create unexpected consequences (for example, no compaction of the data when the SETOAM statement specified compaction). It is your installation's responsibility to write the ACS routines so as to not alter the data class construct for OAM tape volumes.

DEMOUNTWAITTIME(*seconds*)

An optional parameter that specifies the time, in seconds, that OAM waits before demounting and deallocating a tape drive that OAM is currently not using. For *seconds*, specify a decimal number between 1 and 9999. When the time interval expires, OAM rewinds and unloads the currently mounted tape cartridge and demounts and deallocates the device. The default for this optional parameter is 120 seconds.

Some circumstances might affect how this parameter is enforced:

- If a new tape drive allocation request comes in and OAM has already used the maximum number of tape drives (MAXTAPERETRIEVETASKS + MAXTAPESTORETASKS), OAM ignores the DEMOUNTWAITTIME and deallocates this drive in order to allocate another.
- If OAM is canceled, a DEMOUNT occurs, and DEMOUNTWAITTIME is ignored.
- If a request to vary the drive offline is sent while the DEMOUNTWAITTIME is in effect for that drive in an OAM session, the drive cannot vary offline until the specified DEMOUNTWAITTIME elapses.
- If OAM finishes reading and writing to a tape before the DEMOUNTWAITTIME elapses, a demount, unload, rewind, or release of the allocated drive cannot take place until the specified DEMOUNTWAITTIME is complete.

DSNWITHSGNAME

An optional parameter that specifies that the object storage group name is to be appended to the OAM tape data set names (OAM.PRIMARY.DATA, OAM.BACKUP.DATA, OAM.BACKUP2.DATA). For example, if OAM receives a store request for a new object for storage group GROUP55, GROUP55 is appended as the low-level qualifier to the OAM.PRIMARY.DATA data set name: OAM.PRIMARY.DATA.GROUP55. The tape management system can parse the data set low-level qualifier (storage group) to determine from which pool a scratch volume should be selected for a mount request in the stand-alone (non system-managed tape) environment. By associating object storage group names with tape management scratch pools, specific media type volumes can be assigned to specific pool names and segregated, and thus, preventing a WORM volume request from being used for a rewritable volume request and vice versa.

Once DSNWITHSGNAME is specified, all new OAM object tape data set names for all tape storage groups will have the storage group appended; however, the data sets written in the original data set name format can be retrieved and if the tape is not filled, new objects will be written on these tapes until filled.

Note: The data set name format for each tape volume is tracked in the TAPEVOL table in the DSNFMT column. If the tape volume has a DSNFMT value of blank, then the data set name written on the volume is the original data set name format with no storage group name low-level qualifier or has no current OAM data set written on the tape. If a tape volume has a DSNFMT value of "G" for group, then the data set name written on the volume has the storage group name appended.
Since the optionally appended low level qualifier of the OAM tape data set name is accessible as filter criteria in the 'ALLOC' ACS routine environment and can be used to assist in the assignment of SMS constructs, please ensure that if the DSNWITHSGNAME option is enabled, ACS routines are also examined to ensure that the appended low level qualifier is handled appropriately.

L2DATACLASS(*name*)

An optional parameter that specifies the SMS data class to be used, when storing objects to tape sublevel 2 for Object storage groups that do not have their own L2DATACLASS specification on the STORAGEGROUP subparameter of the SETOAM statement. Tape sublevel is associated with the OAM sublevel parameter specified in the SMS storage class construct. If you specify L2DATACLASS at the OAM global level, this specification of L2DATACLASS applies to each of the Object storage groups that do not explicitly specify an L2DATACLASS. Using the L2DATACLASS keyword on the SETOAM statement at the global level allows the installation to modify Tape Device Selection Information and volume expiration date for those Object storage groups that do not have an explicit L2DATACLASS keyword in their STORAGEGROUP subparameter list. There is no global level OAM default L2DATACLASS.

Note: L2DATACLASS does not apply to Object Backup storage groups.

In an IBM tape library environment, specify the SETOAM L2DATACLASS parameter using a data class to request the desired media type or recording technology. To request tape data encryption in the tape library environment and in the stand-alone environment, use L2DATACLASS to request an encryption recording format; you can also specify the encryption key labels and encoding mechanisms using data class. In addition, you can specify L2DATACLASS to take advantage of performance scaling or performance segmentation in an IBM tape library environment and also in the stand-alone environment. Use the performance options in the data class to improve the random retrieval rate of primary objects in Object storage groups. For objects written to Object Backup storage groups, you can specify a data class that does not specify performance scaling, then you can use the full capacity of the volume.

Note: Tape data encryption is supported on the 3592, starting with the 3592 Model E05. Performance scaling and performance segmentation is supported on all the 3592 models, starting with the 3592 Model J1A.

Recommendation: Do not allow ACS routines to assign or change the data class assignment of an OAM tape volume. The data class for OAM tape volumes is determined by the SETOAM statement of the CBROAMxx PARMLIB member at MVS scratch tape allocation. The SETOAM statement provides this information at the storage group level or at the OAM global level, and it best suits the requirements for the tape volume that is being allocated. Allowing ACS routines to alter this specification can create unexpected consequences (for example, no compaction of the data when the SETOAM statement specifies compaction). It is your installation responsibility to write the ACS routines so as to not alter the data class construct for OAM tape volumes.

MAXRECYCLETASKS(*nn*)
Can be specified at the global level. The *nn* is the maximum number of MOVEVOL tasks that can be run concurrently by the RECYCLE function. Valid values for *nn* are 0 - 15. The default is 1 if no value is specified. A value of 0 indicates that no RECYCLE operations can be run at the storage group or global level.

MAXTAPERETRIEVETASKS(*tasks*)
An optional parameter that specifies the maximum number of tasks within the OAM address space that can concurrently read objects from tape. This parameter controls the maximum number of tape drives that can be concurrently allocated to the OAM address space for reading object data from tape. This parameter allocates tape drives for processing the following requests:

- OSREQ RETRIEVE requests, where the primary copy of the object being retrieved is stored on tape.
- OSREQ RETRIEVE requests, where VIEW=BACKUP or VIEW=BACKUP2 and the backup copy is on tape.
- Requests to read the primary copy of an object from tape during the OSMC storage management cycle.
- Requests to read a backup copy or the primary copy of an object from tape during the execution of the OSMC optical volume recovery utility.
- Single object recovery.
- OSREQ retrieve requests where the primary copy is on optical, but volume is not readable, access to backup is activated and the backup copy is on tape.

If one or more OAM applications are retrieving objects from multiple Object storage groups, and the primary copies of the objects are being retrieved from tape volumes, the number or tasks specified with this parameter should be greater than or equal to the maximum number of Object storage groups being read from concurrently. This parameter eliminates the need to constantly mount and demount tapes belonging to different Object storage groups to satisfy OSREQ RETRIEVE requests.

The default is 1, which allows at least one task to be attached for processing read requests from tape volumes. This default allows groups previously writing objects to tape, but no longer having an explicit SETOAM statement in the CBROAM*xx* member, to be able to retrieve their previously written objects if a CBROAM*xx* PARMLIB member was successfully processed during OAM initialization. A CBROAM*xx* PARMLIB member must be successfully processed during OAM initialization for OAM object tape storage to be active on the system.

MAXTAPESTORETASKS(*tasks*)
An optional parameter that specifies the maximum number of tape drives used for writing objects to tape volumes. Use this parameter to process the following:

- OSREQ STORE requests where the primary copy of the object is to be stored on tape.
- OSMC class transition processing where the primary copy of the object is to be stored on tape.

- Writing of backup copies of objects during the OSMC storage management cycle.
- Single object recovery.
- Producing backup copies if backups are directed to tape.
- MOVEVOL for backup copies if objects are directed to tape.

This parameter and the MAXTAPERETRIEVETASKS parameter control the maximum number of tape drives that can be concurrently allocated to the OAM address space. For *tasks*, specify a decimal number between 1 and 100. The number specified for *tasks* with the MAXTAPESTORETASKS parameter should be greater than the number of Object storage groups that OAM applications are using for storing objects to tape volumes. If the number is less than the number of Object storage groups for the OAM applications, OAM will be frequently mounting and demounting tape volumes belonging to different Object storage groups to satisfy OSREQ STORE requests. Also, if objects are being written to the Object Backup storage group during the storage management cycle for multiple storage groups, and the backup copies are being written to tape volumes, then the number of tasks specified with this parameter should encompass this activity. Specify the maximum number of concurrent storage groups by using MAXS= in the PARM field on the JCL EXEC statement in the OAM cataloged procedure. If the MAXTAPESTORETASKS parameter is not specified on any SETOAM statement, OAM sets the default to 1.

Guideline: You should verify that there are enough tape drives (specified in the TAPEUNITNAME keyword) available to handle the values assigned to the MAXTAPESTORETASKS and MAXTAPERETRIEVETASKS parameters of the SETOAM statement. How the drives are spread among the Object storage groups or the Object Backup storage group depends on the values assigned to the MAXTAPESTORETASKS, MAXTAPERETRIEVETASKS, TAPEDRIVESTARTUP, and the TAPEUNITNAME parameters for each Object and the Object Backup storage group. OAM drive allocations can exceed the value of MAXTAPERETRIEVETASKS (up to the total value of MAXTAPERETRIEVETASKS and MAXTAPESTORETASKS) due to DEMOUNTWAITTIME, even if no STOREs are being done. If there are not enough tape drives, and the recovery logic for allocation is not able to obtain a tape drive for this request, then the request fails. (See "TAPEDRIVESTARTUP(threshold in megabytes)" on page 127 for details on TAPEDRIVESTARTUP.)

MOUNTWAITTIME(*minutes*)

An optional parameter that specifies the time, in minutes, that OAM waits for a tape volume to be mounted. For *minutes*, specify a decimal number between 1 and 120. When this interval expires, OAM issues message CBR6405D to the operator asking if the tape volume can be mounted. If the response is "Y", OAM resets the timer for another *minutes* interval. If the next interval expires and the tape volume is still not mounted, the same message is sent to the operator. If the reply is "C" and the mount was for tape that was to be read, OAM ends the task, fails the request to retrieve the object, and the volume is marked as "lost". If the operator replies "C" to this message and the mount was for a tape volume that was the target of a nonspecific (grouped) write request, a message is issued, and the volume is marked "lost" in the OAM internal control blocks, and the write request is retried on another volume. This processing applies to stand-alone, automated tape library dataservers, and manual tape libraries. If you do not specify this parameter on any SETOAM statement, the OAM default is five minutes. If this is a mount for an MVS scratch tape, and retry is attempted, the request fails.

Recommendation: Issue the F OAM,DISPLAY,LOSTVOL command to determine the volume serial number of each lost volume, both tape and optical. See "Displaying volumes that have LOSTFLAG set" on page 352 for further information on this command.

OAMSCRATCHSYNCH(*mode*)

An optional parameter that specifies how OAM manages tape volumes that were introduced into the OAM inventory before z/OS V1R5 and are now being returned to OAM scratch status. For such tape volumes, the original unit name (OUNITNAM) and data class (DATACLAS) fields in the TAPEVOL table are blank. OAM needs to use the values stored in these fields during subsequent allocation of OAM scratch volumes. As a result, these tape volumes are not eligible for reuse when they are returned to OAM scratch status.

OAMSCRATCHSYNCH can be used to make such volumes eligible for reuse by enabling OAM to synchronize their OUNITNAM and DATACLAS fields with the UNITNAME and DATACLASS values associated with the storage group for the volume before it was returned to scratch.

Specify this parameter at the global level only.

Specify *mode* as either DISABLED or ENABLED:

DISABLED
OAM does not check for blank DATACLAS and OUNITNAM fields in the TAPEVOL table and does not alter those fields. OAM also does not modify the UNITNAME field in the TAPEVOL table that is associated with this tape volume. If you omit the OAMSCRATCHSYNCH parameter, DISABLED is the default.

ENABLED
OAM synchronizes blank DATACLAS and OUNITNAM fields in the TAPEVOL table that is associated with a tape volume that is being returned to OAM scratch status, as follows:

- If the DATACLAS field is blank, OAM sets DATACLAS to the same value as the DATACLASS that was associated with the storage group for the volume before it was returned to scratch. If the OUNITNAM field is blank and the value in UNITNAME is a generic device, OUNITNAM and UNITNAME are both set to the same value as the UNITNAME that is associated with the storage group for the volume. (Generic devices include the 3480, 3480*x*, 3490, or 3590-1.)
- If OUNITNAM is blank and the value in UNITNAME is an esoteric device name, both OUNITNAM and UNITNAME remain unchanged.

PERCENTVALID(*nnn*)
You can only specify this keyword at the global (all storage groups) level. *nnn* represents the global default percentage of valid data threshold that is used to determine whether a full tape volume is a candidate for RECYCLE processing. This SETOAM value is used only if the optional PV= keyword is not specified on the RECYCLE command. The PERCENTVALID value that is specified on the RECYCLE command takes precedence over the PERCENTVALID value in the SETOAM statement. Valid values for *nnn* are 0 - 100. The default is 0 if no value is specified.

STORAGEGROUP(*name*)
An optional parameter that specifies tape related parameters for a specific Object or Object Backup storage group which is in the active configuration, and which was previously defined using the ISMF storage group application. This parameter on the SETOAM statement provides additional information beyond what was specified using ISMF for the Object or Object Backup storage group to which it pertains. At times, the information overrides what was specified using ISMF. For example, if the Object Backup storage groups has a valid SETOAM statement, the backup copies of objects are written on tape regardless of the optical disk libraries supplied in the ISMF definition of the Object Backup storage group. For *name*, specify the name of an Object or Object Backup storage group. The parameter listed below applies to the name of the Object or Object Backup storage group that you specify.

This parameter can take a number of subparameters. See "SETOAM keyword definitions for STORAGEGROUP subparameters" on page 121 for descriptions of the STORAGEGROUP subparameters.

TAPECAPACITY(*unitname kilobytes*)
An optional parameter that specifies a unit name with a numeric value in kilobytes from 1 to 2 147 483 646. This parameter allows you to specify a tape capacity for tapes written using a general recording technology, as well as a different capacity for tapes written to drives associated with esoteric unit names. This parameter indicates the tape capacity desired for the three general specifications (CST18, CST36, or ECCST) and esoteric unit name specifications. All esoteric unit name specifications are verified as valid esoteric unit names that are defined to the system. The tape capacity specified on this parameter becomes the value at which OAM considers a tape volume filled.

If this parameter is not specified in the SETOAM statement, the following defaults are used:

- Standard 18-track recording technology defaults for CST18
- Standard 36-track recording technology defaults for CST36

- Enhanced capacity recording technology defaults for ECCST

If you specify the TAPECAPACITY parameter as part of the SETOAM statement, but do not specify CST18, CST36, or ECCST, or do not enter valid generic or esoteric unit names, OAM does not start and issues message CBR0325I.

If you specify a 3480, 3480*x*, or 3490 unit name on this parameter, OAM accepts it, but considers it to be an esoteric unit name. Generally, the needs of these unit types are covered by the CST18, CST36, or ECCST keywords so they do not normally need to be specified. However, if they are specified, OAM accepts them and ensures that they are valid unit names. If the esoteric unit name used with this parameter matches the *tapeunitname* subparameter on the SETOAM STORAGEGROUP parameter, OAM uses the tape capacity associated with the tapeunitname subparameter instead of any equivalent CST18, CST36, or ECCST specification for this parameter.

Use this parameter at the OAM global level. However, you can specify a desired tape capacity at the storage group level by specifying an esoteric unit name on the *tapeunitname* subparameter on the SETOAM STORAGEGROUP parameter with the *tapecapacity* parameter.

Example: If esoteric TAPE1 is specified on the *tapecapacity* parameter (such as, TAPECAPACITY(TAPE1 5000000)), when the SETOAM STORAGEGROUP(*storagegroupname* TAPEUNITNAME(TAPE1) is used on the same SETOAM statement, the storage group uses the tape capacity of 5 000 000.

If the tape capacity value in the Tape Volume Table is different than that specified on the *tapecapacity* parameter of the SETOAM statement, the value of the SETOAM statement is used for the duration of the OAM session, or until changed or deleted on the SETOAM statement. The Tape Volume Table capacity is not changed after it is initially set during the first time the volume was written to. This is to avoid a changing capacity and a volume fluctuating between full or not full based on a differing capacity. If the capacity in the Tape Volume Table must be changed, you can use SPUFI to dynamically perform this update and OAM will accept it. The free space and percent full calculations for the volume are based on the updated capacity and are recalculated when the volume is written to again as a partial volume, or when OAM is restarted.

Attention: Use caution when you use SPUFI to update fields in the DB2 tables because you might get unexpected results.

Restriction: With the support of the 3590 Model E and later devices, even if the device is in 3490E emulation mode, the capacity of the media is derived from the hardware; therefore, you do not need to specify the SETOAM TAPECAPACITY keyword. Also, specifying this keyword is not valid with a 3590 Model B or a device that emulates a 3590 Model B (3590-1).

TAPEDISPATCHERDELAY(*seconds*)

An optional parameter that specifies a numeric value of 1 through 60 and can be used at the global level only. This specifies that OAM wait a specified number of seconds before demounting a tape volume, even if other work is available for this drive. This delay allows time for a new read request to come into OAM that requires the currently mounted tape volume. This delay can greatly reduce the number of mounts and demounts of volumes for certain applications. This keyword provides function similar to the OPTICALDISPATCHERDELAY keyword associated with the SETOPT statement in the CBROAMxx PARMLIB member.

The OAM tape dispatcher will delay processing of a unit of work for a specific period of time only when all of the following conditions are met:

- A nonzero tape dispatcher delay value has been specified with the TAPEDISPATCHERDELAY keyword on the SETOAM statement in the CBROAMxx PARMLIB member.
- A read request for an object on a currently mounted tape volume has just been completed.
- There is no request for the currently mounted tape volume waiting to be processed on the OAM tape dispatcher queue.
- The OAM tape dispatcher has found a request for another tape volume and is about to dispatch this unit of work.

If another read request for the currently mounted tape volume arrives within the delay interval, that unit of work will be dispatched immediately upon arrival. If no read request for the currently mounted volume arrives within the delay interval, another request for a different tape volume is dispatched.

TAPEEXPIRATION(*YYYY/DDD*)

An optional parameter that specifies the year and date (*YYYY/DDD*) assigned to the data sets on OAM object tape volumes used for expiration purposes where:

- *YYYY* is a four-digit number that specifies a year from 1900 through 2155, and
- *DDD* is a three-digit number that specifies a day from 001 through 366.

The TAPEEXPIRATION date overrides the expiration date defined in the DATACLASS parameter for the data sets residing on the tape volume.

If you specify the TAPEEXPIRATION date for the data sets residing on the OAM object tape volume as the current date or a date preceding the current system date, the data sets are considered previously expired and are therefore eligible for immediate replacement. OAM issues the CBR0317I message to allow you to change the TAPEEXPIRATION value in the SETOAM statement in the CBROAM*xx* PARMLIB member being used if necessary.

Expiration dates of 1999/365 and 1999/366 are considered "never-scratch" dates. Data sets with these expiration dates are not deleted or written over. Check with your tape management system to determine what "never-scratch" date should be specified as the TAPEEXPIRATION date and for other policy-type specifications that are needed in the tape management system to indicate that the tapes and data are being externally managed by OAM. For example, if you are using DFSMSrmm to manage OAM objects on tape, the following vital record specifications, shown in TSO/E format, might be appropriate:

```
RMM ADDVRS DSNAME('OAM.PRIMARY.DATA') COUNT(99999) LOCATION(HOME)
RMM ADDVRS DSNAME('OAM.BACKUP.DATA') COUNT(99999) LOCATION(HOME)
RMM ADDVRS DSNAME('OAM.BACKUP2.DATA') COUNT(99999) LOCATION(HOME)
RMM ADDVRS DSNAME('OPEN') JOBNAME(oamtaskname) COUNT(99999) LOCATION(HOME
RMM ADDVRS DSNAME('ABEND') JOBNAME(oamtaskname) COUNT(99999) LOCATION(HOME)
```

The above DSNAME examples are data set names without DSNWITHSGNAME specified in the SETOAM statement. If DSNWITHSGNAME were specified, the storage group name would be appended as the data set name's low level qualifier. The JOBNAME value is the name of the job and started task for the OAM address space that opens the tape data sets.

TAPEFULLTHRESHOLD(*kilobytes*)

An optional parameter that specifies a numeric value of 0 through 999 999 representing the number of kilobytes of available free space allowed for any volume belonging to any object tape storage group in the configuration. When the number of kilobytes of free space for a tape volume falls below the TAPEFULLTHRESHOLD parameter specified at the OAM global level, the volume is marked full and is not used for any further write requests. The default value for this parameter is zero.

It is important to select a threshold value the allows tape volumes to be marked full in a consistent manner. Consider the size of the objects stored, and if the size of the objects is consistent, select a threshold value that is slightly larger than that size. If volumes are not being selected for new objects and they are not being marked full, increase the value of this parameter.

During OAM initialization, the tape volume full status is checked with the TAPEFULLTHRESHOLD parameter (if specified at the global level) to determine the volume’s free space and the TAPEPERCENTFULL subparameter (if specified at the storage group level) to determine the volume’s percent full status. The volume full status is changed from full to not full if:

- Free space for the volume is greater than the TAPEFULLTHRESHOLD parameter value and the volume percent full value is less than the TAPEPERCENTFULL subparameter.

The volume full status is changed from not full to full if:

- Free space for the volume is less than or equal to the TAPEFULLTHRESHOLD parameter value or the volume percent full value is equal to or greater than the TAPEPERCENTFULL subparameter value.

TAPERECYCLEMODE(*expiration_mode*)

Expired volumes are dispositioned according to the value that you specify with the TAPERECYCLEMODE keyword. Volumes are expired when they are selected by the OSMC Shelf Manager or if they are processed by the MOVEVOL utility with the RECYCLE option.

Restriction: If your tape management system does not allow overwriting tape volumes or data sets by another of the same name, you cannot use the GROUP or OAMSCRATCH option.

GROUP

Leaves the tape volume in the currently assigned Object or Object Backup storage group. OAM writes the CBR2166I message to the console log for each expired tape volume. If you omit the TAPERECYCLEMODE parameter, GROUP is the default.

The following record fields are reset for expired tape volumes in the DB2 TAPEVOL table:

- **Expiration date** is reset to `01-01-0001`.
- **Full status** is set to `N`.
- **Last block ID** is reset to `0`.
- **Freespace** is set to the volume's capacity.
- **Percent full** is reset to `0`.
- **Number of logical kilobytes written** is reset to `0`.
- **Number of physical kilobytes written** is reset to `0`.
- **Number of logical kilobytes deleted** is reset to `0`.
- **Number of logical blocks written** is reset to `0`.
- **Readable status** is reset to Y.
- **Writable status** is reset to Y.
- **Backup type** is reset to the current backup type (1 or 2) if this is an object backup storage group.
- **Dataset name format** is set to blank.
- **Tape sublevel** is set to blank.
- **Capacity overflow** is reset to `0`.
- **Freespace overflow** is reset to `0`.
- **Number of logical kilobytes written overflow** is reset to `0`.
- **Number of physical kilobytes written overflow** is reset to `0`.
- **Number of logical kilobytes deleted overflow** is reset to `0`.

MVSSCRATCH

Removes all knowledge of the tape volume from the OAM inventory and returns the tape volume to the MVS scratch pool. OAM writes the CBR2165I message to the console log for each expired tape volume that is purged from the OAM inventory. In an OAMplex, the OAM member that initially releases the tape volume issues CBR2165I. Then the other OAM members each issue a CBR7404I message indicating that the tape volume is no longer known to that member.

Guideline: Because OAM does not interface with the tape management system, you need to use message automation or manually return the tapes to MVS scratch status. For information on how DFSMSrmm releases tape volumes from OAM, see "Deleting recycled tape and optical volumes from OAM" on page 229.

OAMSCRATCH

Leaves the tape volume under OAM control as an available scratch tape volume to be reassigned to an Object or Object Backup storage group to be reused and rewritten from load point with new data. A message, CBR2164I, is written to the hardcopy console log for each tape volume that completes expiration processing and is reassigned to OAM scratch status. You can reassign this tape volume to any Object or Object Backup storage group that requires a scratch volume with the same attributes.

The following record fields are reset for expired tape volumes in the DB2 TAPEVOL table:

- **Storage group name** is set to `*SCRTCH*`.
- **Volume type** is set to S.
- **Expiration date** is reset to `01-01-0001`.
- **Full status** is set to `N`.
- **Last block ID** is reset to `0`.
- **Freespace** is set to the volume's capacity.
- **Percent full** is reset to `0`.
- **Number of logical kilobytes written** is reset to `0`.
- **Number of physical kilobytes written** is reset to `0`.
- **Number of logical kilobytes deleted** is reset to `0`.
- **Number of logical blocks written** is reset to `0`.
- **Tape compaction status** is reset to blank.
- **Device EPI value** is reset to `0`.
- **Readable status** is reset to Y.
- **Writable status** is reset to Y.
- **Backup type** is set to blank.
- **Dataset name format** is set to blank.
- **Tape sublevel** is set to blank.
- **Volume Attributes Flag** is reset to 0 for rewritable and to 1 for WORM
- **Capacity overflow** is reset to `0`.
- **Freespace overflow** is reset to `0`.
- **Number of logical kilobytes written overflow** is reset to `0`.
- **Number of physical kilobytes written overflow** is reset to `0`.
- **Number of logical kilobytes deleted overflow** is reset to `0`.
- The **unit name** is reset as follows:
  - If the **original unit name** is non-blank, the **unit name** is reset to the value in the **original unit name**.
  - If the **original unit name** is blank, the **unit name** is reset according to the installation's specifications for SETOAM OAMSCRATCHSYNCH.
- The **original unit name** is reset as follows:
  - If the **original unit name** is non-blank, it is unchanged.
  - If the **original unit name** is blank, the **unit name** is reset according to the installation's specifications for SETOAM OAMSCRATCHSYNCH.
- The **data class** is reset as follows:
  - If the **data class** is non-blank, it is unchanged.
  - If the **data class** is blank, it is set according to the installation's specifications for SETOAM OAMSCRATCHSYNCH.

TAPESDB(LARGE | SMALL)
An optional subparameter that specifies whether the maximum block size permitted for a tape volume can be greater than 32760. The maximum block size for a volume is set at the time the first object is written to the volume.

If, when the first object is written to a volume, TAPESDB is set to SMALL (or not set), the maximum block size for the volume is set to 32760.

If, when the first object is written to a volume, TAPESDB is set to LARGE, the maximum block size for that volume is set to the optimal system-determined block size for the device or the maximum permitted block size, whichever is less. The maximum permitted block size is obtained from either the Data Class definition or the TAPEBLKSZLIM keyword in the DEVSUPxx member of PARMLIB, as shown in Table 22 on page 121:

Table 22. Determining maximum permitted block size

Block size limit set in data class?	**TAPEBLKSZLIM set in DEVSUP*xx*?**	**Maximum permitted block size**
Yes	Yes or No	Data class block size limit
No	Yes	TAPEBLKSZLIM value
No	No	32760

See System-Determined Block Size in *z/OS DFSMS Using Data Sets* for details on how the optimal system-determined block size is calculated.

For all subsequent objects written to an OAM tape volume, the maximum block size is the one set for the volume when the first object was written, regardless of the current SETOAM TAPESDB setting and regardless of the size of the first object written.

OAM can read objects from volumes with maximum block sizes greater than 32760, even if TAPESDB is not set or is set to SMALL at the time the object is read.

If TAPESDB is set to LARGE at the time the first object is written to a volume, the volume cannot be read on a system running z/OS Version 1. If volumes are being shared with a z/OS Version 1 system, TAPESDB should be set to SMALL or not set.

Note: This is a global specification only and cannot be specified at the storage group level. Additionally, the TAPESDB specification cannot be modified using the MODIFY OAM,UPDATE command.

SETOAM keyword definitions for STORAGEGROUP subparameters

DATACLASS(*name*)

An optional subparameter of the STORAGEGROUP parameter that specifies the SMS data class to be associated with this Object or Object Backup storage group. Usage of the DATACLASS keyword on the SETOAM statement allows an installation to affect things such as, TDSI, and the tape volume expiration date on an individual Object or Object Backup storage group level. If you do not specify DATACLASS on the SETOAM statement for a *specific* storage group, but you do specify DATACLASS at the global level of the SETOAM statement, the global OAM DATACLASS specification applies to the specific storage group.

In an IBM tape library environment, specify the SETOAM DATACLASS parameter using a data class to request the desired media type or recording technology. To request tape data encryption in the tape library environment and in the stand-alone environment, use DATACLASS to request an encryption recording format; you can also specify the encryption key labels and encoding mechanisms using data class. In addition, you can specify DATACLASS to take advantage of performance scaling or performance segmentation in an IBM tape library environment and also in the stand-alone environment. Use the performance options in the data class to improve the random retrieval rate of primary objects in Object storage groups. For objects written to Object Backup storage groups, you can specify a data class that does not specify performance scaling, then you can use the full capacity of the volume.

Note: Tape data encryption is supported on the 3592, starting with the 3592 Model E05. Performance scaling and performance segmentation is supported on all the 3592 models, starting with the 3592 Model J1A.

Recommendation: Do not allow ACS routines to assign or change the data class assignment of an OAM tape volume. The data class for OAM tape volumes is determined by the SETOAM statement of the CBROAM*xx* PARMLIB member at MVS scratch tape allocation. The SETOAM statement provides this information at the storage group level or at the OAM global level, and it best suits the requirements for the tape volume being allocated. Allowing ACS routines to alter this specification could create unexpected consequences (for example, no compaction of the data when the SETOAM statement specified compaction). It is your installation responsibility to write ACS routines to not alter the data class construct for OAM tape volumes.

DEMOUNTWAITTIME(*seconds*)

An optional subparameter of the STORAGEGROUP parameter. It specifies the time, in seconds, that OAM waits before demounting and deallocating a tape drive (allocated for the storage group specified with the STORAGEGROUP parameter), that OAM is currently not using. For *seconds*, specify a decimal number between 1 and 9999. When the time interval expires, OAM rewinds and unloads the currently mounted tape cartridge and demounts and deallocates the device. The default for this optional parameter is 120 seconds.

Some circumstances might affect how this parameter is enforced:

- If a new tape drive allocation request arrives and OAM has already used the maximum number of tape drives (MAXTAPERETRIEVETASKS + MAXTAPESTORETASKS), then OAM ignores the DEMOUNTWAITTIME and deallocates this drive to allocate another.
- If OAM is canceled, a DEMOUNT occurs, and DEMOUNTWAITTIME is ignored.
- If a request to vary the drive offline is sent while the DEMOUNTWAITTIME is in effect for that drive in an OAM session, the drive cannot vary offline until the specified DEMOUNTWAITTIME elapses.
- If OAM finishes reading and writing to a tape before the DEMOUNTWAITTIME elapses, a demount, unload, rewind, or release of the allocated drive cannot take place until the specified DEMOUNTWAITTIME is complete.

L2DATACLASS(*name*)

An optional subparameter of the STORAGEGROUP parameter that specifies the SMS data class to be associated with this object storage group when you use tape sublevel 2. Tape sublevel is associated with the OAM Sublevel parameter specified in the SMS storage class construct. Usage of the L2DATACLASS keyword on the SETOAM statement allows an installation to affect things such as, TDSI, and the tape volume expiration date on an individual Object storage group level. If you do not specify L2DATACLASS on the SETOAM statement for a specific object storage group, but you do specify L2DATACLASS at the global level of the SETOAM statement, the global L2DATACLASS specification applies to the specific object storage group.

Note: The L2DATACLASS keyword can not be associated with an Object Backup storage group.

In an IBM tape library environment, specify the SETOAM L2DATACLASS parameter using a data class to request the desired media type or recording technology. To request tape data encryption in the tape library environment and in the stand-alone environment, use L2DATACLASS to request an encryption recording format; you can also specify the encryption key labels and encoding mechanisms using data class. In addition, you can specify L2DATACLASS to take advantage of performance scaling or performance segmentation in an IBM tape library environment and also in the stand-alone environment. Use the performance options in the data class to improve the random retrieval rate of primary objects in Object storage groups. For objects written to Object Backup storage groups, you can specify a data class that does not specify performance scaling, then you can use the full capacity of the volume.

Note: Tape data encryption is supported on the 3592, starting with the 3592 Model E05. Performance scaling and performance segmentation is supported on all the 3592 models, starting with the 3592 Model J1A.

L2TAPEUNITNAME(*unitname*)

A required subparameter of the STORAGEGROUP parameter, if you use the tape sublevel 2 function. Tape sublevel is associated with the OAM Sublevel parameter specified in the SMS storage class construct. This subparameter specifies the type of tape drive that OAM uses when writing data to an Object storage group using Tape Sublevel 2. This L2TAPEUNITNAME is the MVS unit name that OAM uses to initially allocate a scratch tape when an object is stored to this object storage group and stored on a tape volume. For *unitname*, specify the name of a valid MVS esoteric (group of devices defined to a group name) or a generic unit name. Valid generic unit names are:

- 3480—a base 3480 device
- 3480*x*—a 3480 device with the IDRC feature, or a base 3490 device
- 3490—a 3490E device
- 3590-1—a 3590 device (or a device that emulates a 3590-1)

The unit name specified is associated with each tape volume used for output during the process of writing objects to tape that belong to a specified object storage group. This unit name is saved in the corresponding rows in the TAPEVOL table for each of these tape volumes, and is used during later allocations of these tape volumes for either reading or writing processing.

Note: The L2TAPEUNITNAME keyword cannot be associated with an Object Backup storage group.

L2TAPEUNITNAME is a required keyword when using tape sublevel 2, and is specified for all tape sublevel 2 allocations. In the automated tape library dataservers and manual tape libraries, this information might be used by the ACS filter routines, but is not required for device allocation. In the stand-alone environment, this information is **critical** in the allocation decision making process.

Note:

1. Even though a tape unit name is specified for the group, the ACS routines (for environment ALLOC), can override the L2TAPEUNITNAME specification by assigning the allocation to a Tape storage group, thereby steering the allocation into an ATLDS or an MTL.
2. When OAM requests a mount for a generic tape with a L2TAPEUNITNAME of 3480, it accepts the 3480 tape drive chosen regardless of whether that tape drive has IDRC. MVS Allocation does not use the JCL or dynamic allocation parameter for compaction when determining device eligibility. If tape compaction is requested, and the tape is mounted on a 3480 tape drive that does not have IDRC, allocation fails. To prevent this failure, OAM does not allow tape compaction with a generic L2TAPEUNITNAME of 3480. OAM uses the NOTAPECOMPACTION keyword with all data for this L2TAPEUNITNAME.

Using Esoteric Unit Names: To ensure that objects written on any drive in the esoteric group can be read on any drive in the same esoteric group, OAM does not allow a mixed esoteric unit name that consists of different device types; for example, 3590-1 and 3490E. Avoid using an esoteric unit name that consists of tape drives that write in different recording technologies; for example, a 3590 Model H and a 3590 Model E. Depending on which emulation mode is being used, both drives appear as though they have the same device type (3590-1 or 3490E). However, the read/write request might fail if MVS allocation selects an incompatible tape drive.

MAXTAPERETRIEVETASKS(*tasks*)

An optional subparameter of the STORAGEGROUP parameter that specifies the maximum number of tape drives used for reading objects from tape volumes belonging to a specific Object or Object Backup storage group. This parameter specifies the maximum number of tasks within the OAM address space that can concurrently read objects from tape for the storage group specified with the STORAGEGROUP parameter. This subparameter and the MAXTAPESTORETASKS subparameter control the maximum number of tape drives that can be concurrently allocated to the OAM address space for reading from and writing to tape volumes belonging to the specified Object or Object Backup storage group. For *tasks*, specify a decimal number between 1 and 100.

The value specified with the MAXTAPERETRIEVETASKS *subparameter* of the STORAGEGROUP parameter for a specific Object or Object Backup storage group cannot exceed the global maximum number of tape retrieve tasks specified with the MAXTAPERETRIEVETASKS *parameter* of the SETOAM statement. If it does, an error message is issued, and the SETOAM statement is rejected. If you do not specify this subparameter on any SETOAM statement, the OAM default is 1.

MAXTAPESTORETASKS(*tasks*)

An optional subparameter of the STORAGEGROUP parameter that specifies the maximum number of tape drives used for writing objects to tape volumes belonging to a specific Object or Object Backup storage group. This parameter specifies the maximum number of tasks within the OAM address space that can concurrently write objects to tape volumes belonging to the Object or Object Backup storage group specified with the STORAGEGROUP parameter. This subparameter and the MAXTAPERETRIEVETASKS subparameter control the maximum number of tape drives that can be concurrently allocated to the OAM address space for writing to and reading from tape volumes belonging to the specified Object or Object Backup storage group. For *tasks*, specify a decimal number between 1 and 100.

The value specified with the MAXTAPESTORETASKS *subparameter* of the STORAGEGROUP parameter for a specific Object or Object Backup storage group, cannot exceed the global maximum number of tape store tasks specified with the MAXTAPESTORETASKS *parameter* of the SETOAM statement. If it does, an error message is issued, and the SETOAM statement is rejected.

Requirement: To use more than one tape drive for a storage group to write object data to tape, the TAPEDRIVESTARTUP threshold must be low enough to trigger the startup of the additional tape drive. This threshold is a value (in megabytes) of write data pending for this storage group. See the discussion concerning TAPEDRIVESTARTUP on page "TAPEDRIVESTARTUP(threshold in megabytes)" on page 127 for more information.

SGMAXRECYCLETASKS(*nn*)

An optional parameter that you can specify at the storage group level. The *nn* is the maximum number of MOVEVOL tasks that can be run concurrently by the RECYCLE function for a storage group. The value for SGMAXRECYCLETASKS cannot exceed the value for MAXRECYCLETASKS. Valid values for *nn* are 0 - 15. The default is 1 if no value is specified. A value of 0 indicates that no RECYCLE operations can be run at the storage group level specified.

If you only want to recycle volumes from one group, the setting for all other groups would be 0 to ensure that the group with a non-zero value receives all the recycling processing.

If you specify a value for a group that is higher than the value for another group, the system selects more of the volumes to recycle from the group with the higher value. However, RECYCLE processing might be working with the original order of volumes that are sorted by the amount of valid data for each volume, and might select volumes from other groups to satisfy the limit before it processes the higher-value group.

SGMAXTAPERETRIEVETASKS(*tasks*)

An optional parameter you can be specify at the storage group level only. When you specify this keyword at a storage group level, it is identical to the MAXTAPERETRIEVETASKS keyword. This keyword SGMAXTAPERETRIEVETASKS is the preferred naming convention, however, for compatibility the MAXTAPERETRIEVETASKS keyword continues to be maintained.

SGMAXTAPERETRIEVETASKS(*tasks*) and its alternate MAXTAPERETRIEVETASKS(*tasks*) provide an optional subparameter of the STORAGEGROUP parameter that specifies the maximum number of tape drives used for reading objects from tape volumes belonging to a specific Object or Object Backup storage group. This parameter specifies the maximum number of tasks within the OAM address space that can concurrently read objects from tape for the storage group specified with the STORAGEGROUP parameter. The SGMAXTAPERETRIEVETASKS subparameter and the SGMAXTAPESTORETASKS subparameter control the maximum number of tape drives that can be concurrently allocated to the OAM address space for reading from and writing to tape volumes belonging to the specified Object or Object Backup storage group. For tasks, specify a decimal number between 1 and 100. The value specified with the SGMAXTAPERETRIEVETASKS keyword (or MAXTAPERETRIEVETASKS subparameter of the STORAGEGROUP parameter for a specific Object or Object Backup storage group) cannot exceed the global maximum number of tape retrieve tasks specified with the MAXTAPERETRIEVETASKS parameter of the SETOAM statement. If it does exceed, an error message is issued, and OAM initialization is terminated. If you do not specify this subparameter on any SETOAM statement, the OAM default is 1.

The `F OAM,UPDATE,SETOAM` command can be used to dynamically update the SGMAXTAPERETRIEVETASKS value. No restart of the OAM address space is required.

SGMAXTAPESTORETASKS(*tasks*)

An optional parameter that you can specify at the storage group level only. This keyword is identical to the MAXTAPESTORETASKS keyword when specified at a storage group level. This keyword SGMAXTAPESTORETASKS is the preferred naming convention, however the keyword MAXTAPESTORETASKS is maintained for legacy compatibility.

SGMAXTAPESTORETASKS(*tasks*) and its alternate MAXTAPESTORETASKS(*tasks*) provide an optional subparameter of the STORAGEGROUP parameter that specifies the maximum number of tape drives used for writing objects to tape volumes belonging to a specific Object or Object Backup storage group. This parameter specifies the maximum number of tasks within the OAM address space that can concurrently write objects to tape volumes belonging to the Object or Object Backup storage group specified with the STORAGEGROUP parameter. This subparameter and the SGMAXTAPERETRIEVETASKS subparameter control the maximum number of tape drives that can be concurrently allocated to the OAM address space for writing to and reading from tape volumes belonging to the specified Object or Object Backup storage group. For tasks, specify a decimal number between 1 and 100. The value specified with the SGMAXTAPESTORETASKS keyword (or the MAXTAPESTORETASKS subparameter of the STORAGEGROUP parameter for a specific Object or Object Backup storage group), cannot exceed the global maximum number of tape store tasks specified with the MAXTAPESTORETASKS parameter of the SETOAM statement. If it does exceed, an error message is issued, and OAM initialization is terminated.

The `F OAM,UPDATE,SETOAM` command can be used to dynamically update the SGMAXTAPESTORETASKS value. No restart of the OAM address space is required.

TAPEUNITNAME(*unitname*)

A required subparameter of the STORAGEGROUP parameter that specifies the type of tape drive that OAM uses when writing data to an Object or Object Backup storage group. This TAPEUNITNAME is the MVS unit name that OAM uses to initially allocate a scratch tape when an object is stored to this Object or Object Backup storage group and stored on a tape volume. For *unitname*, specify the name of a valid MVS esoteric (group of devices defined to a group name) or a generic unit name. Valid generic unit names are:

- 3480—a base 3480 device
- 3480*x*—a 3480 device with the IDRC feature, or a base 3490 device
- 3490—a 3490E device
- 3590-1—a 3590 device (or a device that emulates a 3590-1)

The unit name specified is associated with each tape volume used for output during the process of writing objects to tape that belong to a specified Object or Object Backup storage group. This unit name is saved in the corresponding rows in the TAPEVOL table for each of these tape volumes, and is used during later allocations of these tape volumes for either reading or writing processing.

TAPEUNITNAME is a required keyword, and is specified for all allocations. In the automated tape library dataservers and manual tape libraries, this information might be used by the ACS filter routines, but is not required for device allocation. In the stand-alone environment, this information is **critical** in the allocation decision-making process.

Note:

1. Even though a tape unit name is specified for the group, the ACS routines (for environment ALLOC), can override the TAPEUNITNAME specification by assigning the allocation to a Tape storage group, thereby steering the allocation into an ATLDS or an MTL.
2. When OAM requests a mount for a generic tape with a TAPEUNITNAME of 3480, it accepts the 3480 tape drive chosen regardless of whether that tape drive has IDRC. MVS Allocation does not use the JCL or dynamic allocation parameter for compaction when determining device eligibility. If tape compaction is requested, and the tape is mounted on a 3480 tape drive that does not have IDRC, allocation fails. To prevent this failure, OAM does not allow tape compaction with a generic TAPEUNITNAME of 3480. OAM uses the NOTAPECOMPACTION keyword with all data for this TAPEUNITNAME.

Using Esoteric Unit Names: To ensure that objects written on any drive in the esoteric group can be read on any drive in the same esoteric group, OAM does not allow a mixed esoteric unit name that consists of different device types; for example, 3590-1 and 3490E. Avoid using an esoteric unit name that consists of tape drives that write in different recording technologies; for example, a 3590 Model H and a 3590 Model E. Depending on which emulation mode is being used, both drives appear as though they have the same device type (3590-1 or 3490E). However, the read/write request might fail if MVS allocation selects an incompatible tape drive.

TAPECOMPACTION | NOTAPECOMPACTION

Specify either the optional TAPECOMPACTION or NOTAPECOMPACTION subparameter of the STORAGEGROUP parameter. These parameters specify whether the objects for this storage group are to be written in compacted or noncompacted format. See Table 23 on page 126 for an example of this selection process.

TAPECOMPACTION specifies that the compaction feature of the tape drive is enabled when OAM is writing objects to tape which belong to the specified Object or Object Backup storage group. This parameter is ignored if the unit name specified with the TAPEUNITNAME parameter is a mixed esoteric group.

If you specify an esoteric unit name for a mixed esoteric group that consists of at least one IBM 3480 tape drive without the IDRC feature and one IBM 3480 or 3490 tape drive with the IDRC feature, the TAPECOMPACTION keyword is ignored and the NOTAPECOMPACTION keyword is assumed.

To enable tape compaction, perform one of the following tasks as appropriate:

- Specify the TAPECOMPACTION keyword on the SETOAM statements in the CBROAM*xx* PARMLIB member.
- Omit the TAPECOMPACTION and the NOTAPECOMPACTION keywords on the SETOAM statements in the CBROAM*xx* PARMLIB member and specify a DATACLASS on the SETOAM statement. In the definition of the data class (specified with the DATACLASS keyword on the SETOAM statement), specify a COMPACTION option of "YES".
- Omit the TAPECOMPACTION and NOTAPECOMPACTION keywords on the SETOAM statements in the CBROAM*xx* PARMLIB member and do not specify a DATACLASS keyword on the SETOAM statements. Instead, specify the COMPACT=YES option in the DEVSUP*xx* PARMLIB member processed during IPL.

NOTAPECOMPACTION specifies that the compaction feature of the tape drive is disabled when OAM is writing objects to tape belonging to the specified Object or Object Backup storage group.

If you do not specify this subparameter on any SETOAM statement, the OAM default for the specified Object or Object Backup storage group is determined from the DATACLASS associated with this storage group. To use either tape compaction or no tape compaction, specify the DATACLASS with either TAPECOMPACTION or NOTAPECOMPACTION. If no DATACLASS is associated with this storage group or if the DATACLASS associated with this storage group has a blank TAPECOMPACTION specification, then whether tapes added to this storage group will use the compaction feature is determined by the DEVSUP parameter defaults provided during the allocation process.

Table 23. Example of the TAPECOMPACTION / NOTAPECOMPACTION Selection Process

Was TAPECOMPACTION or NOTAPECOMPACTION specified?	
YES	Do what is specified.
NO	Check data class specification.
Was DATACLASS specified?	

Table 23. Example of the TAPECOMPACTION / NOTAPECOMPACTION Selection Process (continued)	
Was TAPECOMPACTION or NOTAPECOMPACTION specified?	
YES	Was compaction either YES or NO? **YES** Do what is specified. **NO** Do what is specified.
NO	DATACLASS was not specified, did not apply, or had a blank compaction specification. Consider the DEVSUP specification.
Is there a DEVSUP specification?	
YES	Do what is specified.
NO	If 3480, NOTAPECOMPACTION used. If 3480*x* or 3490, TAPECOMPACTION used. If 3590-1, TAPECOMPACTION used.

Requirement: If you want to change the TAPECOMPACTION | NOTAPECOMPACTION attribute associated with the storage group and you want OAM to only write data in the new format, you must update the tape volume table using the MODIFY OAM,UPDATE,VOLUME,*volser*,WRITABLE,N command (or by using SPUFI while the OAM address space is not active) to mark the existing tape volumes in the storage group unwritable. You can update the SETOAM TAPECOMPACTION parameter by using the MODIFY OAM,UPDATE,SETOAM,*scope*,TCOMP,Y command or by updating the CBROAM*xx* member of PARMLIB and restarting the OAM address space.

TAPEDRIVESTARTUP(*threshold in megabytes*)

An optional subparameter of the STORAGEGROUP parameter that specifies the drive startup threshold used for writing objects to tape volumes belonging to a specific Object or Object Backup storage group. The parameter indicates when OAM is to start the use of another tape drive for writing objects to tape volumes belonging to the storage group specified with the STORAGEGROUP parameter. When the number of MB of object data waiting to be written to tape is divided by the number of tape drives currently writing object data to tape exceeds the threshold specified by *threshold in megabytes*, OAM attempts to use another tape drive to write object data to the specified Object or Object Backup storage group.

However, the maximum number of tape drives being used by OAM to write object data to a specific Object or Object Backup storage group is limited by the value specified with the MAXTAPESTORETASKS subparameter of the STORAGEGROUP parameter on the SETOAM statement. The limit specified with this subparameter will never be exceeded.

Additionally, the maximum number of tape drives being used by OAM to write object data to all Object or Object Backup storage groups is limited by the value specified with the global MAXTAPESTORETASKS parameter of the SETOAM statement. The limit specified with this parameter will never be exceeded.

For *threshold in megabytes (MB)*, specify a decimal number between 1 and 9999. If you do not specify this subparameter on any SETOAM statement, the OAM default is 9999.

Requirement: Drive startup threshold in an optical environment is determined differently than the threshold in a tape storage environment. See the discussion concerning "DRIVE STARTUP THRESHOLD" on page 161 for more information.

TAPEEXPIRATION(*YYYY/DDD*)

An optional subparameter of the STORAGEGROUP parameter that specifies the year and date (*YYYY/DDD*) assigned to the data sets on OAM object tape volumes belonging to a specific Object or Object Backup storage group for expiration purposes where:

- *YYYY* is a four-digit number that specifies a year from 1900 through 2155
- *DDD* is a three-digit number that specifies a day from 001 through 366

The TAPEEXPIRATION date for the data sets on the tape volumes belonging to the Object or Object Backup storage group overrides the expiration date defined in the DATACLASS parameter for these data sets.

If you specify the TAPEEXPIRATION date for the data sets on the tape volumes belonging to the specific Object or Object Backup storage group as the current date or a date preceding the current system date, the data sets are considered previously expired and are therefore eligible for immediate replacement. OAM issues the CBR0317I message to allow you to change the TAPEEXPIRATION value in the SETOAM statement of the CBROAM*xx* PARMLIB member being used, if necessary.

Expiration dates of 1999/365 and 1999/366 are considered "never-scratch" dates. Data sets with these expiration dates are not deleted or written over. Check with your tape management system to determine what "never-scratch" date should be specified as the TAPEEXPIRATION date and for other policy-type specifications that are needed in the tape management system to indicate that the tapes and data are being externally managed by OAM. For example, if you are using DFSMSrmm to manage OAM objects on tape, the following vital record specifications, shown in TSO/E format, might be appropriate:

```
RMM ADDVRS DSNAME('OAM.PRIMARY.DATA') COUNT(99999) LOCATION(HOME)
RMM ADDVRS DSNAME('OAM.BACKUP.DATA') COUNT(99999) LOCATION(HOME)
RMM ADDVRS DSNAME('OAM.BACKUP2.DATA') COUNT(99999) LOCATION(HOME)
RMM ADDVRS DSNAME('OPEN') JOBNAME(oamtaskname) COUNT(99999) LOCATION(HOME
RMM ADDVRS DSNAME('ABEND') JOBNAME(oamtaskname) COUNT(99999) LOCATION(HOME)
```

The above DSNAME examples are data set names without DSNWITHSGNAME specified in the SETOAM statement. If DSNWITHSGNAME were specified, the storage group name would be appended as the data set name's low level qualifier. The JOBNAME value is the name of the job and started task for the OAM address space that opens the tape data sets.

TAPEFULLTHRESHOLD(*kilobytes*)

An optional subparameter of the STORAGEGROUP parameter that specifies a numeric value of 0 through 999 999 representing the number of KB of available free space allowed for each volume belonging to the Object storage group specified in the STORAGEGROUP parameter. When the number of KB of free space for a tape volume falls below the TAPEFULLTHRESHOLD subparameter for the storage group to which that volume belongs, the volume is marked full and is not used for any further write requests. The default value for this parameter is zero.

Recommendation: You should select a threshold value that allows tape volumes to be marked full in a consistent manner. Consider the size of the objects stored, and if the size of the objects is consistent, select a threshold value for the storage group that is slightly larger than that size. If volumes are not being selected for new objects and they are not being marked full, increase the value of this parameter.

During OAM initialization, the tape volume full status is checked with the TAPEFULLTHRESHOLD subparameter to determine the volume’s free space and the TAPEPERCENTFULL subparameter to determine the volume’s percent full status. The volume full status is changed from full to not full if:

- Free space for the volume is greater than the TAPEFULLTHRESHOLD subparameter value and the volume percent full value is less than the TAPEPERCENTFULL subparameter.

The volume full status is changed from not full to full if:

- Free space for the volume is less than or equal to the TAPEFULLTHRESHOLD subparameter value or the volume percent full value is equal to or greater than the TAPEPERCENTFULL subparameter value.

TAPEPERCENTFULL(*percent*)

An optional subparameter of the STORAGEGROUP parameter that specifies the percent full utilization used for writing objects to tape volumes belonging to a specific Object or Object Backup storage group. This parameter indicates at what percent of utilization OAM stops writing objects to tape volumes belonging to the storage group specified with the STORAGEGROUP parameter on the SETOAM statement.

When the tape volume utilization percentage for a tape volume belonging to an Object or Object Backup storage group reaches or exceeds the threshold specified by *percent*, OAM stops writing objects to the tape volume. The tape volume is marked full and another tape volume belonging to the specified Object or Object Backup storage group is selected for the continuation of writing objects to that storage group. If there are no tape volumes in the storage group with enough space to accommodate the object to be written, or if TAPEDRIVESTARTUP processing is attempting to start an additional tape drive and an additional scratch tape is required to start that drive, an OAM scratch tape is added to the group. If there are no OAM scratch tapes available, then an MVS scratch tape is requested and added to the Object or Object Backup storage group.

For *percent*, specify a decimal number between 1 and 100. If you do not specify this subparameter on any SETOAM statement, the OAM default is 100.

The TAPEPERCENTFULL subparameter is retroactive for the Object and the Object Backup storage groups defined in the ACDS. Whenever OAM is started and the TAPEPERCENTFULL for an Object or Object Backup storage group which is currently defined in the ACDS has been changed since the last time OAM was started, that new TAPEPERCENTFULL value is applied to all tapes which currently belong to the subject STORAGEGROUP. This situation means that several changes might be made to the tape volume table rows for the volumes in the storage group:

- The free space (FRESPACE) for a volume might increase or decrease depending on whether the TAPEPERCENTFULL is increased or decreased.
- The volume full indicator (FULL) might change from full to not full, or from not full to full, depending on whether the TAPEPERCENTFULL is increased or decreased.

The adjustment is made to all affected tapes regardless of whether the tapes were previously marked full, unreadable, or unwritable. If new volumes are added to the storage group, they conform to the new TAPEPERCENTFULL value specified on the SETOAM statement that is being used for the current OAM initialization.

Requirement: If you modify the TAPEPERCENTFULL value using the MODIFY OAM,UPDATE command, a volume you have marked full might be subsequently marked not full. This is because the volume's current tape full percentage is less than the value of the TAPEPERCENTFULL parameter on the SETOAM statement. If you intend to mark the volumes in an Object or Object Backup storage group as full, then you must increase the value of the volume's percent full value (PFULL) to 100.

[NOT Programming Interface Information]

Note: OAM can mark a tape volume full when:

- An 18-track tape reaches sector 69.
- A 36-track tape reaches sector 1 on the second wrap. This is done to prevent OAM from falling into EOV processing.

[End NOT Programming Interface Information]

For optical storage, specify the volume full threshold parameter to determine the threshold value for an optical volume. See "VOLUME FULL THRESHOLD" on page 163 for information on the volume full threshold parameter for optical volumes.

SETOPT statements for options

Note: In a multiple OAM configuration, keywords and values related to optical processing are not supported on the SETOPT statement as described below. If these keywords and values are specified, they are ignored.

The SETOPT statement and its associated keywords of the CBROAM*xx* PARMLIB member define general rules or OPTIONS at a global and storage group level that span all of the OAM environments of disk, optical, and tape.

Requirement: Creating or updating the CBROAM*xx* PARMLIB member with SETOPT statements is required to return rewritable optical volumes to a common scratch pool or to reinitialize the volumes to their original storage group affiliation.

Related reading: See "Using the UPDATE command to set SETOAM, SETOSMC, and SETOPT values" on page 366 for information on changing the SETOPT values dynamically or on defining the values when the CBROAM*xx* PARMLIB member is not used at initialization.

Sample SETOPT statement

Figure 14 on page 130 shows examples of SETOPT statements in the CBROAM*xx* PARMLIB member that can be used as samples for your installation. The descriptions of each of the keywords are found in the discussion of the SETOPT statement (see "SETOPT keyword definitions" on page 132).

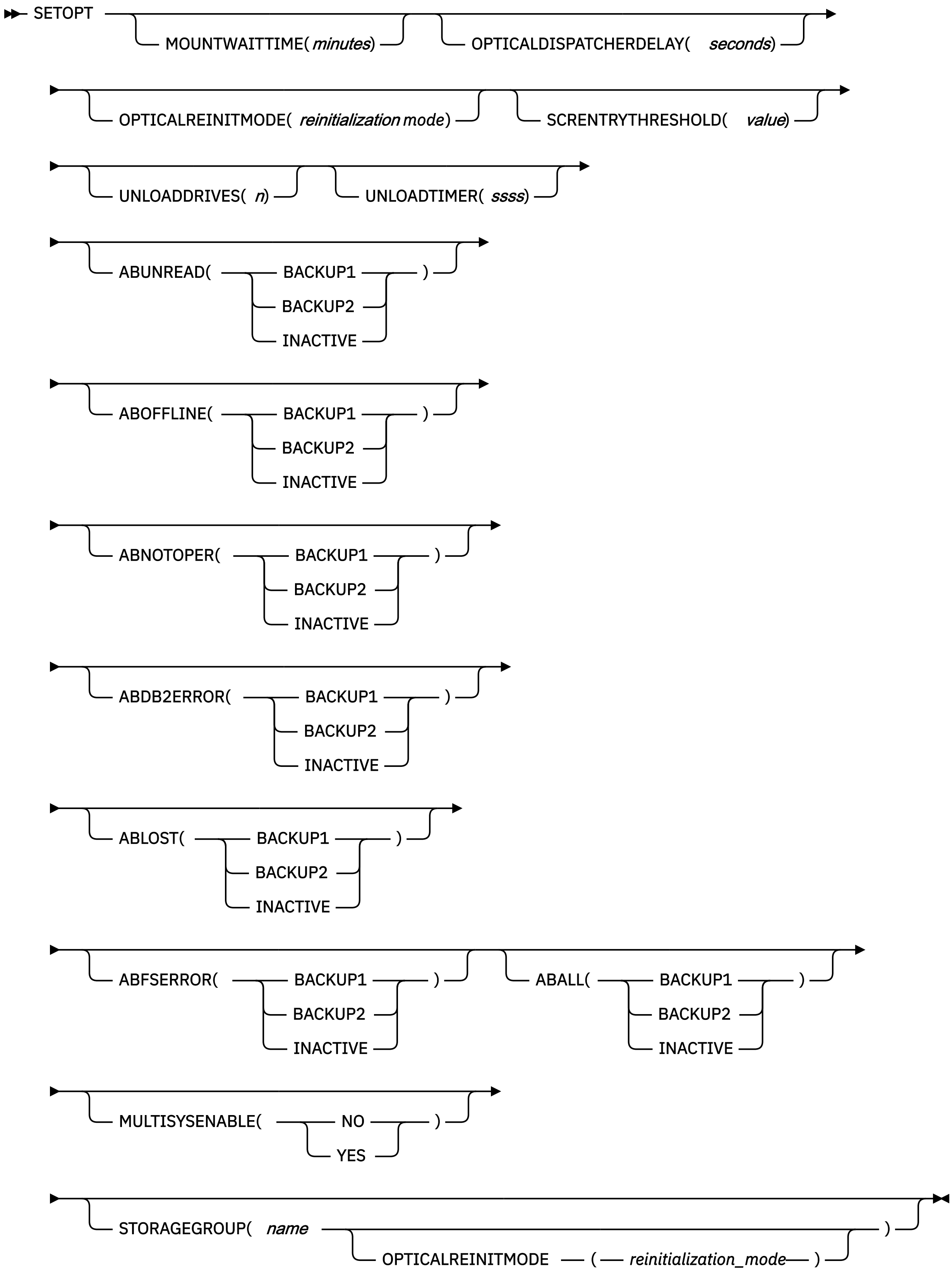
```
SETOPT OPTICALREINITMODE(GROUP)
       OPTICALDISPATCHERDELAY(4)
       MOUNTWAITTIME(3)
       UNLOADDRIVES(6)
       UNLOADTIMER(120)
       MULTISYSENABLE(YES)
       STORAGEGROUP(GROUP01 OPTICALREINITMODE(GROUP))

              OR

SETOPT OPTICALREINITMODE(GROUP)
       OPTICALDISPATCHERDELAY(3)
       SCRENTRYTHRESHOLD(5000)
       STORAGEGROUP(GROUP02 OPTICALREINITMODE(OAMSCRATCH))

              OR
SETOPT ABUNREAD(BACKUP1)
       ABOFFLINE(BACKUP1)
       ABNOTOPER(BACKUP1)
       ABDB2ERROR(BACKUP1)
       ABLOST(BACKUP1)

              OR

SETOPT ABALL(BACKUP1)
```

Figure 14. CBROAMxx PARMLIB Member Samples Using the SETOPT Statement and Optional Parameters

Rule: If you specify the STORAGEGROUP option, you must also specify *name*. All other parameters under STORAGEGROUP are optional.

The syntax for the SETOPT statement follows.

SETOPT Statement Syntax: OAM Global Level Parameters



SETOPT keyword definitions

The SETOPT statement and its associated keywords of the CBROAM*xx* PARMLIB member define general rules or OPTIONS at a global and storage group level that spans all of the OAM environments of disk, optical, and tape.

Note: In a multiple OAM configuration, keywords and values that are related to optical processing are not supported on the SETOPT statement as described below. If these keywords and values are specified, they are ignored.

The following keywords are defined as they pertain to the OAM SETOPT statement.

MOUNTWAITTIME

Specifies the amount of time (in minutes) that can pass while a volume is waiting to be mounted on an operator-accessible drive within an optical library. After this time has expired, message CBR4426D is issued to allow the operator to retry or to cancel the volume mount request.

This value can be any numeric value 1 - 9999. If the operator retries the mount request, the value that is specified in the MOUNTWAITTIME parameter is used for the retry. The default value of this parameter is five minutes.

Because this parameter can be changed dynamically, the new value that is assigned to the MOUNTWAITTIME is used for any currently executing volume mount request. The time of the initial mount request is compared to the value specified in the MOUNTWAITTIME parameter and is deducted from the specified parameter value. For example, if a volume mount request has been waiting for one minute and the MOUNTWAITTIME is changed to indicate a wait time of two minutes from a previous value of three, the volume mount request has only one more minute to finish before CBR4426D is issued. If a retry is requested after CBR4426D is issued, and the MOUNTWAITTIME is dynamically changed after the reply to retry the request, the minutes specified in the MOUNTWAITTIME parameter are compared to the time of the most recent reply to this message to determine the exact wait time.

Note: In a multiple OAM configuration, MOUNTWAITTIME is not supported and will be ignored if specified.

OPTICALDISPATCHERDELAY

Specifies the number of seconds that the OAM optical dispatcher is to delay processing of certain requests to minimize flipping of optical disk cartridges in an automated optical storage library. The OAM optical dispatcher delays processing of a unit of work for a specific period of time, when ALL of the following conditions are true:

- A read request for an object on a currently mounted optical disk volume has just been completed.
- No request for the currently mounted optical disk volume is waiting to be processed on the OAM optical dispatcher queue.
- The OAM optical dispatcher has found a read request for another optical disk volume (either the opposite side of the currently mounted volume or for an unmounted optical disk volume) and is about to dispatch this unit of work.
- A nonzero optical dispatcher delay value has been specified with the OPTICALDISPATCHERDELAY keyword on the SETOPT statement in the CBROAM*xx* PARMLIB member.

In this situation, the OAM optical dispatcher delays the dispatching of this selected unit of work (for the number of seconds specified by the installation) expecting that another read request for the currently mounted optical disk volume will arrive within this delay interval. The OAM optical dispatcher delays dispatching of the selected unit of work for up to the number of seconds specified with the OPTICALDISPATCHERDELAY keyword on the SETOPT statement in the CBROAM*xx* PARMLIB member.

If another read request for the currently mounted optical disk volume arrives within the delay interval, that unit of work is dispatched immediately upon arrival. If no read request for the currently mounted optical disk volume arrives within the delay interval another request for a different optical disk volume (either the opposite side of the currently mounted optical disk volume or an unmounted optical disk volume) is dispatched.

You can use the OPTICALDISPATCHERDELAY value to circumvent a performance problem when IBM optical disk libraries (IBM 3995 optical libraries) are used with certain microfiche replacement applications. The problem involves the constant servicing of requests for data on both sides of an optical disk cartridge resulting in the cartridge being constantly flipped over to access data on the opposite side of the optical disk cartridge. This constant flipping of the cartridge results in longer response times for requests to read data from each side of the optical disk cartridge.

Valid value *seconds* specifies the number of seconds that the OAM optical dispatcher is to delay dispatching of specific units of work under the circumstances described above. Valid values for seconds is a decimal number 1 - 60. If you need to use this parameter, use a low value 1 - 5.

Note: In a multiple OAM configuration, OPTICALDISPATCHERDELAY is not supported and will be ignored if specified.

OPTICALREINITMODE

Specifies reinitialization mode for rewritable optical cartridges. The following values are valid:

GROUP

Expired rewritable optical cartridges remain assigned to the original Object or Object Backup storage group when reinitialized. This option is the default.

OAMSCRATCH

Expired rewritable optical cartridges revert to *SCRTCH* storage group when reinitialized. These cartridges are available to be reassigned to any Object or Object Backup storage group.

Use the OPTICALREINITMODE keyword to determine whether an optical cartridge at reinitialization should maintain its storage group affiliation or revert to a scratch storage group. A cartridge's reinitialization mode is set according to any OPTICALREINITMODE option in effect when OSMC Shelf Space Manager selects the cartridge for reinitialization, not according to the options in effect when the optical cartridge is physically reinitialized.

When the MOVEVOL command with the RECYCLE option completes and all of the objects have been successfully moved off of the optical volumes, the volumes are either assigned to their current Object or Object Backup storage group or returned to OAM scratch, depending on the SETOPT OPTICALREINITMODE statement that is specified in the CBROAM*xx* member of PARMLIB. When an optical volume is returned to OAM scratch, it is available to be reassigned to any Object or Object Backup storage group. Rewritable optical media is marked for reinitialization and the cartridge is physically reformatted the next time that the cartridge is mounted on a drive to reclaim the used space on the cartridge. WORM optical media is not physically reformatted because the used space cannot be reclaimed on WORM media.

To understand how the OPTICALREINITMODE keyword affects the optical volumes at reinitialization, it is important to understand the reinitialization process itself. The following information gives you an overview of the process.

An optical media cartridge contains two logical optical disk volumes. Each optical volume is assigned a unique volume serial number (*volser*). The optical media types are either WORM or rewritable.

Shelf Space Manager (a component of OSMC) processes expired optical disk cartridges as follows:

- For WORM cartridges:
  - If all objects on both *volsers* have been deleted and both *volsers* are full and no objects have been written to this cartridge in the past 24 hours, then the cartridge is ejected if it is library-resident. Message CBR2153I is issued to inform the installation that all of the objects on the WORM cartridge were expired and the cartridge was removed from the OAM Configuration Database (OCDB). The WORM cartridge no longer contains valid data; you can dispose of it according to federal, state, and local laws.
- For rewritable cartridges:
  - If all objects on both *volsers* are deleted and no objects were written to this cartridge in the last 24 hours, then the volume empty (VOLEMPTY) indicators in the OAM volume table in the OCDB

for both *volsers* contained in the cartridge are set to indicate that the cartridge is ready to be reinitialized. Message CBR2154I is issued to inform the installation that this rewritable cartridge will be reinitialized the next time it is mounted on an optical drive.

- For all cartridges:
  - The expiration date needs to be the current day or earlier.

Tip: You can use the MODIFY OAM,UPDATE,VOLUME command to update the volume expiration date. See "Updating fields in the DB2 Volume Table and the Tape Volume Table" on page 374 for more information on this command.

When a rewritable optical cartridge that is selected by Shelf Space Manager for reinitialization is mounted, both sides of the cartridge are reformatted. The volumes on the reformatted cartridge retain their original volume serial numbers. With the SETOPT statements, you can specify whether the reinitialized cartridge should maintain its storage group affiliation (default) or revert to the scratch storage group.

Note: In a multiple OAM configuration, OPTICALREINITMODE is not supported and will be ignored if specified.

SCRENTRYTHRESHOLD
Specifies the amount of free space, in KB, that will determine a WORM optical volume's eligibility to be assigned as a scratch volume. If a new WORM optical volume has less free space than specified, a message is issued to validate the entry or labeling of the volume as a scratch cartridge. The default value is 0.

Note: In a multiple OAM configuration, SCRENTRYTHRESHOLD is not supported and will be ignored if specified.

UNLOADDRIVES
Specifies that the number of optical drives specified by *n* are desired to be empty, unloading drives if necessary, when the value of UNLOADTIMER has been reached. *n* is a numeric value 1 - 6. This keyword can be specified at the global level only.

Note: In a multiple OAM configuration, UNLOADDRIVES is not supported and will be ignored if specified.

UNLOADTIMER
Specifies the period of inactivity, in seconds, to wait before unloading the optical drives specified in UNLOADDRIVES. *ssss* is a numeric value 1 - 9999. This keyword can be specified at the global level only.

Note: In a multiple OAM configuration, UNLOADTIMER is not supported and will be ignored if specified.

ABUNREAD
Specifies that when a retrieve for an object is attempted and the optical or tape volume on which the object resides is marked not readable, the specified backup copy of the object is retrieved.

Valid values for ABUNREAD are:

BACKUP1
Specifies that when a primary copy of an object that is read by an application is not available for the specified reason, and BACKUP1 is specified for ABUNREAD, an attempt is made to retrieve the object from the first backup copy of the object.

BACKUP2
When a primary copy of an object that is read by an application is not available for the specified reason, and BACKUP2 is specified for ABUNREAD, an attempt is made to retrieve the object from the second backup copy of the object.

INACTIVE
When a primary copy of an object that is read by an application is not available for the specified reason, and INACTIVE is specified, then automatic access to backup is disabled for ABUNREAD. No attempt is made to retrieve the object from either backup copy of the object and the retrieve will fail.

ABOFFLINE
Specifies that when a retrieve for an object is attempted and the optical or tape volume on which the object resides is in a library that is offline or pending offline, the specified backup copy of the object is retrieved.

Valid values for ABOFFLINE are:

BACKUP1
Specifies that when a primary copy of an object that is read by an application is not available for the specified reason, and BACKUP1 is specified for ABOFFLINE, an attempt is made to retrieve the object from the first backup copy of the object.

BACKUP2
When a primary copy of an object that is read by an application is not available for the specified reason, and BACKUP2 is specified for ABOFFLINE, an attempt is made to retrieve the object from the second backup copy of the object.

INACTIVE
When a primary copy of an object that is read by an application is not available for the specified reason, and INACTIVE is specified, then automatic access to backup is disabled for ABOFFLINE. No attempt is made to retrieve the object from either backup copy of the object and the retrieve will fail.

ABNOTOPER
Specifies that when a retrieve for an object is attempted and the optical or tape volume on which the object resides is in a library that is marked non-operational, the specified backup copy of the object is retrieved.

Valid values for ABNOTOPER are:

BACKUP1
Specifies that when a primary copy of an object that is read by an application is not available for the specified reason, and BACKUP1 is specified for ABNOTOPER, an attempt is made to retrieve the object from the first backup copy of the object.

BACKUP2
When a primary copy of an object that is read by an application is not available for the specified reason, and BACKUP2 is specified for ABNOTOPER, an attempt is made to retrieve the object from the second backup copy of the object.

INACTIVE
When a primary copy of an object that is read by an application is not available for the specified reason, and INACTIVE is specified, then automatic access to backup is disabled for ABNOTOPER. No attempt is made to retrieve the object from either backup copy of the object and the retrieve will fail.

ABDB2ERROR
Specifies that if a DB2 error occurs while OAM is retrieving object data from the 4 KB, 32 KB, or LOB storage table and the first or second backup copy exists, OAM retrieves the object data from the specified backup copy. This function allows access to backup copies of objects that reside on removable media (optical or tape) when the DB2 resident data is unavailable, such as during the recovery of DB2 tables.

Restriction: The object directory entry is necessary for OAM to proceed with any object request. If a DB2 error occurs while OAM attempts to retrieve the object directory entry, OAM does not retrieve the backup copy of the object. Without the object directory information, OAM cannot determine the primary or backup location of the object.

Valid values for ABDB2ERROR are:

BACKUP1
Specifies that when a primary copy of an object that is read by an application is not available for the specified reason, and BACKUP1 is specified for ABDB2ERROR, an attempt is made to retrieve the object from the first backup copy of the object.

BACKUP2
When a primary copy of an object that is read by an application is not available for the specified reason, and BACKUP2 is specified for ABDB2ERROR, an attempt is made to retrieve the object from the second backup copy of the object.

INACTIVE
When a primary copy of an object that is read by an application is not available for the specified reason, and INACTIVE is specified, then automatic access to backup is disabled for ABDB2ERROR. No attempt is made to retrieve the object from either backup copy of the object and the retrieve will fail.

ABLOST
Specifies that when a retrieve for an object is attempted and the optical or tape volume on which the object resides is marked lost or not-defined, the specified backup copy of the object is retrieved.

Valid values for ABLOST are:

BACKUP1
Specifies that when a primary copy of an object that is read by an application is not available for the specified reason, and BACKUP1 is specified for ABLOST, an attempt is made to retrieve the object from the first backup copy of the object.

BACKUP2
When a primary copy of an object that is read by an application is not available for the specified reason, and BACKUP2 is specified for ABLOST, an attempt is made to retrieve the object from the second backup copy of the object.

INACTIVE
When a primary copy of an object that is read by an application is not available for the specified reason, and INACTIVE is specified, then automatic access to backup is disabled for ABLOST. No attempt is made to retrieve the object from either backup copy of the object and the retrieve will fail.

ABFSERROR
Specifies that if an error occurs while OAM is retrieving an object from the file system and the first or second backup copy exists, OAM retrieves the object data from the specified backup copy. This function allows access to backup copies of objects that reside on removable media (optical or tape) when the file system data is unavailable, such as the file system being unmounted or otherwise unavailable such as for maintenance procedures including backup of the file system data.

Valid values for ABFSERROR are:

BACKUP1
Specifies that when a primary copy of an object that is read by an application is not available for the specified reason, and BACKUP1 is specified for ABFSERROR, an attempt is made to retrieve the object from the first backup copy of the object.

BACKUP2
When a primary copy of an object that is read by an application is not available for the specified reason, and BACKUP2 is specified for ABFSERROR, an attempt is made to retrieve the object from the second backup copy of the object.

INACTIVE
When a primary copy of an object that is read by an application is not available for the specified reason, and INACTIVE is specified, then automatic access to backup is disabled for ABFSERROR. No attempt is made to retrieve the object from either backup copy of the object and the retrieve will fail.

ABALL
Specifies that when a retrieve for an object is attempted and the optical or tape volume on which the object resides is not available for any of the above reasons, the specified backup copy of the object is retrieved.

Valid values for ABALL are:

BACKUP1
Specifies that when a primary copy of an object that is read by an application is not available for the specified reason, and BACKUP1 is specified for one of the above keywords for automatic access to backup, an attempt is made to retrieve the object from the first backup copy of the object.

BACKUP2
When a primary copy of an object that is read by an application is not available for the specified reason, and BACKUP2 is specified for one of the above keywords for automatic access to backup, an attempt is made to retrieve the object from the second backup copy of the object.

INACTIVE
When a primary copy of an object that is read by an application is not available for the specified reason, and INACTIVE is specified, then automatic access to backup is disabled for all of the above reasons. No attempt is made to retrieve the object from either backup copy of the object and the retrieve will fail.

Note: There are no default values for the automatic access to backup (AAB) CBROAMxx PARMLIB SETOPT keywords at the time of OAM initialization. If no AAB keywords are specified in the CBROAMxx PARMLIB SETOPT statement, then the preexisting values for each of the AAB reasons will be retained. If one or more AAB keywords are specified in the CBROAMxx PARMLIB SETOPT statement, then the associated AAB reasons will be set as specified, and all unspecified AAB reasons will be set to inactive. A system IPL will disable all AAB reasons.

MULTISYSENABLE
Specifies whether object and object backup storage groups can be enabled to more than one system in a non-OAMplex environment.

Valid values for MULTISYSENABLE are:

NO
Specifies that when a storage group has been encountered in a NON-OAMplex environment that is enabled to more than one system, the storage group will not be defined to the OAM address space. OSREQ requests for DB2 sublevel resident objects will be processed. No other OAM or OSREQ function will be honored for objects in the storage group. The default MULTISYSENABLE specification is NO.

YES
Specifies that when a storage group has been encountered in a non-OAMplex environment that is enabled to more than one system, the storage group will be defined to OAM.

Note: This is a global specification only and cannot be specified at the storage group level. Additionally, the SETOPT MULTISYSENABLE specification cannot be modified using the MODIFY OAM,UPDATE command, the MODIFY OAM,RESTART operator command, or an OAM restart resulting from an SCDS activation.

Attention: The SETOPT statement of CBROAMxx PARMLIB member provides the MULTISYSENABLE keyword at a global level to bypass OAM's default behavior and to instead allow object or object backup storage groups to be enabled to more than one system in a non-OAMplex SYSplex environment. This configuration consists of a shared SCDS, but non-shared DB2 and primarily allows for the same object or object backup storage group name to be used on multiple systems. If MULTISYSENABLE(YES) is specified in CBROAMxx PARMLIB, then message CBR0165I will be issued to warn the installation when a storage group has been encountered that is enabled to more than one system in a non-OAMplex environment and that the storage group(s) will be defined to OAM. This message will be issued only once per OAM initialization.

In a non-OAMplex environment, the SMS Storage Group Status panel is used to allow status to be specified for each system defined to SMS. Currently, you must specify an option other than NOTCON for the one system that will be running OAM and you must specify the NOTCON option for all other systems. If you specify more than one system as other than NOTCON for an object or object backup storagegroup, a CBR0162I message is issued during OAM initialization and the storage group is ignored by OAM.

In an OAMplex environment, it is normal to have an object or object backup storage group that is defined as enabled to multiple systems. In fact, they are normally enabled to all the systems in the OAMplex. This is because the OAM members in the OAMplex share OAM tables in DB2 and it is desirable for all members of the OAMplex to have equal access to the same data.

In a non-OAMplex environment, instances of OAM on individual LPARs are independent of each other. They do NOT share common DB2 tables and therefore the object data accessible to OAM on one LPAR is not accessible to OAM on another LPAR. The default behavior is to not allow a given object or object backup storage group to be defined as enabled to more than one system when OAM is initialized in stand-alone (non-OAMplex) mode.

Some installations desire to have the same storage group name enabled on multiple systems even though not running in an OAMplex. These installations might have a given storage group name (GROUP22 for example) which contains *ABC-payroll object data* in the DB2 tables accessible to OAM running on LPAR #1, and contains *XYZ-insurance policy object data* in the DB2 tables accessible to the OAM running on LPAR #2.

The SETOPT MULTISYSENABLE keyword available in the CBROAMxx PARMLIB member provides the customer the option to define a given object or object backup storage group as enabled to more than one system in a non-OAMplex environment. In a non-OAMplex when MULTISYSENABLE(YES) is specified, then OAM will issue a CBR0165I informational message during OAM initialization if one or more object or object backup storage groups are defined as enabled to multiple systems in the active SMS configuration.

The MULTISYSENABLE option benefits installations with stand-alone (non-OAMplex) instances of OAM on multiple systems in an SMSplex, but desire to use a common object or object backup storage group name across systems.

Note: Even though the storage group name is defined as enabled for multiple systems, the DB2 tables are unique to each system and therefore the object data is different for each system.

STORAGEGROUP(*name*)
Specifies the name of an Object or Object Backup storage group that was previously defined using ISMF. The following subparameter can also be specified for this storage group:

OPTICALREINITMODE
Reinitialization mode for rewritable optical cartridges belonging to this Object or Object Backup storage group. If this keyword is not specified for a given storage group, the reinitialization mode for rewritable optical cartridges belonging to that storage group is set using the OPTICALREINITMODE set at the global level.

A cartridge's reinitialization mode is set according to any OPTICALREINITMODE option in effect when OSMC Shelf Space Manager selects the cartridge for reinitialization—not according to the options in effect when the optical cartridge is physically reinitialized.

Valid values for the OPTICALREINITMODE option are:

GROUP
Expired rewritable optical cartridges remain assigned to the original Object or Object Backup storage group when reinitialized. This option is the default.

OAMSCRATCH
Expired rewritable optical cartridges belonging to this Object or Object Backup storage group revert to *SCRTCH* storage group when reinitialized. These cartridges are available to be reassigned to any Object or Object Backup storage group.

Note: In a multiple OAM configuration, OPTICALREINITMODE is not supported and will be ignored if specified.

SETOSMC statements for use in the OSMC environment

The SETOSMC statement and its associated keywords of the CBROAM*xx* PARMLIB member determine the valid values of settings for various OSMC processing. They associate an Object storage group with the Object Backup storage groups that store the first or second backup copies of objects. The SETOSMC statement determines which Object Backup storage groups are to contain the first and second copies of the objects that are associated with an Object storage group. If you do not provide global or a storage group level SETOSMC SECONDBACKUPGROUP(secondbackupgroupname) statement, OAM cannot create second backup copies.

You can create or update this PARMLIB member for second backup copies of objects. Figure 15 on page 139 shows examples of SETOSMC statements in the CBROAM*xx* PARMLIB member that you can use as samples for your installation. You can specify both the STORAGEGROUP subparameters FIRSTBACKUPGROUP and SECONDBACKUPGROUP in a single SETOSMC STORAGEGROUP statement, as shown in the second example in Figure 15 on page 139. The descriptions of the keywords are found in the discussion of the "SETOSMC keyword definitions" on page 140.

Related reading: For information about dynamically updating SETOSMC parameters, see "Updating SETOSMC values" on page 371.

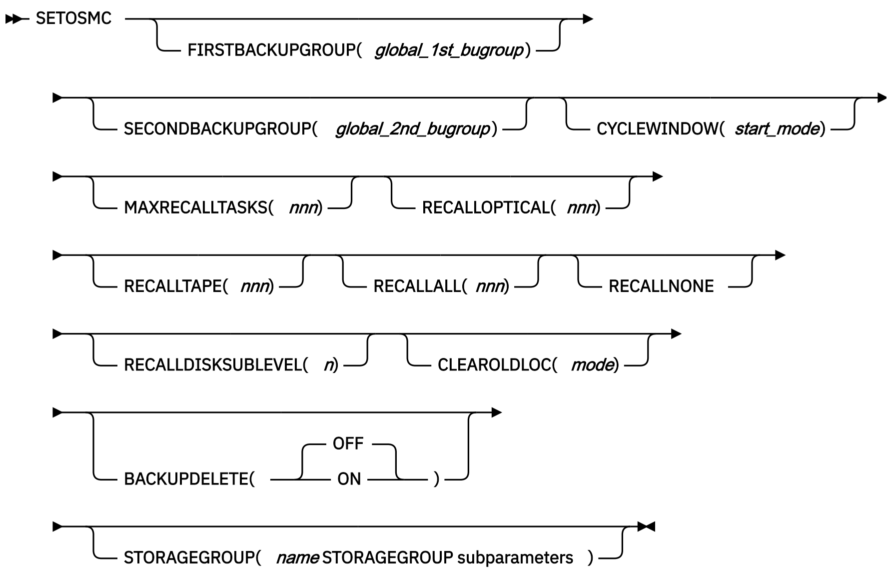
```
SETOSMC FIRSTBACKUPGROUP(global_1st_bugroup)
        SECONDBACKUPGROUP(global_2nd_bugroup)
        STORAGEGROUP(name
                     FIRSTBACKUPGROUP(1st_bugrp))
        STORAGEGROUP(name
                     SECONDBACKUPGROUP(2nd_bugrp))
        CYCLEWINDOW(STARTSTOP)

              OR

SETOSMC STORAGEGROUP(name
                     FIRSTBACKUPGROUP(1st_bugrp)
                     SECONDBACKUPGROUP(2nd_bugrp))
```

Figure 15. CBROAMxx PARMLIB Member Samples Using the SETOSMC Statement and Optional Parameters

Notes:

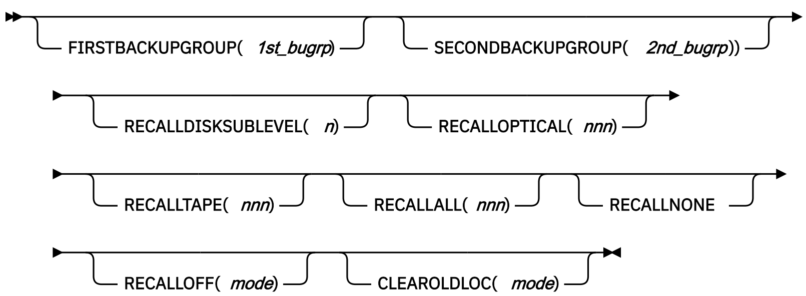
1. In a multiple OAM configuration, keywords and values related to optical processing are not supported on the SETOSMC statements. If these keywords or values are specified in a multiple OAM configuration, they are ignored.
2. You can use the SETOSMC statement at both the global level and at the storage group level. If you specify parameters without a *name* value, OAM provides the defaults for all Object storage groups in the configuration. If you specify parameters with a *name* value, OAM provides the specific Object Backup storage groups to be used for that Object storage group's backup copies.

The syntax for the SETOSMC statement follows.



SETOSMC statement syntax: OAM global level parameters

STORAGEGROUP subparameters



SETOSMC keyword definitions

The following keywords are defined as they pertain to the CBROAM*xx* PARMLIB member SETOSMC statement:

FIRSTBACKUPGROUP(*global_1st_bugroup*)

Specifies the default Object Backup storage group that OSMC uses to store the first backup copy of objects when:

- The Object storage group to which the object belongs is not specified in a SETOSMC statement on the FIRSTBACKUPGROUP parameter, and
- The management class that is assigned to the object specifies that a backup copy is to be written.

SECONDBACKUPGROUP(*global_2nd_bugroup*)

Specifies the default Object Backup storage group that OSMC uses to store the second backup copy of objects when:

- The Object storage group to which the object belongs is not specified in a SETOSMC statement on the SECONDBACKUPGROUP parameter, and
- The management class that is assigned to the object specifies that more than one backup copy is to be written.

CYCLEWINDOW(*start_mode*)

Specifies the start window mode for the OSMC storage management cycle for a given Object or Object Backup storage group. You can choose either the default STARTONLY mode or the STARTSTOP mode for the CYCLE START and CYCLE STOP times. The START/ONLY mode defines only the start time for processing the storage group. The START/STOP mode defines both the start and end times for processing the storage group. You can use this keyword at the global level only. The CYCLEWINDOW keyword is valid for storage groups that were started automatically.

To start the storage management cycle for an Object storage group automatically, specify the Cycle Start Time, End Time, and OSMC Processing System fields in the ISMF Object Storage Group Define/Alter panel. For an Object Backup storage group, specify the Cycle Start Time, End Time, and OSMC Processing System fields in the ISMF Object Backup Storage Group Define/Alter panel.

Restriction: If you manually start the storage management cycle by issuing one of the following commands, the CYCLEWINDOW mode and the start and stop times for the storage group cycle are ignored:

- MODIFY OAM,START,STORGRP,*group_name*
- MODIFY OAM,START,OSMC

STARTONLY
Uses the storage group's defined cycle start and end times as a start window only. OSMC starts processing the storage group at the defined start time. If you initialize OSMC before or during the times that this start window specifies, processing for that storage group completes, regardless of how much time it takes. If you omit the CYCLEWINDOW parameter, STARTONLY is the default mode.

STARTSTOP
Uses the storage group's defined cycle start and end times to automatically begin processing that storage group. OSMC starts processing the storage group at the defined start time and stops processing the storage group at the defined end time. Processing stops in the same way as if you had issued a STOP command for that storage group. No new work is scheduled, and all work in progress is allowed to complete.

STORAGEGROUP(*name*)
Specifies the name of an Object storage group that was previously defined using ISMF. This is the name of the storage group to which the following subparameters apply:

FIRSTBACKUPGROUP(*1st_bugrp*)
Specifies the Object backup group to which the first backup copy of an object belonging to the Object storage group *name* is directed when that object is associated with a management class that specifies that a backup copy is to be written.

SECONDBACKUPGROUP(*2nd_bugrp*)
Specifies the Object backup group to which the second backup copy of an object belonging to the Object storage group *name* is directed when that object is associated with a management class that specifies that more than one backup copy is to be written.

MAXRECALLTASKS(*nnn*)
Specifies the maximum number of RECALL tasks that can be run concurrently. Valid values are 0–255. The default is 0. A value of 0 indicates that no RECALL operations are to be run. This applies to both implicit (recalls through SETOSMC enablement) and explicit (recalls through OSREQ macro requests with the RECALL keyword) recalls.

RECALLDISKSUBLEVEL(*n*)
Specifies the disk sublevel where objects residing on optical or tape devices will be placed when they have been recalled. This keyword can be specified at the global level or at the storage group level and applies to both explicit and implicit recalls. This parameter may be abbreviated to RECALLD. Valid values for *n* are:

1
Specifies that objects will be recalled to disk sublevel 1 of the OAM storage hierarchy (DB2 object storage tables). This is the default value.

2
Specifies that objects will be recalled to disk sublevel 2 of the OAM storage hierarchy (file system)

RECALLOPTICAL(*nnn*)
Specifies that objects residing on optical devices are recalled to disk for *nnn* days when retrieved. Valid values are 0–255. The default is 0. The object's pending action date in the object directory is set

to today's date + the number of days specified in *nnn*. A value of 0 indicates that the RECALL is for the current day only. On the next OSMC cycle the object is restored to removable media. This applies to implicit recalls only. This keyword can be specified at the global level or at the storage group level.

Note: RECALLOPTICAL applies only in a classic OAM configuration. If RECALLOPTICAL is specified in a multiple OAM configuration, it will be ignored.

RECALLTAPE(*nnn*)
Specifies that objects residing on tape devices are recalled to disk for *nnn* days when retrieved. Valid values are 0–255. The default is 0. The object's pending action date in the object directory is set to today's date + the number of days specified in *nnn*. A value of 0 indicates that the RECALL is for the current day only. On the next OSMC cycle the object is restored to removable media. This applies to implicit recalls only. This keyword can be specified at the global level or at the storage group level.

RECALLALL(*nnn*)
Specifies that objects residing on optical or tape devices are recalled to disk for *nnn* days when retrieved. Valid values are 0–255. The default is 0. The object's pending action date in the object directory is set to today's date + the number of days specified in *nnn*. A value of 0 indicates that the RECALL is for the current day only. On the next OSMC cycle the object is restored to removable media. This applies to implicit recalls only. This keyword can be specified at the global level or at the storage group level.

RECALLNONE
Specifies that objects residing on optical or tape devices are not recalled to disk when retrieved. This applies to implicit recalls only. This keyword can be specified at the global level or at the storage group level.

RECALLOFF(*mode*)
Specifies whether objects residing on tape or optical devices are to be recalled to DB2 when retrieved. This keyword can be specified at the storage group level only, and allows recall processing to be disabled at the storage group level regardless whether the RECALL parameter is specified on the OSREQ RETRIEVE request. This keyword applies to both implicit and explicit recalls. Valid values for *mode* are:

OFF
Explicit and implicit recalls are enabled. This is the default.

ON
Explicit and implicit recalls are disabled.

CLEAROLDLOC(*mode*)
Specifies whether OAM is to retain the original volume location information when OSMC processing transitions objects from optical or tape media to the DB2 sublevel or file system sublevel. Values for *mode* are:

OPT
Specifies that previous volser and sector location values be cleared in the object directory when an optical object transitions from optical media to DB2.

Note: A value of OPT for CLEAROLDLOC cannot be specified in a multiple OAM configuration.

TAPE
Specifies that previous volser and blockid location values be cleared in the object directory when an object transitions from tape media to DB2 or FileSystem.

BOTH
Specifies that previous volser and sector or blockid location values be cleared in the object directory when an optical or tape object transitions from tape or optical media to DB2 or FileSystem.

Note: In a multiple OAM configuration, a value of BOTH for CLEAROLDLOC is ignored for optical and thus has the same effect as specifying TAPE.

NONE

This is the default. Specifies that previous volser and sector or blockid location values be left unchanged in the object directory when an object transitions from tape or optical media to DB2 or FileSystem.

Old location values are cleared only when an object transitions from optical or tape to the DB2 sublevel or file system sublevel during an OSMC cycle. The old location values are not cleared when an object is recalled to the DB2 sublevel or file system sublevel, even if CLEAROLDLOC is active.

BACKUPDELETE(ON|OFF)

An optional parameter that specifies whether OSMC should delete any existing backup copies of an object that exceed the required number of backup copies for the object's current management class. (For example, if two backup copies exist, but the management class only requires one, the second backup copy is deleted.) The default value is OFF, which means that all existing backup copies are kept.

Note: This is a global specification only and cannot be specified at the storage group level. Additionally, the BACKUPDELETE specification cannot be modified using the MODIFY OAM,UPDATE command.

SETTLIB statement for tape library settings

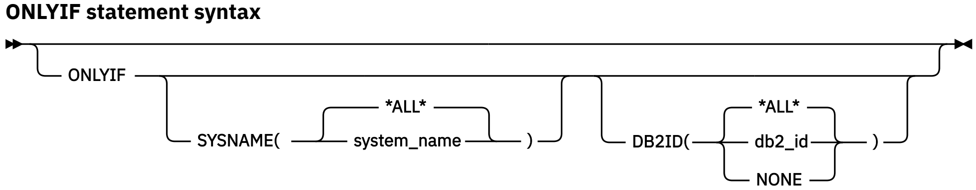
The optional SETTLIB statement and its associated keywords in the CBROAMxx PARMLIB member can be used to configure various tape library related settings. For more information, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*

ONLYIF statements

The ONLYIF statement can be used to specify whether specific statements within the CBROAM*xx* PARMLIB member are to be processed on a given system and, further, for a given system to determine whether statements are to be processed by a given OAM address space, identified by its association with the DB2ID specified. The scope of the ONLYIF statement is in effect until the next ONLYIF is encountered.

The syntax for the ONLYIF statement follows.

ONLYIF statement syntax



ONLYIF

Note: The ONLYIF statement and each of the OAMXCF, SETOAM, SETOPT, SETDISK, SETOSMC OR SETTLIB statements that follow it must be on separate lines.

SYSNAME

Specifies the system name on which the set of OAMXCF, SETOAM, SETOPT, and SETOSMC statements that follow are processed. Valid values are 1-8 character system name or the reserved string *ALL*. If a *system_name* is specified, then the following statements are only processed if the system OAM is initializing on has a matching system name. The system name is defined by the SYSNAME parameter in the IEASYM*xx* or IEASYS*xx* PARMLIB members. If *ALL* is specified, the statements are processed on all systems. If the SYSNAME keyword is not specified, the default value is *ALL*.

DB2ID

Specifies the DB2 identifier (SSID or Group Attachment Name) associated with the OAM address space for which the set of OAMXCF, SETDISK, SETOAM, SETOPT, SETOSMC, and SETTLIB statements that follow are to be processed. Valid values are a 1-4 character DB2 identifier, NONE, or the reserved string *ALL*. If a *db2_id* is specified, then the following statements are only processed by an OAM address space that was started with a DB2 subsystem with an SSID or

Group Attachment Name of *db2_id*. If NONE is specified, then the following statements are only processed by an OAM tape library address space that was started with D=NONE. If *ALL* is specified, the statements are processed by all OAM address spaces. If the DB2ID keyword is not specified, the default value is *ALL*.

Note: By using SYSNAME(*ALL*) and a DB2 Group Attachment Name for DB2ID, the same set of following statements are processed for all members of an OAMplex.

In the following example, OAMXCF OAMMEMBERNAME(OAM1) is processed only if the system is SYS1 and OAMXCF OAMMEMBERNAME(OAM2) is processed only if the system is SYS2. All of the statements after ONLYIF SYSNAME(*ALL*) are processed on all systems but the STORAGEGROUP statement for GROUP02 is only processed by OAM instances associated with a DB2 subsystem that has DBG2 as either the SSID or Group Attachment Name.

```
ONLYIF SYSNAME(SYS1)
OAMXCF OAMMEMBERNAME(OAM1)

ONLYIF SYSNAME(SYS2)
OAMXCF OAMMEMBERNAME(OAM2)

ONLYIF SYSNAME(*ALL*)
OAMXCF OAMGROUPNAME(OPLEX01)

SETOAM PERCENTVALID (99) MAXRECYCLETASKS(3)
            MAXTAPESTORETASKS(1) MAXTAPERETRIEVETASKS(1)
            TAPEDISPATCHERDELAY(3)
            DSNWITHSGNAME
            TAPEEXPIRATION(2009/364)
            MOUNTWAITTIME(1)
            DEMOUNTWAITTIME(3)
            TAPEFULLTHRESHOLD(9999)
            TAPERECYCLEMODE(OAMSCRATCH)
            DATACLASS(VTSM2CU)

    STORAGEGROUP(GROUP01 TAPEUNITNAME(3490) L2TAPEUNITNAME(3490)
        L2DATACLASS(VTSM2CU)
        NOTAPECOMPACTION SGMAXTAPESTORETASKS(1)
        SGMAXTAPERETRIEVETASKS(1))

ONLYIF DB2ID(DBG2)

    STORAGEGROUP(GROUP02 TAPEUNITNAME(3490) L2DATACLASS(VTSM2CU)
        L2TAPEUNITNAME(3490)
        NOTAPECOMPACTION SGMAXTAPESTORETASKS(1)
        SGMAXTAPERETRIEVETASKS(1))
```

Note: The value of an ONLYIF statement cannot be displayed with the F OAM,DISPLAY command or updated with the F OAM,UPDATE command.

OAMXCF statements in an OAMplex

To use OAM in a Parallel Sysplex, in which multiple OAMs share a common DB2 database and communicate between instances of OAM, it is necessary for each instance of OAM that is to be part of an OAMplex to join an XCF group. Creating or updating the CBROAM*xx* PARMLIB member to include the OAMXCF statements is required to assign OAM instances to an XCF group. Each OAM that joins the group must supply the XCF group name and a member name for that instance of OAM in the sysplex. Keywords for the OAMXCF statement for that instance of OAM in the OAMplex are provided in the CBROAM*xx* PARMLIB member that allow specification of the XCF group and member name to become part of the XCF group. All instances of OAM that join the same XCF become an OAMplex. The scope of the OAMplex must match the scope of the DB2 sharing group. See "Using the UPDATE command to set OAMXCF values" on page 373 for information on changing the OAMXCF values dynamically and on defining the values when the CBROAM*xx* PARMLIB member is not used at initialization.

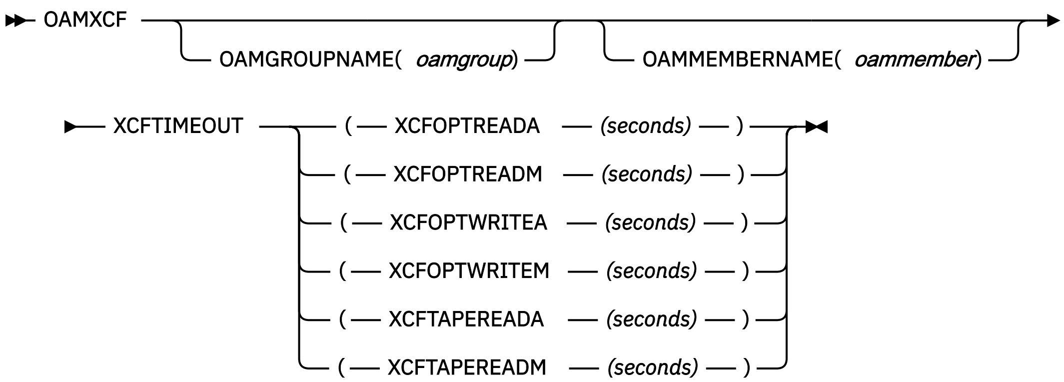
In the Parallel Sysplex, using transaction shipping through XCF, it is possible for a transaction to take too long to complete or for a response to never be returned for completion. OAM fails requests after a certain amount of time in order to free a user from a wait state. The XCFTIMEOUT keyword, available in the OAMXCF statement, allows you to customize timeout values for your environment. These various timeout values (in seconds) specify the length of time that an instance of OAM in the sysplex is to wait for completion of a transaction that was shipped to another instance of OAM in the sysplex. If a response is not received within the specified timeout value, OAM fails the request, returning a nonzero return and reason code to the caller. Many factors determine the expected response time for a transaction, such as optical compared to tape, reads compared to writes, and automated compared to manual environments. OAM provides different timeout values for different transaction types and environments.

Related reading: For the procedure for adding OAMs to an OAMplex, see "Adding OAM systems to an existing OAMplex" on page 180.

Figure 18 on page 181 is an example of an OAMXCF statement in the CBROAM*xx* PARMLIB member that can be used as a sample in your installation. The descriptions of these keywords are found in the discussion of the "OAMXCF keyword definitions" on page 145.

The syntax for the OAMXCF statement follows.

OAMXCF statement syntax



OAMXCF keyword definitions

The following keywords are defined as they pertain to OAMXCF statements:

OAMGROUPNAME

Identifies the XCF group name that all instances of OAM within this OAMplex are to join. An XCF group name is 1–8 characters long. Valid characters are A–Z, 0–9, and national characters ($, #, @). If OAMGROUPNAME is specified, then OAMMEMBERNAME becomes a required keyword. If OAMXCF statements exist in the CBROAM*xx* PARMLIB member, both OAMMEMBERNAME and OAMGROUPNAME are required.

OAMMEMBERNAME

Identifies the specific XCF member name that is to be associated with this instance of OAM in the Parallel Sysplex, when this OAM joins the OAM group in the sysplex. An XCF member name is 1–16 characters long. Valid characters are A–Z, 0–9, and national characters ($, #, @). If OAMMEMBERNAME is specified, then OAMGROUPNAME becomes a required keyword. If OAMXCF statements exist in the CBROAM*xx* PARMLIB member, both OAMMEMBERNAME and OAMGROUPNAME are required.

XCFTIMEOUT

Identifies the number of seconds (1 to 999 999) that this instance of OAM waits for a response that indicates the completion of a shipped transaction from another instance of OAM in an OAMplex.

Tip: Seconds=0 indicates that OAM is to wait indefinitely for a shipped transaction completion response from another instance of OAM in the OAMplex. This value is the default. The maximum value is 999 999 seconds, which is approximately 11.5 days.

The following are valid timeout values:

(XCFOPTREADA(*seconds*))

Indicates the number of seconds that an OAM originating an optical read request, which is shipped to another OAM within the OAMplex that owns the library where the object resides for processing, should wait for completion of the read request.

(XCFOPTREADM(*seconds*))
: Indicates the number of seconds that an OAM originating an optical read request for a shelf-resident volume, which is shipped to another OAM within the OAMplex that owns the library where the object resides for processing, should wait for completion of the read request.

(XCFOPTWRITEA(*seconds*))
: Indicates the number of seconds that an OAM originating an optical write request targeted for an object storage group that contains real (automated) optical libraries, which is shipped to another OAM within the OAMplex that owns the optical library defined to the object storage group for processing, should wait for completion of the write request.

(XCFOPTWRITEM(*seconds*))
: Indicates the number of seconds that an OAM originating an optical write request targeted for an object storage group that contains pseudo libraries, which is shipped to another OAM within the OAMplex that owns the pseudo library defined to the object storage group for processing, should wait for completion of the write request.

(XCFTAPEREADA(*seconds*))
: Indicates the number of seconds that an OAM originating a tape read request targeted for an automated tape library dataserver, which is shipped to another OAM within the OAMplex that owns the library in which the object resides for processing, should wait for completion of the read request.

(XCFTAPEREADM(*seconds*))
: Indicates the number of seconds that an OAM originating a tape read request targeted for an MTL, which is shipped to another OAM within the OAMplex that owns the library in which the object resides for processing, should wait for completion of the read request.

Note: In a multiple OAM configuration, these keywords are not supported and ignored if specified.

5h Updating the PROCLIB

You must perform this step both for migration and at initial installation.

Sample jobs are provided in SAMPLIB to assist you in making the needed additions to PROCLIB. Before running each SAMPLIB member:

- Update the JOB statement.
- Verify that the system name that is specified is the same as the name of your DB2 subsystem.
- Ensure that the high-level qualifier on the `//OUT DD JCL` statement matches the naming standard at your installation.

5h1 *Modify, if necessary, then run CBRIPROC SAMPLIB job.*

You must perform this step for migration and at initial installation.

Change and run SAMPLIB member CBRIPROC (see "CBRIPROC" on page 423) to create member OTIS in PROCLIB.

If the DB2 load module library containing DSNALI is not in the LNKLST concatenation, either include the DB2 load module library in the SYS1.LINKLIB concatenation (LNKLST*xx*) or add a STEPLIB DD to this procedure.

If a STEPLIB is used, then that concatenation must be APF-authorized.

5h2 *Modify, if necessary, then run CBRAPROC SAMPLIB job.*

This step is performed at initial installation and for migration. Perform this step only if you need to start the OAM address space. The OAM address space is needed for writing objects to tape or optical storage, using a file system, or if you are running OSMC for expiration processing. Do not perform this step if you are using only DB2 sublevel storage (no file system, optical volumes, or tape devices) and you are not running OSMC.

If the installation will use OAM objects, modify and run SAMPLIB member CBRAPROC (see "CBRAPROC" on page 422) to create member OAM in PROCLIB.

If you do not modify CBRAPROC, the following member is created as the default:

```
//OAM PROC OSMC=YES,MAXS=2,UNLOAD=9999,EJECT=LRW,REST=YES
//IEFPROC EXEC PGM=CBROAM,REGION=0M,
//PARM=('OSMC=&OSMC,APLAN=CBROAM,MAXS=&MAXS,UNLOAD=&UNLOAD',
//       'EJECT=&EJECT,RESTART=&REST')
//SYSABEND DD SYSOUT=A
```

If you want to have access to SETOAM, SETOPT, SETOSMC, SETDISK, SETTLIB, and OAMXCF statements in the PARMLIB member (required for many functions including writing to tape volumes, using an OAMplex, multiple backups, and more), you must update this job step to include 'OAM=&OAM'. You must also supply the default OAM=*xx* (where *xx* is the low order suffix of your CBROAM*xx* PARMLIB member) specification on the PROC statement as the following example indicates:

```
//OAM PROC OSMC=YES,MAXS=2,UNLOAD=9999,OAM=xx,EJECT=LRW,REST=YES
//IEFPROC EXEC PGM=CBROAM,REGION=0M,
//PARM=('OSMC=&OSMC,APLAN=CBROAM,MAXS=&MAXS,UNLOAD=&UNLOAD',
//       'OAM=&OAM,EJECT=&EJECT,RESTART=&REST')
//SYSABEND DD SYSOUT=A
```

If you are using a multiple OAM configuration, you must update this job step to include 'D=xxxx'. You must also supply the default DB2ID=*xxxx* (where *xxxx* is the DB2 SSID or Group Attachment Name) specification on the PROC statement as the following example indicates:

```
//OAM PROC OSMC=YES,MAXS=2,UNLOAD=9999,EJECT=LRW,REST=YES,DB2ID=DB2A
//IEFPROC EXEC PGM=CBROAM,REGION=0M,
//PARM=('OSMC=&OSMC,APLAN=CBROAM,MAXS=&MAXS,UNLOAD=&UNLOAD',
//       'EJECT=&EJECT,RESTART=&REST,D=&DB2ID')
//SYSABEND DD SYSOUT=A
```

Alternatively, in a multiple OAM configuration, you might choose to create and manually edit multiple members in PROCLIB, each with a unique member name and each with a unique D=&D specification (one for a Tape Library OAM address space, if needed, and one or more for each Object OAM address space, if needed).

Note: A symbol other than DB2ID can be used, but the symbol used on the PROC statement to define the default value must be the same as the symbol used after "D=&" in the EXEC statement's PARM string.

With the PARM=*keyword*, you can specify values for the following parameters:

D

(For multiple OAM configuration only) Specifies the 1 - 4 character SSID or Group Attachment Name of the DB2 subsystem that is associated with the OAM address space. For a Tape Library OAM address space in a multiple OAM configuration, specify "NONE". The DB2 identifier is required in a multiple OAM configuration and must match the DB2 identifier that is specified on a prerequisite OAM subsystem definition (in the IEFSSN*xx* member of PARMLIB or via SETSSI command). If the DB2 identifier is not specified or does not match a DB2 identifier on an OAM subsystem definition (in the IEFSSN*xx* member of PARMLIB or via SETSSI command) in a multiple OAM configuration, then OAM address space initialization terminates. Note that only one OAM address space can be started on a system for each DB2 identifier.

In a classic OAM configuration, specification of D= is invalid and causes the OAM address space initialization to terminate.

APLAN

Specifies the name of the DB2 application plan for LCS. CBROAM is the name of the DB2 application plan that is specified when an SQL BIND command was issued for the LCS OAM configuration database request modules (DBRMs).

EJECT

Specifies which volumes are ejected from an optical library when the library is full and there is a request to add additional volumes to the library. The valid parameter values on this keyword are as follows:

- LRW—least recently written date for the volume. The volumes are ejected based on how long it has been since an object has been written on the volumes, regardless of how often objects are being

read from the volume. The volumes that have the oldest least recently written date are ejected to make room for the additional volumes requested. This is the default parameter.
- LRM—least recently mounted date for the volume. The volumes that are in the slot for the longest time without being mounted are ejected to make room for the additional volumes requested.

MAXS

The MAXS parameter in the OAM cataloged procedure specifies the maximum number of Object or Object Backup storage group tasks that can be processed concurrently during an OSMC storage management cycle. If storage management cycles for groups overlap, you can use the MAXS parameter to limit resource consumption. When determining the value for MAXS, consider the number of optical and tape drives that are available for storage management processing to avoid drive contention. If MAXS is not specified, a default of 2 is assigned. See "OAM cataloged procedure parameter (MAXS)" on page 166 for more information about the MAXS parameter.

Requirement: If concurrent processing includes Object storage groups writing to tape volumes, you must specify the correct corresponding (global level) MAXTAPERETRIEVETASKS and MAXTAPESTORETASKS values on the SETOAM statement. For more information concerning these keywords, see "SETOAM keyword definitions for global level" on page 112.

OAM

OAM=*xx* specifies the suffix of the CBROAM*xx* PARMLIB member that OAM should process during OAM address space initialization. The two alphanumeric characters (*xx*) must immediately follow the OAM=*keyword* in the PARM field. If the two characters immediately following the OAM=*keyword* are invalid or not specified, error message CBR0025I is issued. OAM only reads PARMLIB member CBROAM*xx* if the OAM=*keyword* is specified on the PARM field of the JCL EXEC statement in the OAM cataloged procedure. If no OAM=*keyword* is specified on the PARM field of the JCL EXEC statement, no PARMLIB member is read by OAM and object tape storage is not active. If the object tape storage is not active, OAM cannot read any objects back or write any new objects to tape until OAM is initialized with a valid OAM=*xx* specification, and a valid corresponding CBROAM*xx* PARMLIB member. OAM processes PARMLIB member CBROAM*xx* during OAM address space initialization.

OSMC

Specifies whether to initialize OSMC when OAM initializes. When YES, OSMC initializes when OAM initializes; when NO, OSMC is prevented from initializing when OAM initializes. (Operator commands requiring OSMC do not execute; the storage management cycle does not run.)

RESTART

Specifies whether OAM should automatically restart when it receives notification that a new SCDS is activated. The valid parameter values on this keyword are as follows:

- **Yes**—If RESTART=YES is specified, when OAM is notified that an SCDS activation has occurred, OAM automatically restarts. This is the default value. How soon OAM is notified of the SCDS activation depends on the time interval that is specified with the INTERVAL keyword in the IGDSMS*xx* PARMLIB member.
- **No**—If RESTART=NO is specified, when OAM is notified that an SCDS activation has occurred, OAM continues processing normally. Message CBR0092I is issued to acknowledge that a new SCDS has been activated. The installation should ensure that the OAM RESTART command is issued if an OAM address space restart is necessary. See "Restarting the OAM address space" on page 364 for details concerning this command.

UNLOAD

Unloads the least recently used 3995 optical disk drive, inside an IBM 3995 optical library, after the number of seconds of inactivity specified on the keyword. This unload only occurs if no available optical drives are within this library. That is, there are no empty online and operational drives. Thus, during periods of inactivity, you can cause at least one drive to be ready to accept the next mount request without first having to do a demount.

The valid parameter values on this keyword are as follows:

- 0 - 9998 specifies the number of seconds of inactivity before the demount might occur.

- 9999 specifies that the cartridge is not to be demounted during periods of inactivity. This is the default.

Restriction: The UNLOAD keyword applies only to 3995 optical disk drives that are library-resident.

To process objects greater than 256MB on the file system sublevel or on tape, the OAM address space uses 64–bit addressing and virtual storage above the 2G bar. Additional configuration may be required to specify a MEMLIMIT value that will provide virtual storage above the 2G bar to the OAM address space. Typically, specifying REGION=0M in CBRAPROC as shown in the sample above, implies a MEMLIMIT value that provides the OAM address space with the needed virtual storage above the 2G bar. For more information, see "MEMLIMIT for OAM" on page 35.

5h3 ***Verify or create device numbers.***

You must perform this step both for migration and at initial installation. Do **not** perform this step if you are not using or do not plan to use optical devices.

Define 3995 device numbers, as well as CTC device numbers, using the hardware configuration definition (HCD).

Related reading: For more information, see "Defining 3995 device numbers" on page 390 and *z/OS HCD User's Guide*.

6 Creating DB2 databases for object tables and directories

Sample jobs for creating databases are provided in SAMPLIB to help install OAM. These sample jobs help you create the DB2 databases, create the application plans, grant authority to use the application plans, and access the databases. Before running the sample jobs, you must:

- Change the JOB statement to meet your installation's requirements.
- Verify that the user ID that is specified on the JOB statement has the correct authority to perform the requested operations.
- Add a JOBLIB for the appropriate DSNLOAD if it is not in the linklist.
- Verify that the subsystem name specified with the SYSTEM keyword on the DSN command is the name of your DB2 subsystem.
- Change the plan name on the RUN statement to match your current DB2 version and release level.
- Start DB2.

If you choose to have SMS manage your DB2 VSAM data sets, create an ACS routine for the VSAM DB2 allocations. Enable the Object and or the Object Backup storage group, enable the volumes, and then validate and activate the SCDS.

Note: When planning for larger data object support (greater than 50MB) keep in mind that:

- If large object support (LOB) is not enabled, objects larger than 32640 bytes and less than or equal to 256 MB are stored in multiple rows in the 32 KB DB2 table. For example, a 256 MB object takes more than 8000 rows to store in the 32 KB table. You might have to increase the number of DB2 locks specified (per user and table space) in order to prevent lock escalations and timeouts. Ensure that the tables used to store the larger objects have the required capacity.
- If LOB is enabled, objects larger than 32640 bytes are stored directly into DB2 in a LOB column with each BLOB as a varying length string up to the OAM maximum object size of 2000 MB.
- Objects greater than 256 MB require LOB to be enabled.

Object databases

Before OAM can operate, you must create object storage databases. These databases contain either objects or information about objects. OAM requires a separate object storage database for each storage group.

Understanding object databases

[NOT Programming Interface Information]

This segment documents information that is provided to help you diagnose OAM problems.

OAM allows multiple DB2 databases to be used for object storage. Each object storage database has an object directory table space, a 4 KB Object Storage Table space, and a 32 KB Object Storage Table space. Within each table space is one table (an object directory table, a 4 KB Object Storage Table, and a 32 KB Object Storage Table, respectively). Each database has three indexes into the object directory table and one index into each of the object storage tables. The high-level qualifier on the object storage database must match the high-level qualifier on the object storage definition in the SCDS that was created using ISMF and the high-level qualifier and package name in PBIND.

LOB support adds the LOB storage database structure to the 4 KB and 32 KB object storage table hierarchy. This structure consists of a LOB base table and a LOB auxiliary table. The LOB base table resembles a 32 KB table with the addition of a ROWID column and changing the OTOBJ column datatype from 'long varchar' to BLOB. The LOB auxiliary table contains the actual BLOB object represented by the OTOBJ column in the LOB base table. The LOB auxiliary table is managed exclusively by DB2 and is transparent to OAM.

Note: Reference to the LOB storage structure refers to both the LOB base table and the LOB auxiliary table.

Object storage administration uses one additional database. The object storage administration database name is OAMADMIN. This database contains a management class table space, a storage class table space, and a collection name table space. Within each table space is one table (a management class table, a storage class table, and a collection name table, respectively). The management class table and the storage class table each have one index, and the collection name table has three indexes.

Each OAM DB2 object storage database has its own separately defined set of VSAM data sets. There is one VSAM data set for each table space, and one VSAM data set for each index within the database.

Attention: The information from the OAMADMIN tables is crucial to the operation of OSMC. **IBM strongly advises against altering these tables.**

[End NOT Programming Interface Information]

Understanding the database creation jobs

Three jobs are supplied in SAMPLIB to assist you in defining the databases to DB2 that are required in your installation (CBRISQL0, CBRISQLX, and CBRISQLY). Three jobs are supplied in SAMPLIB to assist in allocating the VSAM data sets needed for each of the object storage databases you require (CBRIALC0, CBRIALCX, CBRIALCY).

The job CBRILOB is provided in SAMPLIB to create the LOB Storage Structure, which is required if LOB support is to be enabled. You must modify CBRILOB to include the desired LOB enabled storage groups. This job is only required for storage groups that are to be LOB enabled.

SAMPLIB member CBRISQL0 job allocates and defines the object storage databases. SAMPLIB member CBRIALC0 allocates the data sets needed for each object storage database. There is no minimum number of databases or views that must be created. Additionally, there is no maximum limit to the number of object storage databases you might have in your installation. These jobs can be run multiple times to define as many object storage groups and data sets as are required to suite your environment.

SAMPLIB members CBRIALCX, CBRIALCY, CBRISQLX, and CBRISQLY allocate the data sets and define the databases required for object administration. These four SAMPLIB jobs must be modified and executed successfully before OSR or OSMC can be used. SAMPLIB members CBRIALCX and CBRIALCY must be run before SAMPLIB members CBRISQLX and CBRISQLY.

Each step number in all the SAMPLIB jobs is unique. Within the jobs that allocate and define the object storage databases, each step number corresponds to the database name qualifier for the data sets being allocated, and to the database name being defined to DB2.

Related reading: For more information on the SAMPLIB jobs, see these topics:

- "CBRIALC0" on page 424
- "CBRIALCX" on page 426

- “CBRIALCY” on page 428
- “CBRILOB” on page 429
- “CBRISQL0” on page 432
- “CBRISQLX” on page 435
- “CBRISQLY” on page 437

For you to use an OAM object storage database, the allocate job step must be executed, followed by the related DB2 database definition job step. Modify to include the required storage groups and then run the CBRIALCX, CBRIALCY and CBRISQLX, CBRISQLY jobs and the database definition job steps (CBRISQL0) that correspond to the allocation job steps previously run (CBRIALC0). For more information, see step 6d.

6a *Add additional steps to the Database Creation Jobs.*

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

After you have identified the databases needed in your installation, locate the SAMPLIB jobs that contain the steps that will allocate the data sets and define the databases. Add steps for each storage group as needed within the jobs. The remaining jobs and steps must be modified before they are executed.

6b *Modify the OAM Data Set Allocation Jobs.*

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

All allocation jobs must be modified as follows:

1. Change the JOB statement to meet your installation’s requirements.

   ____

2. Change *cat_name* to the DB2 catalog name used in your installation.

   ____

Modify individual allocation job steps as follows:

1. Change *vol_ser* to the volume serial numbers of the volumes where the data sets will reside.

   ____

2. Change *pri_alloc sec_alloc* to the number of cylinders (or tracks) needed for the initial size and the secondary extent size for the data set. The entire statement can be changed to TRACKS(*pri_alloc sec_alloc*) if the data set is not expected to be large. This space allocation must be individually determined for each data set needed by the database. See “Estimating resource requirements” on page 71 for information about database space allocation calculations.

   ____

3. Add allocations for each object storage group defined in your installation.

   ____

The allocation job CBRIALCX and CBRIALCY must be modified as above. See, “CBRIALCX” on page 426, and “CBRIALCY” on page 428 for more information.

If you choose to have SMS manage your DB2 VSAM data sets, create an ACS routine for the VSAM DB2 allocations. Enable the storage group, enable the volumes, then validate and activate the SCDS.

6c *Run the OAM Data Set Allocation Jobs.*

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

After the allocation jobs have been modified, they must be run to successful completion before proceeding. If an error occurs within a particular step, correct the error and rerun either the entire job or just the failing job step.

Run CBRIALC0 for as many groups as needed; then, run CBRIALCX and CBRIALCY. (Because the delete/define refers to specific volume serials, the storage class must be defined with the Guaranteed Space = Y attribute.) For more information on these SAMPLIB jobs, see “CBRIALC0” on page 424, “CBRIALCX” on page 426, and “CBRIALCY” on page 428.

6d *Modify the OAM Database Definition Jobs.*

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

After the data sets have been allocated, run the jobs and steps that define the databases to DB2. The jobs and steps related to data set allocation must be modified before they are executed. Sample jobs are shown in Appendix B, “Sample library members,” on page 421 and reside in SAMPLIB.

For all database definition jobs for the object storage databases, change the JOB statement to meet your installation’s requirements.

Modify all database definition job steps as follows:

1. Change "DB2" in the statement DSN SYSTEM(DB2) to the subsystem ID for DB2 in your installation.

   ____

2. Change the data set name in the statement LIB('DB2.RUNLIB.LOAD') to the data set name used for the DB2.RUNLIB.LOAD data set in your installation.

   ____

3. Change *cat_name* to the DB2 catalog name used in your installation. This must be the same name that was used as the *cat_name* in the allocation job.

   ____

4. Change the *auth_id* to the IDs authorized to the respective group.

   ____

5. Change the PLAN name on the RUN statement to match your current DB2 version and release level.

   ____

The database definition jobs CBRISQLX and CBRISQLY must be modified as previously described in the data set allocations job CBRIALCX and CBRIALCY. For OSR or OSMC to run, the CBRISQLX and CBRISQLY jobs must complete successfully.

Related reading: For more information on these SAMPLIB jobs, see “CBRISQLX” on page 435, “CBRISQLY” on page 437, “CBRIALCX” on page 426, and “CBRIALCY” on page 428.

6e *Run the OAM Database Definition Jobs.*

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

Note: Before z/OS V1R7, the CBRISQL0 sample job created UNIQUE indexes OBJDIRX1 and OBJDIRX2. In z/OS V1R7, CBRISQL0 was modified to create these indexes without the UNIQUE attribute to alleviate a potential OAM OSREQ error due to duplicate time stamp (RC8 RSN30020100). If your OAM Database was defined before z/OS V1R7, it is recommended that you drop UNIQUE indexes OBJDIRX1 and OBJDIRX2, and create them as non-UNIQUE per sample job “CBRISQL0” on page 432. See APAR II13964 for more information.

The DB2 subsystem must be active before starting any of the database definition jobs. All object storage database definition job steps and the object administration database definition job must complete successfully.

Run job CBRISQL0 as many times as needed for the groups used in your installation. Then run jobs CBRISQLX and CBRISQLY.

Related reading: For more information on these SAMPLIB jobs, see “CBRISQLX” on page 435 see “CBRISQL0” on page 432, “CBRISQLY” on page 437, and “CBRISQLY” on page 437.

6f ***Modify OAM LOB data set allocation and definition jobs.***

Perform this step only if you intend to enable LOB support.

Modify all database definition job steps as follows:

1. Change *vol_ser* to the volume serials that your target database should reside on.

   ____

2. Change *pri_alloc* and *sec_alloc* to the desired number of cylinders for each particular VSAM ESDS being defined. For example, CYLINDER(*pri_alloc sec_alloc*) may be CYLINDER(200 10).

   ____

3. Change *cat_name* to the name of the catalog you want to use under DB2.

   ____

4. Include the REUSE keyword in the DEFINE CLUSTER command for each data base, if you intend to use the DSN1COPY utility to copy these data bases.

   ____

5. Change *osg_hlq* to the high level qualifier to be used for the object storage group definition and tables. This is the qualifier used on the object storage group define through ISMF and used by OAM and OSR for all access to the object storage group's directories and data tables.

   ____

6. Change *ds_size* to the maximum size allowed for each data set. Refer to the description of the CREATE TABLESPACE statement DSSIZE parameter in IMS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPH2) for limitations.

   ____

7. Change *auth_id* to the identifiers authorized for the respective group.

   ____

8. Change the name in the DSN SYSTEM(DB2) statement to the name of the DB2 Subsystem in your installation.

   ____

9. Change the data set name in the RUN statement LIB('DB2.RUNLIB.LOAD') phrase to the data set name used in your installation for the DB2 RUNLIB.LOAD data set.

   ____

10. Change the PLAN name (DSNTIA71) in the RUN statement to match your current DB2 version and release level.

    ____

11. Add additional job steps, repeating all statements in the STEP00 and STEP01, for each object storage group defined in your configuration. In each repeated step, change the qualifier to match the qualifier for each object storage group.

    ____

Related reading: For more information on this SAMPLIB job, see “CBRILOB” on page 429.

6g ***Run OAM LOB data set allocation and definition jobs.***

Perform this step only if you intend to enable LOB support.

If you have run this job and want to start over again, issue a DROP for each LOB base table and for each base tablespace and auxiliary tablespace that was previously defined in DB2 by this job.

Related reading: For more information on this SAMPLIB job, see “CBRILOB” on page 429.

7 Creating the OAM configuration database

If you plan to start the OAM address space for object support, you must create the OAM configuration database. This database contains the Library table, the Drive table, the Slot table, the Volume table, the Deleted Object table, and the Tape Volume table.

Sample jobs are provided to help you create the OAM configuration database, create the application plans, and grant authority for the application plans to be used.

7a *Modify, if necessary, and then run the CBRSAMPL SAMPLIB job.*

You must perform this step at initial installation. For migration, verify that this step was performed during the original OAM installation. If this step was not performed during the original OAM installation, perform this step when you migrate from a previous release. Do **not** perform this step if you are using only DB2 sublevel storage (no file system, optical volumes, or tape devices), and do not start the OAM address space for processing objects.

Member CBRSAMPL in SAMPLIB (see "CBRSAMPL" on page 438) creates the OAM configuration database.

You must make the following changes before running the job:

- In the CREATE STOGROUP statement, add:
  - Your volume serial numbers (VOLUMES parameter).
  - The name of your ICF catalog (use 6 characters in the VCAT parameter).

The following summarizes the changes you can make:

- Choose a name for the DB2 storage group and use that name for each STOGROUP parameter.
- Change a catalog password (PASSWORD parameter) in the CREATE STOGROUP statement.

Browse the output to ensure that the job completed successfully. Each SQL statement executed should have an SQLCODE of 0. Check that the final statement of the sample job (COMMIT) has executed.

Attention: The OAM configuration database is a single DB2 database with six tables. Avoid creating two OAM configuration databases with two sets of tables. Two OAM configuration databases can be mistakenly created by running sample job CBRSAMPL twice under two different user IDs. The user ID associated with the job is the user ID supplied by the system programmer with the USER= keyword on the JOB statement.

Requirement: Use ISMF to define the tape or optical libraries and optical drives to OAM.

7c *Run the CBRSMR1D SAMPLIB job if you are migrating from any release earlier than z/OS V1R13.*

Do not perform this step at initial installation. You must perform this step when you migrate from any release earlier than z/OS V1R13 DFSMS.

Note: Regardless of whether you intend to exploit the new function, you must modify and run this job.

This job adds the File System Delete Table to the OAM Configuration Database.

8 Merging object tables and OCDB for an OAMplex

8a *Run the CBRSMERG SAMPLIB job.*

Do not perform this step at initial installation. Perform this step when all of the following conditions exist: you are setting up an OAMplex, you have multiple OAMs running on separate MVS images in a sysplex, and you want to merge two or more separate OCDBs. Do not perform this step if you are using only DB2 sublevel storage (no file system, optical volumes, or tape devices).

CBRSMERG performs a database merge of the OAM configuration databases for use with DB2 data sharing in an OAMplex. You might use this sample job or some other DB2 method to perform this database merging. For more information on this SAMPLIB member, see "CBRSMERG" on page 448.

8b *Run the CBRSG100 SAMPLIB job.*

Do **not** perform this step at initial installation. Perform this step only when all of the following conditions exist: you are setting up an OAMplex; you have multiple OAMs running on separate MVS images in a sysplex; and you want to merge two or more separate OAMADMIN tables, object storage databases, or both.

CBRSG100 performs a database merge of OAM databases for use with DB2 data sharing in an OAMplex. You might use this sample job or some other DB2 method to perform the database merging. For more information on this SAMPLIB member, see "CBRSG100" on page 453.

9 Creating and binding DB2 packages

9a *Run the CBRPBIND SAMPLIB job.*

You must perform this step both for migration and at initial installation.

CBRPBIND performs a DB2 BIND of DBRMs to create the packages that are needed to access the OAM Object storage group tables. The use of the DB2 packages allows user defined qualifiers for the object storage groups table definitions. CBRPBIND must be modified and run prior to running the CBRABIND, CBRHBIND, or CBRIBIND jobs. It is written for the existing 100 Object storage groups with high-level qualifiers of GROUP00–GROUP99, so it needs to be modified for your installation requirements.

CBRPBIND now uses the VALIDATE(RUN) instead of the VALIDATE(BIND) statement. For VALIDATE(BIND), DB2 verifies the authorization at bind time. For VALIDATE(RUN), DB2 verifies the authorization initially at bind time, but if the authorization check fails, DB2 rechecks it at run time.

Statements embedded within various OSR and OSMC packages now reference the V_OSM_LOB_BASE_TBL view. However, these views might not be known to DB2 if either LOB support is disabled (default, LOB=N in IEFSSNxx Parmlib member) or LOB support is only partially enabled (LOB=P in IEFSSNxx Parmlib member). Each storage group that does not have the V_OSM_LOB_BASE_TBL created must use VALIDATE(RUN); otherwise, bind authorization occurs at bind time and the bind fails.

However, it might be desirable to use the VALIDATE(BIND) statement for each object storage group that you want to VERIFY that a V_OSM_LOB_BASE_TBL view has indeed been created. This is not required and should only be used as a precautionary measure to ensure that the LOB view has been created. The bind fails if the LOB view is not created.

If you are not using optical storage and have not created or have deleted the DB2 tables related solely to optical storage (OLIBRARY, DRIVE, SLOT, VOLUME, DELOBJT), the BIND statements for packages CBRKCMD, CBRKCME, CBRKCMF, CBRKCMI, CBRKCMR, and CBRKISQL must be removed from the MISCPKG step of CBRPBIND to prevent bind errors.

Related reading: For more information on the CBRPBIND SAMPLIB job, see "CBRPBIND" on page 461.

10 OSR application plans

For OSR to function correctly, you must create the OSR application plan, bind it to DB2, and grant authority for the plan to be used.

10a *Run the CBRIBIND SAMPLIB job.*

If you are not planning to create the OAM configuration database but you do plan to store objects without starting the OAM address space, you must perform this step both for migration and at initial installation. If you plan to create the OAM configuration database, skip this step and proceed to step 11a.

Creating the OSR application plan

You must create the OSR application plan, bind it to DB2, and grant authority for the plan to be used. SAMPLIB members CBRIBIND and CBRIGRNT are provided for this purpose. CBRIBIND binds the packages created in CBRPBIND to the OSR application plan. You need to modify this job to include the high level qualifiers for the installation's storage groups (currently setup for GROUP00-GROUP99). This step is not required if you plan to create the OAM configuration database and start the OAM address space. The OSR application plan for the optical or tape environment is provided as part of the CBRABIND and CBRAGRNT jobs. After execution, the job output should contain the following message about CBRIDBS:

```
DSNT200I BIND FOR PLAN CBRIDBS SUCCESSFUL
```

Recommendation: If you do not plan to use OSMC, see the information on the required application plan CBRHSMSI under ""11 OSMC application plans" on page 156."

Related reading:

- If you do not receive this message, see "OAM Diagnostic Aids" in the *z/OS DFSMSdfp Diagnosis* for more information.
- For information on the DSN FREE subcommand, see IMS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPH2).
- For more information about the DB2 catalog tables, see IMS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPH2).

10b ***Run the CBRIGRNT SAMPLIB job.***

If you are not planning to create the OAM configuration database, but you do plan to store objects without starting the OAM address space and you have previously run CBRIBIND, you must perform this step both for migration and at initial installation. If you plan to create the OAM configuration database, skip this step and proceed to step 11a.

Granting authority

You must grant applications the authority to use plan CBRIDBS and to access the databases. The statements that grant this authority are shown in SAMPLIB. Also see "Examples of Granting and Revoking Privileges" in IMS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPH2).

Completion of a successful grant is indicated by the following message in the job output:

```
DSNT400I SQL CODE=000, SUCCESSFUL EXECUTION
```

Recommendation: If you have applications containing application plans also using the modules identified in the CBRIBIND SAMPLIB job, you must rerun the BIND and GRANT jobs for these *other* applications you have installed as well. The IBM SAA ImagePlus Object Distribution Manager, the IBM Report Data Archive and Retrieval System (RDARS), and the Item Access Facility CICS application (IAFC), are examples of applications that bind OAM and Object Distribution Manager modules into a common plan.

Related reading: If you do not receive this message, see "OAM Diagnostic Aids" in *z/OS DFSMSdfp Diagnosis* for more information.

11 OSMC application plans

To use OSMC, or to store objects using OSREQ , you must create the OSMC application plans, bind them to DB2, and grant authority for those plans to be used. If you are not using OSMC, you must create the application plan for CBRHSMSI, bind it to DB2 and grant authority for it to be used (modify the CBRHBIND and CBRHGRNT sample jobs to include only the statements for CBRHSMSI).

11a ***Run the CBRHBIND SAMPLIB job.***

You must perform this step both for migration and at initial installation.

Creating the OSMC application plans

After creating the databases for OSR, you must create the OSMC application plans. Member CBRHBIND in SAMPLIB is provided for this purpose. After running the job, the following DB2 messages should appear in the job output:

```
DSNT200I BIND FOR PLAN CBRHSMSI SUCCESSFUL
DSNT200I BIND FOR PLAN CBRHOBJP SUCCESSFUL
DSNT200I BIND FOR PLAN CBRHSOBP SUCCESSFUL
DSNT200I BIND FOR PLAN CBRHSVOL SUCCESSFUL
DSNT200I BIND FOR PLAN CBRHSBKV SUCCESSFUL
```

```
DSNT200I BIND FOR PLAN CBRHRDAS SUCCESSFUL
DSNT200I BIND FOR PLAN CBRHWDAS SUCCESSFUL
DSNT200I BIND FOR PLAN CBRHDUPD SUCCESSFUL
DSNT200I BIND FOR PLAN CBRHSBCC SUCCESSFUL
DSNT200I BIND FOR PLAN CBRHSPCC SUCCESSFUL
DSNT200I BIND FOR PLAN CBRHORCL SUCCESSFUL
DSNT200I BIND FOR PLAN CBRHCNTL SUCCESSFUL
```

Recommendation: The user ID associated with the CBRHBIND job must be the same as the user ID associated with the CBRISQL0 job, because the SQL statements in the OSMC application plans contain unqualified DB2 table names. When an unqualified DB2 table name is encountered during the BIND process, DB2 assumes the unqualified table name is the authorization ID of the binder.

Related reading:

- If you do not receive these messages, see “OAM Diagnostic Aids” in *z/OS DFSMSdfp Diagnosis* for more information.
- For additional information on binding a DB2 application plan, see IMS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPH2).

11b ***Run the CBRHGRNT SAMPLIB job.***

You must perform this step both for migration and at initial installation.

Granting authority

To grant authority to use the application plans, run SAMPLIB member CBRHGRNT. There are no changes other than user data that you add to the JCL. If you do not grant this authority, OSMC will not initialize, because it is not authorized to use the application plans.

12 LCS, ISMF, and OSR application plans

After creating the database for LCS, you must create and bind the LCS (CBROAM), ISMF (CBRISMF), and OSR (CBRIDBS) application plans. You must also grant authority for the plans to be used.

12a ***Run the CBRABIND SAMPLIB job.***

If you plan to create the OAM configuration database and to start the OAM address space to store objects, you must perform this step for both migration and at initial installation. Do **not** perform this step if you are using only DB2 sublevel storage (no file system, optical volumes, or tape devices), and do not start the OAM address space for processing objects.

Creating the LCS, ISMF, and OSR application plans

Member CBRABIND binds the packages created in CBRPBIND to the LCS, ISMF, and OSR application plans. Modify this sample job to include the storage group high level qualifiers for the installation (currently set up for GROUP00–GROUP99). This job also includes the addition of the CBRUTIL plan.

After you run the job, the following DB2 messages should appear in the job output:

```
DSNT200I BIND FOR PLAN CBROAM SUCCESSFUL
DSNT200I BIND FOR PLAN CBRISMF SUCCESSFUL
DSNT200I BIND FOR PLAN CBRIDBS SUCCESSFUL
```

Recommendation: The user ID associated with the CBRABIND job must be the same as the user ID associated with the CBRSAMPL job, because the SQL statements in the LCS and ISMF application plans contain unqualified DB2 table names. When an unqualified DB2 table name is encountered during the BIND process, DB2 assumes the unqualified table name is the authorization ID of the binder.

Related reading:

- If you do not receive these messages, see “OAM Diagnostic Aids” in the *z/OS DFSMSdfp Diagnosis* for more information.
- For additional information on binding a DB2 application plan, see IMS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPH2).

12b ***Run the CBRAGRNT SAMPLIB job.***

If you plan to create the OAM configuration database and to start the OAM address space to store objects and you have created the application plans using CBRABIND in step **12a**, you must perform this step for both migration and at initial installation. Do not perform this step if you are using only DB2 sublevel storage (no file system, optical volumes, or tape devices), and do not start the OAM address space for processing objects.

Granting authority

To grant authority for these application plans to be used, run member CBRAGRNT. There are no changes other than user data that you add to the JCL. If you do not grant this authority, the OAM address space will not start, because it is not authorized to use the CBROAM application plan, and ISMF and OSR will be unable to gain access to the OAM configuration database.

Recommendation: If you have applications containing application plans also using the modules identified in the CBRIBIND SAMPLIB job, you must rerun the BIND and GRANT jobs for these *other* applications you have installed as well. The IBM SAA ImagePlus Object Distribution Manager, the IBM Report Data Archive and Retrieval System (RDARS), and the Item Access Facility CICS application (IAFC), are examples of applications that bind OAM and Object Distribution Manager modules into a common plan.

13 Verifying DB2 installation

After creating all required databases and application plans, and after granting authorization for the entire OAM system (OSR, LCS, and OSMC), do the following verification process:

13a ***Verify that all application plans have been created.***

You must perform this step both for migration and at initial installation.

Perform the following verification step if your OAM installation uses optical or tape devices, or if you intend to start the OAM address space for object processing.

Use SPUFI to enter the following command:

```
        SELECT * FROM SYSIBM.SYSPLAN
          WHERE NAME = 'xxxxxxxx';
```

Substitute each of the following plan names for the xxxxxxxx on the WHERE clause:

```
CBROAM        CBRHSVOL      CBRHDUPD      CBRISMF
CBRHSMSI      CBRHSBKV      CBRHSBCC      CBRUTIL
CBRHOBJP      CBRHRDAS      CBRHSPCC      CBRHORCL
CBRHSOBP      CBRHWDAS      CBRIDBS       CBRHCNTL
```

You must perform this step if you are going to use only DB2 sublevel storage (no file system, optical volumes, or tape devices), and do not start the OAM address space for processing objects.

Use SPUFI to enter the following command:

```
        SELECT * FROM SYSIBM.SYSPLAN
          WHERE NAME = 'xxxxxxxx';
```

Substitute each of the following plan names for the xxxxxxxx on the WHERE clause:

```
CBRHSMSI
CBRIDBS
```

Execute the SELECT statement once for each plan. After each execution, one row of information should be returned. See the SYSIBM.SYSPLAN table in IMS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPH2) for a detailed description of the information that should be returned.

13b ***Verify that all application plans have been authorized.***

You must perform this step both for migration and at initial installation.

Perform the following verification step if your OAM installation uses optical or tape devices, or if you intend to start the OAM address space for object processing.

Use SPUFI to enter the following command:

```
        SELECT * FROM SYSIBM.SYSPLANAUTH
          WHERE NAME = 'xxxxxxxx';
```

Substitute each of the following plan names for the xxxxxxxx on the WHERE clause:

```
CBROAM        CBRHSVOL      CBRHDUPD      CBRISMF
CBRHSMSI      CBRHSBKV      CBRHSBCC      CBRUTIL
CBRHOBJP      CBRHRDAS      CBRHSPCC      CBRHORCL
CBRHSOBP      CBRHWDAS      CBRIDBS       CBRHCNTL
```

13c *Verify that all application plans have been created.*

You must perform this step if you are going to use only DB2 sublevel storage (no file system, optical volumes, or tape devices), and do **not** start the OAM address space for processing objects

Use SPUFI to enter the following command:

```
        SELECT * FROM SYSIBM.SYSPLANAUTH
          WHERE NAME = 'xxxxxxxx';
```

Substitute each of the following plan names for the *xxxxxxxx* on the WHERE clause:

```
CBRHSMSI
CBRIDBS
```

Execute the SELECT statement once for each plan. After each execution, one row of information should be returned.

If no information is returned, access authority to the application plan specified on the SELECT statement has not been successfully granted.

Related reading: For more information, see “OAM Diagnostic Aids” in the *z/OS DFSMSdfp Diagnosis* and the SYSIBM.SYSPLANAUTH table in IMS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPH2).

14 IPL the system

You must perform this step both for migration and at initial installation.

Use the new I/O configuration definition to IPL the system. The following messages are issued and can be used as verification that the IPL of the system is successful.

```
CBR8001I OAMx subsystem initialization starting.
CBR8002I OAMx subsystem initialization completed.
```

In a multiple OAM configuration, CBR8001I and CBR8002I will be issued for each OAM subsystem properly defined in the IEFSSN*xx* member of PARMLIB. In a classic OAM configuration, they will only be issued once.

Note: In a classic OAM configuration, if your installation is not using OAM for storing objects and is strictly using OAM for tape storage management, the following message can be ignored when it is displayed in response to IPLing the system:

```
CBR8007I No DB2 SSID or the DB2 SSID value of "NONE" has been specified.
OTIS subsystem is not started.
```

15 Specifying the SMS definitions and programs used by OAM

At this point, OAM is installed; however, you must complete several more steps before you can run applications:

- Define hardware devices, such as optical libraries and optical drives.
- Define OAM-related SMS constructs, such as storage class, management class, storage group, and data class. These constructs express your storage management policy.
- Develop ACS routines to assign constructs to objects. The ACS routines implement your storage management policy.

To complete these tasks, you must first translate your installation's business needs into technical terms. Once that has been accomplished, use ISMF to create construct definitions and programs in the system.

During installation, customization, and testing, it is reasonable to expect that you must adjust the configuration definitions through ISMF. You might also need to perform OAM operator tasks, such as entering optical disks or tape cartridges into associated libraries.

Related reading: For more information, see the following topics:

- Chapter 4, "Administering OAM," on page 183
- Chapter 5, "Operating OAM and OTIS address spaces and OSMC functions," on page 255

- *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*

Translating the business analysis into technical definitions

The SMS definitions are the mechanisms by which the results of your business analysis are implemented. The ideal groups, classes, and cycles identified during analysis must be translated into practical terms. This section discusses the OAM-related SMS parameters and how they can be used to customize OAM for your needs.

As in any translation process, compromises and approximations must be made. For example, you might have to make trade-offs between performance and cost (DASD versus optical versus tape storage) as you implement your storage management policy.

Be alert to the subtle factors that can influence system performance and, therefore, should be considered during this translation process. For example:

- The number of class transitions for a given object during its lifetime can have a significant effect on work load.
- The effective data transfer rate is faster for a single large object than for multiple small objects.

The choice you make for any single parameter has the potential to affect many other parameters. Be prepared to adjust those choices after installation, as you tune and refine your system.

Naming conventions

Naming conventions are an essential part of managing storage through SMS. For example, the ACS routines use names as part of the processing for evaluating class change requests.

Storage group parameters

The following storage group parameters are important when you define Object or Object Backup storage groups for objects stored on optical storage:

- DRIVE STARTUP THRESHOLD
- VOLUME FULL THRESHOLD
- MARK VOLUME FULL AT FIRST WRITE-FAILURE OPTION

The following storage group parameters also play an important role in controlling the storage management cycle for Object and Object Backup storage groups:

- CYCLE START TIME
- CYCLE END TIME
- OSMC PROCESSING SYSTEM NAME
- OAM RETENTION PROTECTION
- OAM DELETION PROTECTION

Related reading: For more information about these parameters, see the information that follows in this section and see “Defining Storage Groups” in *z/OS DFSMSdfp Storage Administration*.

DRIVE STARTUP THRESHOLD

This parameter indicates that OAM needs to start another optical drive to manage additional write requests for a specific Object or Object Backup storage group. When the number of requests to write objects to this Object or Object Backup storage group divided by the number of drives processing write requests for this storage group exceeds this threshold, OAM starts using an additional optical disk drive for writing to this storage group if one is available.

Note: Tape drive startup thresholds are not determined in the same manner as the thresholds in an optical environment. The drive startup threshold for tape drives is specified in megabytes, while the drive startup threshold for optical volumes is specified by the number of write requests. For information of how the tape drive startup thresholds are determined, see the discussion concerning TAPEDRIVESTARTUP on page “TAPEDRIVESTARTUP(threshold in megabytes)” on page 127.

While your planning is typically oriented toward the OSMC storage management cycle, other OSMC functions (for example, Volume Recovery, Move Volume utility, and others) cause objects to be written into specific Object or Object Backup storage groups as well. If you plan to use these other types of OSMC functions, you need to consider and account for the drives these other types of OSMC functions require in addition to the drives required by the OSMC storage management cycle when determining the appropriate value for the threshold.

Exceeding the DRIVE STARTUP THRESHOLD

Figure 16 on page 161 is an example of how OAM manages additional write requests for optical when the threshold is exceeded.

```
SCENARIO:

# OF WRITE REQ.   # OF DRIVES          DRIVE STARTUP THRESHOLD
FOR STRGPA        CURRENTLY RUNNING    FOR STRGPA

12                1                    17

During the storage management cycle, more write requests are created
for STRGPA that exceed the DRIVE STARTUP THRESHOLD set for this storage
group.  The following occurs:

# OF WRITE REQ.   # OF DRIVES          DRIVE STARTUP THRESHOLD
FOR STRGPA        CURRENTLY RUNNING    FOR STRGPA

30                2                    17

OAM starts another drive to handle the excess requests for this
storage group as it continues to process requests on the first drive.
When the number of write requests drops below the threshold,
the number of drives decreases.
```

Figure 16. Another Drive is Started When DRIVE STARTUP THRESHOLD is Exceeded

This parameter allows you to control the number of drives used concurrently for writing objects on optical volumes within a storage group. Setting the DRIVE STARTUP THRESHOLD to the maximum of 9999 with no more than one volume mounted for a storage group increases the likelihood that OAM will write the objects sequentially. The default value for DRIVE STARTUP THRESHOLD is 17.

Improving performance with low DRIVE STARTUP THRESHOLD value

Using a low DRIVE STARTUP THRESHOLD value can allow objects to write concurrently during the storage management cycle, resulting in improved performance and reduced cycle time for the storage group. If more than one drive is used for writing, then the writing of an object to a volume is interspersed with the writing of other objects to their respective volumes. Objects for only one storage group reside on the two optical volumes on an optical disk. Spreading data across volumes makes those volumes unavailable for other storage groups.

You can choose different DRIVE STARTUP THRESHOLD values for different storage groups, allowing you to choose independently between drive write concurrency and the volume-fill characteristic for each of the groups.

The determination of the DRIVE STARTUP THRESHOLD is the same for both Object and Object Backup storage groups.

Recommendation: Make certain that you have enough usable optical disks per storage group (scratch or already assigned to the storage group) to be used simultaneously for the write requests to the storage group once the DRIVE STARTUP THRESHOLD is crossed. OAM does not issue any message to request additional space for a storage group when the DRIVE STARTUP THRESHOLD is crossed and additional space for the storage group is not available. If the DRIVE STARTUP THRESHOLD is crossed, and there are no usable optical disks available, it is as if the threshold were never crossed; OAM continues to perform the write requests on the existing busy disks. Performance is slower than it would have been if additional space had been available when the threshold was crossed.

Table 24 on page 162 contains information to help you select a value for DRIVE STARTUP THRESHOLD, based on average object size and the number of drives on which concurrent activity is permitted. If your average object size is slightly larger or smaller than the size shown in the table, adjust the DRIVE STARTUP THRESHOLD value accordingly.

Table 24. Recommended values for DRIVE STARTUP THRESHOLD

Average object size in bytes	Activity on one drive only	Activity on two drives	Activity on three drives
3 000	>= 4680	2340–4679	1560–2339
4 000	>= 3600	1800–3599	1200–1799
5 000	>= 2880	1440–2879	960–1439
8 000	>= 1800	900–1799	600–899
16 000	>= 864	432–863	288–431
32 000	>= 432	216–431	144–215
40 000	>= 360	180–359	120–179
64 000	>= 216	108–215	72–107
100 000	>= 144	72–143	48–71
128 000	>= 108	54–107	36–53
256 000	>= 48	24–47	16–23
512 000	>= 24	12–23	8–11
1 000 000	>= 9	5–8	3–4
2 000 000	>= 5	3–4	2

VOLUME FULL THRESHOLD

OAM does not select full volumes to satisfy a write request. You might choose to eject full volumes from a library (for example, using ISMF to obtain a list of all full volumes in a library, then using the EJECT line operator) to streamline processing.

Thus, it is important to select a threshold value that allows the volumes to be marked full in a consistent manner. You must consider the size of the objects stored into the group. If the size of the objects is consistent, choose a threshold that is slightly larger than that size. The VOLUME FULL THRESHOLD is set on a storage group basis. When the number of available kilobytes on a volume falls below the VOLUME FULL THRESHOLD for the storage group to which that volume belongs, the volume is marked full and will not be used for any later write requests. If you find that volumes are not being selected for new objects and they are not being marked full, increase the value for this parameter.

Additionally, during OAM initialization, a volume previously set to FULL=Y with the MODIFY OAM,UPDATE command might be subsequently marked not full. This is because the volume's current tape-full percentage is less than the value of the TAPEPERCENTFULL parameter on the SETOAM statement. If you intend to mark the volumes in an Object or Object Backup storage group unavailable as candidates for write requests, you could mark them as WRITABLE=N because the writable status of the volume does not change during OAM initialization.

Note: The combination of the TAPEFULLTHRESHOLD parameter (either at the global or storage group level) and the TAPEPERCENTFULL parameter on the SETOAM statement determines the percent-full-utilization percentage and the tape-full-threshold factor for a tape volume at the storage group level. See “SETOAM keyword definitions for STORAGEGROUP subparameters” on page 121 for more information on the TAPEFULLTHRESHOLD and TAPEPERCENTFULL parameters.

MARK VOLUME FULL AT FIRST WRITE-FAILURE OPTION

As an alternative to, or in addition to VOLUME FULL THRESHOLD, consider using the MARK VOLUME FULL AT FIRST WRITE-FAILURE OPTION parameter. If the application using the storage group stores objects that have a wide variety of sizes, use this parameter so the volumes are marked full as determined by the ability of the volume to contain the object. If there is insufficient space on the volume for the object, choosing this parameter causes a volume to be marked full and an alternate volume to be chosen. For example, a request to write an object larger than 30 KB causes the demount of a volume with a full threshold of 20 KB if there is less than 30 KB free, space, in preference to a volume that contains sufficient space. If only the VOLUME FULL THRESHOLD parameter is used, the volume is not marked full; if the MARK VOLUME FULL AT FIRST WRITE-FAILURE OPTION parameter is used, the volume is marked full.

An additional side effect of not marking volumes full is the potential for additional volume mounts. A volume is selected according to its ability to contain the next object written with the intent of filling that volume. If there is a small amount of free space, the volume might be mounted to write one object. If the next object does not fit on that volume, it is demounted to mount another volume that is capable of containing the next object.

CYCLE START TIME and CYCLE END TIME

As described in “Understanding storage management cycles” on page 64, the storage management cycle ensures that every object scheduled for processing is placed in the correct level of the object storage hierarchy, is deleted, expired, or backed up, and, if necessary, is flagged for action during a later storage management cycle.

Note: CYCLE START TIME and CYCLE END TIME are based on the timezone specification TIME=xxx specified on the OAM subsystem definition within your IEFSSNxx parmlib member. This value may be set to GMT or if the TIME= is not specified, set to local time.

There are five methods by which management cycles can be controlled:

- **Manual Start (All Groups)**

  You can start the storage management cycle manually for all storage groups by using the MODIFY OAM, START, OSMC operator command. START OSMC starts all groups that either have the system name or no system name specified as the OSMC processing system. (See "Starting OSMC functions" on page 263 for command syntax.)

- **Manual Start (Individual Group)**

  You can start the storage management cycle manually for an individual storage group by using the MODIFY OAM,START,STORGRP operator command. Using this approach, you can directly control the processing sequence, such as using this technique to give priority to a storage group with many objects.

- **Manual Stop**

  You can stop the storage management cycle manually by using the MODIFY OAM,STOP,OSMC command. (See "Stopping OSMC" on page 379 for command syntax.)

- **Manual Stop (Individual Group)**

  You can stop the storage management cycle manually for an individual Object or Object Backup storage group by using the MODIFY OAM,STOP,STORGRP operator command.

- **Cycle Start Window**

  You can start the storage management cycle automatically for Object and Object Backup storage groups by specifying CYCLE START TIME and CYCLE END TIME parameters in the storage group definitions. If using the automatic startup in an OAMplex, you should specify an OSMC processing system name to avoid multiple starts for the same storage group on different systems. If you start OSMC during the window delimited by those times, the storage management cycle is started for that storage group. This is the usual method for controlling storage management cycles. You can also indicate that no automatic processing for the storage group is to be performed by specifying NONE for the CYCLE START TIME and leaving the CYCLE END TIME blank.

  The SETOSMC CYCLEWINDOW keyword interacts with the CYCLE START TIME and CYCLE END TIME parameters in the following ways:

  – If SETOSMC CYCLEWINDOW is not specified, STARTONLY is the default mode for the cycle start window.
  – If you specify the SETOSMC CYCLEWINDOW(STARTONLY) keyword in the CBROAM*xx* member, CYCLE START TIME and CYCLE END TIME describe a window during which the storage management cycle might start. These cycle times do not define the length of the processing period. The storage management cycle might continue to run after the specified CYCLE END TIME has passed. Consider adjusting the start time or times if the processing for one group extends into the start time for another; otherwise, resource contention can become severe enough to affect the total time that it takes to perform storage management cycle processing for all groups.
  – If you specify the SETOSMC CYCLEWINDOW(STARTSTOP) keyword in the CBROAM*xx* member, CYCLE START TIME and CYCLE END TIME describe a window during which the storage management cycle starts and ends. The storage management cycle ends at the specified stop time. Although OSMC stops processing new work for this storage group, this group can continue to finish what work it has started.

  Related reading: For more information on the SETOSMC CYCLEWINDOW keyword, see "SETOSMC keyword definitions" on page 140.

All Object or Object Backup storage group definitions must define a window where the storage management cycle starts for the storage group, or indicate that no automatic processing be performed for the storage group. Consider the following issues as you select window start and end times for each storage group:

- Number of optical drives
- Number of tape drives
- Number and size of objects moving through the hierarchy
- Backup requirements
- Time required to process the group

- Impact on end users who might be doing retrievals
- Storage management cycles for other groups
- Application usage patterns
- General maintenance operation requirements

Processing during the storage management cycle for a group does not require use of an optical or tape drive under the following conditions:

- The storage group does not specify a library.
- There is no class transition that requires moving an object to optical or tape storage.
- No objects require backup.

Storage management cycle processing requires at least one drive if any objects are moved or backed up to optical or tape storage. (See "DRIVE STARTUP THRESHOLD" on page 161 and *TAPEDRIVESTARTUP* on page "TAPEDRIVESTARTUP(threshold in megabytes)" on page 127 for other considerations.) If a storage management cycle is in process on more than one Object or Object Backup storage group at a time and the number of groups exceeds drive availability, frequent volume mounts occur in an attempt to satisfy the requests to write objects to optical and or tape volumes. For example, when objects are written to a mounted volume for one group, that volume must be demounted to allow the mounting of another volume to accept the objects for a different group. Unless you limit resource consumption during the storage management cycle by some other means, you must not specify overlapping start windows for more groups than you have drives. (See "OAM cataloged procedure parameter (MAXS)" on page 166.)

If an object requires more than one backup copy, the first and second backup copies are written to separate Object Backup storage groups. Objects are written to backup volumes in the Object Backup storage group specified in the SETOSMC statement for the Object storage group to which they belong, or, if a SETOSMC statement is not specified, the backup copies are written to backup volumes in the default Object Backup storage group specified.

More than one Object storage group can use the same Object Backup storage group for the same backup copy (first or second); therefore, first backup copies of objects from one group can reside on the same volume as first backup copies from other groups and second backup copies of objects from one group can reside on the same volume as second backup copies from other groups. These objects are written during the storage management cycle for the group containing the object. If some groups require object backup, but some do not, consider processing groups that require object backup concurrently with groups that do not require object backup.

Recommendation: Process the storage management cycle for an Object storage group while other activity for the objects in the Object storage group is light. For example, specify a cycle start window during a period when applications are not accessing data heavily. You must consider the effect of concurrent object use in a group during the storage management cycle for that group. DB2 performs deadlock detection on tables (directory tables in particular) that are shared by tasks performing the storage management cycle processing and by user tasks requesting OAM functions through the application interface. The potential for DB2 deadlocks is much greater if an application is accessing data in a group during the storage management cycle for that group.

OSMC processing system name

The OSMC processing system name specifies the system where the OSMC processing for an Object or Object Backup storage group is to be performed. This name is used within an OAMplex where multiple instances of OAM are running in a Parallel Sysplex sharing a common configuration and DB2 database. When the OSMC cycle window occurs, the system specified in the OSMC processing system name of the storage group is where the storage group automatically starts. If the OSMC processing system name is left blank, the storage group cycle runs on all systems in an OAMplex during the cycle window.

If you use the MODIFY OAM,START,OSMC command to start a full OSMC cycle, the storage groups with an OSMC processing system name that matched the system where the command was entered are started. Also, storage groups with no specified OSMC processing system name are started.

If you use the MODIFY OAM,START,STORGRP command, the storage group specified is started on the system when the command was entered. Any specification in the OSMC processing system name for that storage group is ignored.

Recommendation: The system name that you specify for a storage group should be the same system where the hardware associated with the object storage group is connected. If the Object storage group has libraries associated with it that are owned by different OAMs in the OAMplex, then determine if there is a higher volume of data localized to one instance of OAM and use that system as the OSMC processing system name. You might need to modify and tune this information to meet performance objectives for your installation.

OAM deletion protection

The OAM Deletion Protection attribute in the SMS object storage group definition works in combination with the DP=P keyword in IEFSSNxx PARMLIB member to determine the current deletion-protection mode (enabled or disabled) for all objects in this object storage group. The OAM Deletion Protection attribute is ignored when DP=A or DP=N. When deletion-protection is enabled, objects in this object storage group cannot be deleted prior to their expiration date. Deletion-protection does not restrict any changes to an object's expiration date.

Note: If both deletion-protection and retention-protection are enabled then retention-protection takes precedence.

To enable deletion-protection for a given object storage group, use ISMF to set the OAM Deletion Protection attribute to Enabled in the SMS object storage group definition, and then activate the SCDS. For more information on OAM Deletion Protection see “Protecting against inadvertent object deletion” on page 11.

OAM retention protection

New objects stored into an object storage group with this parameter enabled are flagged as retention-protected for the entire life of the objects. A retention-protected object cannot be deleted prior to its expiration date and its expiration date can never move to an earlier date. Enabling retention-protection could result in objects that will never expire and therefore cannot be deleted.

Note: If both deletion-protection and retention-protection are enabled then retention-protection takes precedence.

To enable retention-protection for a given object storage group, use ISMF to set the OAM Retention Protection attribute to Enabled in the SMS object storage group definition, and then activate the SCDS. Any objects stored into an object storage group after the OAM Retention Protection attribute has been enabled will be stored as retention-protected objects. Retention-protected objects can be identified as such by the ODSTATF_RETPROT flag in the object directory.

For more information on OAM Retention Protection see “Protecting against inadvertent object deletion” on page 11.

OAM cataloged procedure parameter (MAXS)

The MAXS parameter in the OAM cataloged procedure specifies the maximum number of Object or Object Backup storage group tasks that can be processed concurrently during an OSMC storage management cycle. If storage management cycles for groups overlap, you can use the MAXS parameter to limit resource consumption. When determining the value for MAXS, take into consideration the number of optical and tape drives that are available for storage management processing to avoid drive contention. If MAXS is not specified, a default of 2 is assigned. (See the discussion of MAXS parameter in “5h Updating the PROCLIB” on page 146.)

If you plan to use these other types of OSMC functions (such as MOVEVOL, RECOVERY or RECYCLE commands), you need to consider and account for the resources these other types of OSMC functions require first, and then distribute the remaining resources for the OSMC storage management cycle with MAXS.

Optical and tape device availability are the resources most likely to cause contention. Writing active data to optical media during the storage management cycle is done on an Object storage group boundary. No active data is mixed between groups; therefore, each group being processed requires a different optical volume. Concurrent requests for different volumes are likely to result in concurrent requests for optical drives. For example, if you have a single, four-drive library and there are concurrent storage management cycles for four storage groups requiring the writing of data to optical media, all four drives are used. If there are concurrent requests to retrieve data from optical volumes during the cycle, those requests and the processing performed during the storage management cycle contend for resources and detract from the performance of each of the functions. Consider using the MAXS parameter with DRIVE STARTUP THRESHOLD to limit resource consumption for writing objects to optical media during storage management cycles, thereby leaving resources available to mount volumes for retrieval requests.

For object tape volumes, the limits set in the parameters and subparameters of the SETOAM statement for TAPEDRIVESTARTUP, MAXTAPESTORETASKS, and MAXTAPERETRIEVETASKS limit resource contention regarding tape library dataservers. These parameters can work with the MAXS parameter resulting in effective resource utilization of the entire storage management environment. For more information on these and other parameters associated with the SETOAM statement, see "SETOAM statements for object tape storage" on page 109.)

The default value for MAXS is two. This default was chosen as a reasonable value when the configuration includes one library with four optical drives and two stand-alone drives. This default allows for concurrent storage management processing for two storage groups and it also allows for overlapping the writing of backup copies to stand-alone drives with the writing of primary optical copies to the library drives. Also, it leaves spare library drives available for retrieve requests and as alternate drives in case of a drive failure.

MAXS could possibly be increased to four when a second library is added to the configuration. Do *not* set MAXS to a value larger than the number of optical or tape drives that are available for storage management processing. Before increasing the value of MAXS, you should verify that there is sufficient processing capacity available to manage the increased work load, because processing requirements are heavier for small objects than for large ones. Also, you need to consider the tuning guidance described in "Tuning OAM" on page 201.

In addition, it is necessary to assign Object and Object Backup storage groups across the libraries in such a way that the library work loads are balanced. This storage assignment prevents any library from becoming a bottleneck.

Storage class and management class parameters

The OAM-related parameters for defining storage and management classes are described in "Defining storage classes" on page 174 and "Defining management classes" on page 175. The parameters are not inherently difficult to understand; however, implementing them effectively can be challenging.

During the translation process, establish parameter values for each class; then, evaluate the entire set of classes to ensure that the performance, retention, backup, and processing cycles they define correspond to the requirements established during the business analysis phase.

ACS routine input variables

Automatic class selection routines are used to implement your installation's storage management policy. These routines must be written using the ACS programming language, a high-level language that uses relational statements to determine class and storage group assignments.

Related reading: The *z/OS DFSMSdfp Storage Administration* contains detailed information about the ACS programming language and the use of ISMF to define and to validate ACS routines.

The ACS routines use three values for the &ACSENVIR variable that is specific to objects. See Table 25 on page 168 and Table 26 on page 169 for a diagram of these variables. Using these values, you can distinguish object selection from data set selection. These values are as follows:

&ACSENVIR='STORE'

The storage class, management class, and perhaps the storage group routines are invoked because of an application's request to store an object. Variable &DSN contains the collection name.

If variable &MEMN (*object name*) is null, the ACS routines are invoked to specify a storage class, management class, storage group for the collection named in &DSN. Therefore, you must supply ACS routines that select a storage class, management class, and Object storage group for the collection. The storage class and management class that is supplied by the ACS routines become the default classes for the collection. The storage group that is selected indicates in which Object storage group the collection is to be a member. All objects in the collection are stored in the storage group that you select.

If variable &MEMN is not null, this ACS invocation validates the storage class and management class that is specified by the application for the object that is named in variable &MEMN. ACS routines ensure that the stated class is acceptable for use with this object, and if not, should substitute an acceptable one. This does not affect the class specifications for the collection.

When neither storage class nor the management class is specified on a request to store an object into an existing collection, the object is assigned the default classes that are associated with that collection.

Attention: When an object is assigned the default classes that are associated with the collection, the ACS routines are *not* invoked, and it is possible to store an object with a name that does not conform to the requirements in the ACS CHANGE or CTRANS environments. *Subsequent attempts to process that object will fail*.

&ACSENVIR='CHANGE'

ACS is invoked to validate an application’s request to change the storage class or management class for the object that is named in variable &MEMN that is part of the collection that is named in variable &DSN. ACS routines should ensure that the stated class is acceptable for use. The appropriate ACS routine is invoked based on the combination of storage class (SC) and management class (MC) specifications included on the application request:

- If only MC is specified, the management class ACS routine is invoked, using the requested MC and the existing SC.
- If only SC is specified, both the storage class and management class ACS routines are invoked, using the requested SC and the existing MC.
- If MC and SC are specified, management class and storage class ACS routines are invoked, using the requested MC and SC.

&ACSENVIR='CTRANS'

During a storage management cycle, ACS is invoked because a class transition event has occurred for the object that is named in variable &MEMN that is part of the collection that is named in variable &DSN. Variables &MGMTCLAS, and &STORCLAS have the names of the classes to which the object is assigned. ACS routines should select the new classes. These new classes can change the placement of the object in the hierarchy and can change the management of the object (including creation of a new transition event).

Table 25. Constructs verified or changed through ACS Routines Invoked by the &ACSENVIR variables—All OAM environments

ACS routines invoked (&ACSENVIR)	STORE	CHANGE	CTRANS	*ALLOC
DATA CLASS	NO	NO	NO	YES
STORAGE CLASS	**YES	***YES	YES	YES
MANAGEMENT CLASS	**YES	***YES	YES	NO
STORAGE GROUP	**YES	NO	NO	YES

Table 25. Constructs verified or changed through ACS Routines Invoked by the &ACSENVIR variables—All OAM environments (continued)

ACS routines invoked (&ACSENVIR)	STORE	CHANGE	CTRANS	*ALLOC

Notes:

- *ACS environment of ALLOC is invoked by MVS during allocation.
- **When storing the first object into a new *collection* through the STORE request, the ACS routines for storage class, management class, and perhaps the storage group are entered.
- ***The ACS routines for management class and storage class are entered for *objects* only when an explicit management class, storage class, or both are specified on a CHANGE request. (See the ***CHANGE Environment Only*** graphic for more information.)

Table 26. Constructs verified or changed through acs routines invoked by the &ACSENVIR variables—Change environment only

ACS routines invoked (OSREQ specifies)	Storage class only	Management class only	Storage class and management class
Storage class changed	YES	NO	YES
Management class changed	NO	YES	YES

In addition to &ACSENVIR, ACS routines might also use the following:

- Read-Write Variables

  &MGMTCLAS
  &STORGRP
  &STORCLAS
  &SYSNAME
  &SYSPLEX

- Read-Only Variables

  &DB2SSID
  &DSN
  &HLQ
  &LLQ
  &MEMHLQ
  &MEMLLQ
  &MEMN
  &MEMNQUAL
  &NQUAL
  &RETPD
  &SIZE

Tip: The &DB2SSID variable is not available on releases prior to V2R3. Attempting to reference &DB2SSID in an ACS routine running on an earlier release will result in an error. When coding ACS routines to be shared across multiple systems, logic must be included that avoids referencing &DB2SSID on any system running at a level prior to V2R3. Variables &SYSNAME and &USER_ACSVAR (available beginning in V2R2) might be of use in doing this.

Tip: &SIZE is valid **only** for STORE requests. &SIZE contains the object size that is converted to kilobytes (KB) and rounded up to the next highest 1 KB if the object size is not at a KB boundary (KB=1 024 bytes).

Related reading: For detailed information about ACS variables, see the *z/OS DFSMSdfp Storage Administration*.

Storing objects in a collection

When the first object is stored in a collection with an OSREQ STORE, the Object storage group for that collection is derived by the SMS Storage Group ACS routine and the collection name entry in the DB2 Collection Name Table is defined with the default management class and storage class. After an object is stored, the collection name entry in the DB2 Collection Name Table helps to locate the object. (See Figure 8 on page 34 for a diagram of the process of storing an object.) The DB2 Collection Name Table entry for the collection name contains a directory token that determines which DB2 database contains the object directory entry for the object. If a collection name entry from the DB2 Collection Name Table is lost, objects in that collection will not be processed by the storage management cycle.

Creating OAM definitions with ISMF

After the translation process has been completed, it is necessary to define all the OAM elements to the system. ISMF provides a series of panels through which SMS parameters can be defined.

Grant DB2 authority to the OAM configuration databases to the ISMF user who will be creating the optical constructs.

Related reading:

- You might also want to read Appendix A, "Sample optical hardware configurations," on page 385 before you begin using ISMF to familiarize yourself with hardware configuration issues and the CTC / 3995 device numbers needed for the library.
- For detailed information about using ISMF, see *z/OS DFSMS Using the Interactive Storage Management Facility*.
- For ISMF information about tape volumes and libraries, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Defining an SCDS

15a ***Define the base SCDS.***

You must perform this step at initial installation. During migration, you might optionally perform this step if you are creating additional SCDSs.

The procedure for defining a source control data set is provided in the *z/OS DFSMSdfp Storage Administration*. It is possible to define several source control data sets describing different configurations; however, only one SCDS can be activated at any time.

15b ***Define libraries and drives in the OAM configuration database.***

You must perform this step at initial installation if you are using optical storage in your hierarchy. During migration, you might optionally perform this step if you are adding or changing libraries or drives.

Defining optical libraries and optical drives in the OAM configuration database results in data being entered in the DB2 tables. It is therefore necessary that the TSO ID of the user entering the ISMF definitions has access to DB2.

Related reading:

- For a more comprehensive discussion of defining optical libraries and optical disk drives or for other topics such as deleting, altering, or copying optical libraries or drives, see Appendix A, "Sample optical hardware configurations," on page 385.
- For ISMF information regarding tape libraries, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Defining storage groups and relating the libraries to the storage groups

15c ***Define Object and Object Backup storage groups.***

You must perform this step at initial installation. During migration, you might perform this step if you are adding or changing storage groups, or adding libraries to your configuration.

Use the Storage Group Application Selection panel to specify one of the following storage group types:

STORAGE GROUP TYPE
DESCRIPTION

OBJECT
Contains primary objects.

OBJECT BACKUP
Contains backup copies of objects.

Use the Object Storage Group Define panel to specify the following:

DESCRIPTION
Provide a free-form description of this storage group (up to 120 characters).

QUALIFIER
For Object storage groups, specify the name of the DB2 object storage database to be used for the object directory and the DASD high level qualifier of the storage group. This qualifier must also be defined as a package in the CBRPBIND job to create the package. This package must also be in the CBRHBIND and CBRABIND or CBRIBIND jobs to bind the package to the OSMC, OAM, and OSR plans.

CYCLE START TIME
For Object and Object Backup storage groups, specify the beginning of a window of time when OSMC can begin daily processing for this storage group. You must specify a value of 0 - 23, or NONE for all Object storage group definitions. A value of 0 - 23 represents an hour of the day. Specify an hour of the day as 00 for midnight, 01 for 1 AM., 23 for 11 PM., and so on. The hour of the day value for CYCLE START TIME must be different from the hour of the day value for CYCLE END TIME. A value of NONE indicates that no automatic processing for the storage group be performed. When NONE is specified, the CYCLE END TIME value must be blank. See "CYCLE START TIME and CYCLE END TIME" on page 163 for more information about this parameter.

Note: CYCLE START TIME and CYCLE END TIME are based on the timezone specification TIME=xxx specified on the OAM subsystem definition within your IEFSSNxx parmlib member. This value may be set to GMT or if the TIME= is not specified, set to local time.

CYCLE END TIME
For Object and Object Backup storage groups, specify the end of a window of time when OSMC can either begin or end daily processing for this storage group, depending on the mode defined in the CYCLEWINDOW keyword in the SETOSMC statement in the CBROAM*xx* PARMLIB member. You must specify a value 0 - 23, or blank (depending on your specification for CYCLE START TIME) for all Object storage group definitions. A value of 0 - 23 represents an hour of the day and is required when a value 0 - 23 was specified for CYCLE START TIME. Specify an hour of the day as 00 for midnight, 01 for 1 AM., 23 for 11 PM., and so on. The hour of the day value for CYCLE END TIME must be different from the hour of the day value for CYCLE START TIME. A value of blank is required when NONE has been specified for CYCLE START TIME. See "CYCLE START TIME and CYCLE END TIME" on page 163 for more information about this parameter.

LIBRARY NAMES
Specify the names of the optical disk libraries in your configuration (either one to eight pseudo or real optical libraries) that can contain volumes belonging to this Object or Object Backup storage group to which objects are written.

VOLUME FULL THRESHOLD
For Object or Object Backup storage groups, specify the number of free kilobytes to be used as a threshold for optical volumes that belong to this storage group. When the number of free kilobytes falls below this threshold, the volume is marked full and no more objects are placed on the volume. See "VOLUME FULL THRESHOLD" on page 163 for more information about this parameter.

If the optical volume table of contents is full, the volume is marked full regardless of what is specified in this parameter.

DRIVE STARTUP THRESHOLD
For Object or Object Backup storage groups, specify the point at which OAM starts using an additional optical disk drive for writing. When the number of requests to write objects to this Object or Object Backup storage group that is divided by the number of optical drives processing write requests for this storage group exceeds this threshold, OAM starts using an additional optical disk drive for writing. The default value for DRIVE STARTUP THRESHOLD is 17.

See "DRIVE STARTUP THRESHOLD" on page 161 for more information about this parameter, including a table of recommended values, based on average object size.

OSMC PROCESSING SYSTEM NAME
For Object and Object Backup storage groups, specify on which system OSMC processing is to run when this storage group is automatically started during the OSMC cycle window specified in the storage group definition. Specifying an OSMC processing system name avoids contention by preventing the storage group from being started automatically on multiple systems concurrently within an OAMplex.

MARK VOLUME FULL AT FIRST WRITE-FAILURE OPTION
For Object or Object Backup storage groups, specify whether the volume should be marked full at first write-failure.

If you specify Y (YES), OAM marks full an optical or tape volume in this Object or Object Backup storage group the first time an object cannot be written on this volume because there is not enough space remaining on the volume.

If you specify N (NO), OAM marks an optical volume in this Object or Object Backup storage group full only when the number of available kilobytes in the user data area falls below the VOLUME FULL THRESHOLD.

If the optical volume table of contents is full, the volume is marked full regardless of what is specified in this parameter.

If you specify N (NO), OAM marks full a tape volume in this Object or Object Backup storage group only when the number of available kilobytes falls below the TAPEFULLTHRESHOLD that is specified on the SETOAM statement in CBROAM*xx* member of PARMLIB.

See "MARK VOLUME FULL AT FIRST WRITE-FAILURE OPTION" on page 163 for more information about this parameter.

OAM DELETION PROTECTION
This parameter in concert with the DP=P keyword in IEFSSNxx PARMLIB member, determines the current deletion-protection mode (enabled or disabled) for all objects in this object storage group. This value is ignored when DP=A or DP=N. When deletion-protection is enabled, objects in this object storage group cannot be deleted before their expiration date. Deletion-protection does not restrict any changes to an object's expiration date.

Note: If both deletion-protection and retention-protection are enabled, then retention-protection takes precedence.

Y
Enable.

N
Disable. This is the default.

OAM DB2 SSID
This parameter is used only for Object and Object Backup storage groups in a multiple OAM configuration. In all other cases, any value specified is ignored. The value indicates the SSIDs or Group Attachment Names of the DB2 subsystems (and related OAM instances) that use this storage group name. A specific SSID or Group Attachment Name can be specified or a wildcard character (%) can be used in one or more positions to indicate that multiple DB2 subsystems can use the storage group name. For example, a value of DB%A indicates that all DB2 subsystems with an SSID or Group Attachment Name that begins with DB and ends with A with any character in the third position can use

the storage group name. The default value %%%% indicates that all DB2 subsystems (and therefore all OAM instances) can use the storage group name.

Note: The relationship between a DB2 subsystem and an OAM instance is defined using the D= keyword in the INITPARM string on the OAM subsystem definition, either in the IEFSSNxx member of SYS1.PARMLIB or on the SETSSI ADD command.

OAM RETENTION PROTECTION

New objects that are stored into an object storage group with this parameter enabled are flagged as retention-protected for the entire life of the objects. A retention-protected object cannot be deleted before its expiration date, and its expiration date can never move to an earlier date. Enabling retention-protect might result in objects that will never expire and therefore cannot be deleted.

Note: If both deletion-protection and retention-protection are enabled, then retention-protection takes precedence.

Y

Enable.

N

Disable. This is the default.

SMS STORAGE GROUP STATUS

For Object or Object Backup storage groups, specify Y (YES) for this parameter. This specification displays another panel where you can specify one of the following values on the SMS Storage Group Status panel for the system running OAM if it is not in an OAMplex, or for the systems in an OAMplex:

ENABLE

Applications can store and retrieve objects in the group; OSMC can process the group.

DISNEW

Applications can retrieve objects from the group, but cannot store objects into the group; OSMC can process the group.

DISALL

Applications can neither store nor retrieve objects in the group; OSMC can process the group.

NOTCON

Neither applications nor OSMC can process objects in the group.

Attention: In an OAMplex environment, it is normal to have an object or object backup storage group that is defined as enabled to multiple systems. In fact, they are normally enabled to all the systems in the OAMplex. This is because the OAM members in the OAMplex share OAM tables in DB2 and it is desirable for all members of the OAMplex to have equal access to the same data.

However, in a non-OAMplex environment, instances of OAM on individual LPARs are independent of each other. They do NOT share common DB2 tables and therefore the object data accessible to OAM on one LPAR is not accessible to OAM on another LPAR. Therefore, the default behavior is to not allow a given object or object backup storage group to be defined as enabled to more than one system when OAM is initialized in a stand-alone (non-OAMplex) mode.

In a non-OAMplex environment, the SMS Storage Group Status panel is used to allow status to be specified for each system that is defined to SMS. By default, you must specify an option other than NOTCON for the one system that will be running OAM and you must specify the NOTCON option for all other systems. If you specify more than one system as other than NOTCON for an object or object backup storage group, a CBR0162I message is issued during OAM initialization and the storage group is ignored by OAM.

Some installations want to have the same storage group name that is enabled on multiple systems even though not running in an OAMplex. These installations, may have a given storage group name (GROUP22 for example) which contains *ABC-payroll object data* in the DB2 tables accessible to OAM running on LPAR #1, and contains *XYZ-insurance policy object data* in the DB2 tables accessible to OAM running on LPAR #2.

The SETOPT MULTISYSENABLE keyword available in the CBROAMxx PARMLIB member provides the customer the option to define a given object or object backup storage group as enabled to more than one system in a non-OAMplex environment. In a non-OAMplex environment, when MULTISYSENABLE(YES) is specified, then OAM will issue a CBR0165I informational message during OAM initialization if one or more object or object storage groups are defined as enabled to multiple systems in the active SMS configuration.

Refer to "SETOPT keyword definitions" on page 132 for more information on the MULTISYSENABLE keyword.

Also, storage group enablement status should not be confused with library connectivity. Changing the connectivity of a library does not automatically change the enablement status of any associated storage group. Storage group definitions might need to be updated to provide the correct storage group enablement status should the library connectivity be changed.

Defining storage groups to direct data to specific optical media types

With the IBM 3995 multifunction optical disk drives, a customer can choose to populate a single library with both WORM and rewritable optical media. This configuration allows the customer to direct data for a particular application to the WORM media while directing data for another application to the rewritable media on the same optical device.

To achieve this control over which media type is used for a specific application's data (objects) residing in the same optical library dataserver as another application's data, it is recommended that the following steps be completed:

- Define one or more Object storage groups for the application that wishes to have its objects stored on WORM media and one or more Object storage groups for the application that wishes to have its objects stored on rewritable media. Also include the library name associated with the library model in the Object storage group definition for each application.
- Enter the media into the 3995 optical library dataserver. The optical disk volumes residing on the WORM or rewritable media should be associated with one of the Object storage groups set up for the application that requires its data to be stored on the specific media type.
- Update the SMS storage group ACS routine to insure that the data belonging to the individual applications is assigned to the appropriate Object storage groups that only have WORM or rewritable optical disk volumes associated with them.
- Make certain there are sufficient optical disk volumes with available space and of the appropriate media type assigned to each Object storage group at all times.
- Make certain there are NO OAM scratch optical disk volumes in the multifunction 3995 Optical Library Dataserver. An OAM scratch optical disk volume can be assigned by OAM to any Object storage group when the Object storage group encounters an 'out-of-space' condition. All optical disk volumes should be preassigned to a given Object storage group based on the media type of the optical disk volume.

Another way for you to direct data belonging to different applications to different types of media is:

- Having two or more multifunction 3995 Optical Library Dataservers and populate one with WORM media and the other with rewritable media.
- Defining one or more Object storage groups for the application or applications for which you want data stored on either WORM or rewritable media, and including the library name associated with the multifunction library in the Object storage group definition.
- Using the Default Media Type option to restrict the media type that can be entered into a specific optical library dataserver. See the description of the DEFAULT MEDIA TYPE parameter in "Defining real 3995 libraries" on page 398 for more information on the Default Media Type option.

Defining storage classes

15d ***Define storage classes.***

You must perform this step at initial installation. During migration, you must perform this step if you are adding or modifying storage classes for object tape storage or file system sublevels.

OAM interprets the parameters used to define the storage class in an attempt to apply the stated performance objective. The following parameters are used by OAM as an indication of the performance objective for the object:

INITIAL ACCESS RESPONSE SECONDS (IARS)
Specify a performance objective relative to the elapsed time (in seconds) that can be tolerated before the first byte of data is made available for an application's request to retrieve an object. Use from 1 to 4 characters to specify a valid value of 0 to 9999. A value of 0 causes the object to be written to disk storage, and a value of greater than 0 causes the object to be written to removable media storage. Any OAM request that tries to use a storage class with a blank value for this parameter fails.

SUSTAINED DATA RATE (SDR)
A subparameter of the storage class parameter that specifies which removable media, optical or tape, is used to accept the primary copy of the object, once the Initial Access Response Seconds parameter determines that the object should be written to removable media. If the SDR is greater than or equal to three, the primary copy of the object is stored on a tape volume. If the SDR for the object is less than three, the primary copy of the object is stored on an optical disk volume.

OAM SUBLEVEL (OSL)
A subparameter of the storage class parameter to indicate what sublevel the storage class is associated with. The valid values are 1 or 2; the default value is 1.

OAM attempts to meet the performance objective by placing the object at a level in the storage hierarchy that comes closest to the objective. Avoid using performance objectives that force objects to be written directly to optical storage. Writing objects directly to optical media without staging them through the DB2 sublevel can degrade system performance and significantly increase the number of optical disks needed per day, due to inefficient optical VTOC directory space utilization. See Table 14 on page 79 for detailed information about the effects of writing objects directly to optical media.

AVAILABILITY
Specify a value for this parameter (STANDARD or CONTINUOUS), even though it is ignored for objects.

Defining data classes

15e ***Define data classes.***

You must perform this step if you are using object to tape storage in your environment and want to direct your tape related Object writes to an ATLDS or an MTL.

You need to specify a DATACLASS to be used to direct work requests to an ATLDS or an MTL. This DATACLASS must be added to the CBROAM*xx* PARMLIB member for each storage group that is directing Object tape writes to the ATLDS or MTL.

Related reading:

- See "Understanding the data class construct" on page 32 for more details concerning data classes.
- See *z/OS DFSMSdfp Storage Administration* for details on how to define your data classes for your installation.

Defining management classes

15f ***Define management classes.***

You must perform this step at initial installation. During migration, you might optionally perform this step if you are adding management classes.

Pages 1 through 3 of the Management Class Define panel are primarily for DFSMShsm's management of data sets. OAM uses the following subset of those parameters to manage objects in the same manner that DFSMShsm manages data sets.

Related reading: See "Defining Management Classes" in *z/OS DFSMSdfp Storage Administration* more information.

EXPIRATION ATTRIBUTES
Specify when an object can be deleted automatically by OAM, assuming deletion-hold is not enabled for the object and assuming the deletion is approved by the auto-delete installation exit.

Note: A value of NOLIMIT for the EXPIRATION ATTRIBUTES means that OAM will not automatically delete the objects that are associated with this management class. Those objects must be explicitly deleted by the application, using the application interface to OAM.

Related reading:

- See "Auto-delete installation exit (CBRHADUX)" on page 573 and *z/OS DFSMS Installation Exits* for information about the installation exit.
- See *z/OS DFSMSdfp Storage Administration* for a detailed discussion of the relationships between the EXPIRATION ATTRIBUTES and the RETENTION LIMIT values.

Note: If an object is in event-based-retention mode (ODEXPDT = '0002–02–02'), then object expiration depends on receipt of an external event trigger by the EVENTEXP keyword on the OSREQ API, and therefore the following management class parameters are ignored.

EXPIRE AFTER DAYS NONUSAGE
Specify when the object is to be automatically deleted by OAM relative to the elapsed time since it was last referenced. (If the object has not been referenced since it was stored, the create date is treated as the last reference.)

Note: Do not use UPD=N on the OAM*n* statement in the IEFSSN*xx* member of PARMLIB if this option is used in your management class.

EXPIRE AFTER DATE/DAYS
Specify when the object is to be automatically deleted by OAM, relative to the elapsed time since it was created or on an explicit date.

RETENTION LIMIT
Specify the retention limit that is allowed on explicit parameters on the application interface to OAM.

AUTO BACKUP
Specify whether you want to back up the object by writing 1 or 2 copies of the object. The backup copies are made during the OSMC storage management cycle. If you set AUTO BACKUP=Y and the number of backup copies that are specified in NUMBER OF BACKUP VERSIONS (DATA SET EXISTS) is 0 or 1, and you have not specified a SECONDBACKUPGROUP keyword on any SETOSMC statement in the CBROAM*xx* member of PARMLIB, then OSMC schedules a single backup copy of the object to be written. If you set AUTO BACKUP=Y and the number of backup copies that is specified in NUMBER OF BACKUP VERSIONS (DATA SET EXISTS) is 2 or above, and you have specified a SECONDBACKUPGROUP keyword on any SETOSMC statement in the CBROAM*xx* member of PARMLIB, then OSMC schedules two backup copies of the object to be written.

BACKUP FREQUENCY
Specify when you want the first backup copy to the object be written. If you set AUTO BACKUP=Y and BACKUP FREQUENCY = 0, OAM schedules the first backup copy to be written immediately after the primary copy of the object is successfully stored. Otherwise, any backup copies are made during the first storage management cycle after the object is stored, or during the first storage management cycle after a new management class is assigned for the object.

OAM suggests you use the immediate object backup cautiously as it might impact the performance of storing and retrieving primary copies of objects.

NUMBER OF BACKUP VERSIONS (DATA SET EXISTS)
Specify the number of backup versions to be made for an object when OSMC processing is done for an Object storage group.

Valid values for the NUMBER OF BACKUP VERSIONS (DATA SET EXISTS) field are as follows:

- 0—creates one backup copy

  Note: When AUTO BACKUP=Y in the management class construct, ISMF/SMS will not accept a value of "0" for the number of backup versions.
- 1—creates one backup copy
- ≥2—creates two backup copies

Page 4 of the Management Class Define panel is not used for data sets; it is used for objects to define an event that causes OAM to invoke ACS for selecting a new storage class or management class. These class transition events are:

TIME SINCE CREATION
Specify the time (YEARS, MONTHS, and DAYS) that must elapse relative to the date the object was created. The YEARS, MONTHS, and DAYS attributes can be used separately or in combination. A maximum date of 9999/12/31 will be used if TIME SINCE CREATION results in a date exceeding the maximum.

TIME SINCE LAST USE
Specify the time (YEARS, MONTHS, and DAYS) that must elapse relative to the date the object was last referenced. If the object has not been referenced since it was stored, the create date is treated as the last reference. The YEARS, MONTHS, and DAYS attributes can be used separately or in combination. A maximum date of 9999/12/31 will be used if TIME SINCE LAST USE results in a date exceeding the maximum.

Restriction: Do not use UPD=N on the OAM*n* statement in the IEFSSN*xx* member of PARMLIB if this option is used in your management class.

PERIODIC
Specify that a class transition will occur at a regular period, based on the calendar (regardless of when the object was created or referenced). This parameter has five attributes, which can be used either separately or in combination:

MONTHLY ON DAY
Specify FIRST, LAST, or a number 1 - 31 indicating the day of the month on which class transition should occur; leave blank if unused.

QUARTERLY ON DAY
Specify FIRST, LAST, or a number 1 - 92 indicating the day of the quarter on which class transition should occur; leave blank if unused.

QUARTERLY IN MONTH
Specify a number 1 - 3 indicating the month in each quarter on which class transition should occur; leave blank if unused.

YEARLY ON DAY
Specify FIRST, LAST, or a number 1 - 366 indicating the day of the year on which class transition should occur; leave blank if unused. For example, choosing the number 366 allows the transition to occur on 1/1 of the next year. In the event of a leap year, OSMC causes the transition to occur on 12/31 of the current year.

YEARLY IN MONTH
Specify a number 1 - 12 indicating the month of the year on which class transition should occur; leave blank if unused.

Restriction: You cannot specify the TIME SINCE CREATION, TIME SINCE LAST USE, and PERIODIC attributes together.

An object’s management class association can change as a result of an application request or a class transition. Should a change occur, OAM applies the new management criteria. This might result in various actions, such as:

- Up to two backup copies might be made where one did not exist before.
- An object’s lifetime might be decreased or increased.
- A new class transition event can cause the invocation of ACS routines in the future.

As you define management classes and prepare and review your implementation of class transition using the Automated Class Selection routines, it is critical to analyze the result of your class transitions to avoid processing inefficiencies, unexpected results, or both.

The usage of TIME SINCE CREATION and TIME SINCE LAST USE attributes must be carefully studied to ensure that one of the class transitions in a series of class transitions assigns a management class, which causes the next class transition to occur **in the future** or the object to expire. Ensuring a management class is assigned to cause the next class transition to occur in the future is accomplished through your extensions to the operating system in the Management Class Automatic Class Selection routine.

Restriction: Do not use the TIME SINCE LAST USE or the TIME SINCE LAST REFERENCE attributes if you are using the new parameter (UPD=N) on the CBRINIT line in the IEFSSN*xx* PARMLIB member with no pending action date. See Updating the IEFSSNxx PARMLIB member for more information.

If your implementation allows for a series of class transitions that do not result in a class transition that is scheduled in the future, or do not result in an object expiring and being deleted, the results of the storage management cycle might be affected. Depending on the number of objects that are processed, operational conditions, or possible processing interruptions due to contention, it is likely that processing will be seriously degraded. This could potentially force the storage management process into a loop that attempts to identify a future date for class transition processing or expiration for one or more objects.

If at any time an object's management class results in the object's expiration date being set to 9999/12/31 while that object is on removable media, that volume's expiration date will be set to 9999/12/31. This will cause the volume to never expire, even if the object's management class changes later allowing the object to expire. Be aware of the effects of expiration dates that can be set by a management class, even if it is being used as an interim management class for an object. This expiration date can be modified using the MODIFY OAM,UPDATE command.

Defining automatic class selection

15g ***Define and test ACS routines.***

You must perform this step at initial installation. During migration, you might optionally perform this step if you have made changes in the SMS definitions and programs used by OAM.

You must supply ACS routines. "Automatic class selection" on page 469 contains listings of source code for sample ACS routines. There can be only one set of ACS routines and exits in an active configuration. This set applies to both objects and data sets. Installation exits are optional and allow you to perform functions that are not permitted in the ACS routines (for example, writing GTF records).

Related reading:

- See *z/OS DFSMSdfp Storage Administration* and *z/OS DFSMS Implementing System-Managed Storage* for information on using ISMF to define ACS routines.
- See *z/OS DFSMS Installation Exits* for information on writing ACS installation exits.

Validating and activating the SMS configuration

15h ***Validate and activate the SMS configuration.***

You must perform this step at initial installation. During migration, you might optionally perform this step if you have made changes in the SMS definitions and programs used by OAM.

You cannot use OAM until a configuration containing all the elements described in this topic have been defined and activated.

Only one SCDS can be activated at any time. Activating another SCDS or reactivating the current SCDS while OAM is running might cause OAM to restart. How soon OAM is notified of the SCDS activation depends on the time interval specified with the INTERVAL keyword in the IGDSMS*xx* PARMLIB member. OAM restarts if RESTART=YES is specified in the PARM string on the EXEC statement for CBROAM in the OAM procedure JCL. During this reinitialization, all libraries and drives are set to either online or offline according to the attributes defined in the SCDS that caused OAM to restart. After the restart completes, all libraries and drives should be displayed and set to the desired operational status.

See *z/OS DFSMSdfp Storage Administration* for information about validating and activating the configuration that you have just defined.

16 Running the OAM Installation Verification Program for object support

16 ***Run the OAM Installation Verification Program for object support.***

You must perform this step at initial installation and at migration.

The OAM Installation Verification Program (CBRSAMIV) verifies that OAM object support is successfully installed and operational. This program activates the OSREQ macro, and allows dummy objects to be tested by having the storage administrator perform OSREQ functions against them without having to perform an explicit OSREQ ACCESS to connect the macro to OAM, or without having to perform an OSREQ UNACCESS to disconnect the macro. Run this job (see "OAM installation verification program and OAMUTIL" on page 464 for a sample of this job), and perform some OSREQ functions as a test to insure the product is successfully installed.

17 Adding new tape devices to the OAM object tape configuration

17 *Add new tape devices to the OAM object tape configuration.*

You must perform this step when installing new tape devices.

To migrate to and implement new tape devices in the OAM object tape configuration, follow the instructions in *z/OS Migration* or in the publication for the new tape device. These publications describe the procedures for migrating to the software support in a stand-alone environment, an IBM tape library, or an OAM object tape user in an OAMplex.

18 Global z/OS UNIX configuration for OAM usage

18 *Configure for OAM usage.*

You must perform this step when preparing to use file system as an OAM storage level.

Specialized OAM installation procedures

It might be necessary to move the OAM application from one system to another to accommodate changes within an installation's storage management policy. The following information can assist you in moving OAM from one system to another (SYS1 and SYS2 are used as example system names).

Procedures for moving OAM to another system

This section provides information on moving OAM from one OAM system to another OAM system, *neither of which is part of an OAMplex*. To merge OAMs into an OAMplex, perform the procedure in "Merging OAMs into an OAMplex" on page 180.

The following steps are necessary when moving the entire OAM application from one single system to another single system. These steps can be followed after the installation and migration steps have been completed, should you need to move OAM from one system to another.

1. Install OAM and DB2 on system SYS2 by following the procedures in "High-level installation and migration checklists" on page 97.

   ____

2. Make copies of the OAM DB2 databases from system SYS1 using DB2 utilities. The OAM DB2 databases are the Object Storage Databases, Object Administration Database, and the OAM Configuration Database described in Appendix D, "Understanding OAM databases," on page 483.

   ____

3. Start DB2 on system SYS2.

   ____

4. Move the copies of the OAM DB2 databases to system SYS2 using DB2 utilities.

   ____

5. Delete all of the optical library and drive definitions from system SYS1 using the ISMF Optical Library Application.

   ____

6. If system SYS2 is not defined in the SCDS, then from system SYS1 add it as a valid system under the Control Data Set Definition Application of ISMF.

   ____

7. Move the SCDS from system SYS1 to system SYS2 by using the copy function of DFSMSdss. This only needs to be done if system SYS2 is not part of the same complex that is sharing the SMS SCDS.

   ____

8. Using the ISMF Storage Group Application on system SYS2, alter the Object and Object Backup storage group definitions to change the connectivity of SYS1 to not connected and the connectivity of system SYS2 to enabled.

   ____

9. Using the ISMF Optical Library Application on system SYS2, alter the library definitions connectivity from system SYS1 to system SYS2. This is only applicable for the controlling libraries. Altering these models automatically alters any connected library expansion units that are associated with the controlling libraries.

   Tip: You might need to delete and redefine all the 3995 operator-accessible drives.

   ____

10. Using the ISMF Optical Library Application on system SYS2, redefine all of the library definitions. Using the ISMF Optical Drive Application on system SYS2, redefine all of the drive definitions that are associated with the libraries. If the definitions do not exist in the DB2 OAM configuration database tables, they are added, but if they do exist, the information is updated in the DB2 tables. The information is added to the SCDS.

    ____

11. Activate this SCDS from system SYS2. Activation might be done from the operator console by using the SETSMS SCDS (source control data set name) command or by using the Activate Configuration option of ISMF.

    ____

12. Start OAM.

Merging OAMs into an OAMplex

Perform the following two steps if you wish to merge OAMs into an OAMplex.

1. Modify and run the CBRSMERG SAMPLIB job. CBRSMERG merges OAM configuration databases for use with DB2 data sharing. See Run the CBRSMERG SAMPLIB job for details.

   ____

2. Modify and run the CBRSG100 SAMPLIB job. CBRSG100 merges the DB2 object databases to use DB2 data sharing. See Run the CBRSG100 SAMPLIB job for details.

   ____

Adding OAM systems to an existing OAMplex

This section provides information on adding OAM systems to an existing OAMplex. When you initialize OAM, it looks for the OAMXCF statement in the CBROAMxx PARMLIB member. If the OAMXCF statement exists, this OAM system joins the XCF group, which means that it joins the OAMplex. Each OAM system in the OAMplex share the DB2 databases (object storage and OAM configuration databases).

1. Update the IGDSMS*xx* PARMLIB member for each OAM system that you are adding to the OAMplex. For more information, see "5 Changing system libraries" on page 104.

   a. For a classic OAM configuration, add the DB2SSID parameter to the IGDSMS*xx* PARMLIB member. The DB2SSID parameter specifies the name of the DB2 system. For a multiple OAM configuration, add the DB2 SSID or DB2 group attachment name to the appropriate OAM subsystem definition in the IEFSSN*xx* PARMLIB member.

b. Add the OAMPROC and OAMTASK parameters to IGDSMS*xx* only if you need to start the OAM address space to initialize SMS on the new system.

c. Issue the SET=SMS*xx* command to update the SMS information.

____

2. Update the CBROAM*xx* PARMLIB member to include the ONLYIF statement to specify whether specific statements, within the CBROAM*xx*, are to be processed on a given system.

```
ONLYIF SYSNAME(system_name|*ALL*)
                 OAMXCF statements
                 SETDISK statements
                 SETOAM statements
                 SETOPT statements
                 SETOSMC statements
```

Figure 17. CBROAMxx PARMLIB Member Samples Using the ONLYIF Statement and Optional Parameters

____

3. Update the OAMXCF statement in the CBROAM*xx* PARMLIB member to include the XCF group name and OAM member name for each OAM system that you are adding to the OAMplex. Each OAM system has a unique OAMMEMBERNAME and belongs to the same OAMGROUPNAME. For more information, see "OAMXCF statements in an OAMplex" on page 144.

```
OAMXCF OAMGROUPNAME(OAMGRP1)
       OAMMEMBERNAME(OAMSYS1)
       XCFTIMEOUT(XCFOPTREADA(20) XCFOPTREADM(50)
                  XCFOPTWRITEA(150) XCFOPTWRITEM(150)
                  XCFTAPEREADA(40) XCFTAPEREADM(50))
```

Figure 18. CBROAMxx PARMLIB Member Samples Using the OAMXCF Statement and Optional Parameters

____

4. Set up an OAM PROC with the OAM=*xx* parameter for the new system, where *xx* is the low-order suffix of your CBROAM*xx* PARMLIB member. For more information, see step **5h2** in "5h Updating the PROCLIB" on page 146.

____

5. Set the OSMC processing system to the preferred system for each Object and Object Backup storage group.

____

6. Update all the Object and Object Backup storage groups so that they are enabled to each system where OAM is running. For more information, see "Defining storage groups and relating the libraries to the storage groups" on page 170.

____

7. Ensure that the DB2 tables are in a data sharing group. Contact your DB2 administrator to set up the data sharing group.

____

8. If applications connect to DB2 and call OAM, the DB2SSID and the data sharing group that they connect to must be the same, or else the "Connects to two DB2s from one TCB" error displays. Contact your DB2 administrator for assistance with this step.

____

9. Run the CBRSMERG and CBRS100 jobs to merge the OAMs into the OAMplex at this point. For more information, see "Merging OAMs into an OAMplex" on page 180.

____

10. Start OAM on each system in the OAMplex. Display the `DISPLAY SMS,OAMXCF` command on each system to verify that they joined the OAMplex successfully.

    ____

Result: Now all of the OAM systems belong to the OAMplex.

Chapter 4. Administering OAM

This topic discusses the following typical OAM administrative tasks.

Topic	Page
"Destroying and deleting expired data" on page 238	"Destroying and deleting expired data" on page 238
"Invoking the OSREQ macro through the OSREQ TSO/E command processor" on page 243	"Invoking the OSREQ macro through the OSREQ TSO/E command processor" on page 243

Attention: Unless OAM is completely stopped, do not do any of the following:

- Stop or start table spaces or indexes related to the OAM databases.
- Start DB2 in maintenance mode.
- Run DB2 utilities against the OAM related databases.
- Update any of the DB2 tables related to the OAM databases.

Monitoring and maintaining the OAM configuration using ISMF

ISMF Library Management makes it possible to monitor and maintain information associated with the OAM configuration and the source control data set. The following information concerning ISMF deals only with its role in an optical storage environment. Although ISMF is used to manage tape libraries and their volumes, it is not discussed in this document in an attempt to prevent redundancy of material found in the *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

ISMF library management

The library management dialog allows you to generate lists of libraries or drives, or to display the attributes of a single library or drive. You can use the dialog to alter the definitions that were defined when OAM was installed or to add new definitions. For example:

- Optical Library Configuration:
  - Add, remove, or alter libraries and library drives
  - Add, remove, or alter operator-accessible drives
  - Redefine channel attachment

For detailed information about using ISMF, see *z/OS DFSMS Using the Interactive Storage Management Facility*.

Typical ISMF library management procedures

This section discusses the effects of some typical configuration maintenance tasks.

Defining an optical library or optical disk drive

The first time that you define an optical library or optical disk drive, the corresponding ISMF DEFINE panel requires that you enter all the attributes for that device in the appropriate panel fields based on the name of the library or drive. When you enter all the information, OAM adds a row that contains that information to the library or drive table in the OAM configuration database. An optical library or optical disk drive definition is also added to the specified SCDS in which the DEFINE occurs.

Subsequent definitions of the same optical library or optical disk drive into a new SCDS result in the REDEFINE panel being displayed with all the attributes associated with that device displayed as read in from the library or drive table. The optical library or optical disk drive definition is added to the specified SCDS.

For more detail on defining optical libraries and optical disk drives with ISMF, see "Sample ISMF session for an IBM 3995 Optical Library Dataserver" on page 395.

SCDS activation and restart

Only one SCDS can be activated at any time. Activating another SCDS or reactivating the current SCDS while OAM is running might cause OAM to restart. During this reinitialization, all libraries and drives are set to either online or offline according to the attributes defined in the SCDS that caused OAM to restart. After the restart completes, all libraries and drives should be displayed and set to the desired operational status.

Deleting an optical library or optical disk drive

Deleting an optical library or optical disk drive definition does not delete the entries from the OAM configuration database. Instead, upon deletion, the device definition is removed from the specified SCDS. If you must delete an optical library or optical disk drive from the OAM configuration database, use QMF, SPUFI, or a similar tool.

For more detail on deleting optical libraries and optical disk drives with ISMF, see "Deleting an optical library" on page 417, and "Deleting an optical disk drive" on page 418.

Altering an optical library or optical disk drive

Altering an optical library or optical disk drive results in an update to the corresponding database row to reflect the changes. Within the specified SCDS, the online status of the definition associated with the device is updated to reflect any change made to the online status.

The function used to alter an optical library or optical disk drive definition is only available when OAM is *not* running. This function prevents OAM from overwriting a change initiated from an ISMF Library Management dialog.

For more detail on altering optical libraries and optical disk drives with ISMF, see "Altering a 3995 optical library" on page 410, and "Altering an optical disk drive" on page 416.

Auditing an optical library

You can use the AUDIT line operator on the Optical Library List to perform inventory tasks against an entire 3995 optical library. The physical location of all the optical volumes associated with a 3995 optical library (full library audit) can be verified using the AUDIT line operator. The 3995-C3A, and pseudo optical libraries cannot be audited.

You also can audit an optical library using the MODIFY OAM,AUDIT operator command. See "Auditing a volume" on page 376 for more information.

Because the library audit might take a long time, a confirmation panel is displayed, asking you to confirm the AUDIT request. To confirm, type in "Y", then press ENTER. See Figure 19 on page 186 for the Confirm Audit Request panel.

```
  Panel  Utilities  Help
 ------------------------------------------------------------------------------
                            CONFIRM AUDIT REQUEST
Command ===>

To Confirm Audit of the Following Library:

   Library Name:  LIB1


   Enter "/" to select option      Perform Audit



   Note: If Audit is performed, Audit requests will be interspersed with other
         requests to the library. Audit may take a long time to complete.
         Use the HELP command for more information on Audit processing.





Use ENTER to Perform Operation;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 19. Confirm Library Audit Request Panel

AUDIT execution can be a lengthy process. During AUDIT execution, other activity in the library is **not** quiesced, and OAM prioritizes AUDIT requests lower than other requested functions. The following activities are examples of activities that OAM can still process while AUDIT is in progress:

- Requests to read objects from optical volumes within the library that is being audited.
- Requests to write objects to optical volumes within the library that is being audited.
- Optical volume ejects from the library that is being audited.

It might take **several** hours for you to receive notification that a full library audit or an extensive volume list audit has completed. Therefore, when scheduling an audit, be sure to take work load and time factors into consideration.

When you receive a completion message, you can enter the LISTVOL line operator to display the Mountable Optical Volume List for all optical volumes in the audited library. Check the VOLUME ERROR STATUS column for the results of the audit. OAM also issues messages for errors found. If you log off the TSO/E session before completion, the messages are stored in a broadcast data set and displayed the next time you log on. See Table 29 on page 194 for information on the AUDIT completion messages and the results displayed in the VOLUME ERROR STATUS column.

For more information about auditing a tape library using the AUDIT line operator, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Remapping an optical library

OAM keeps an inventory of the optical volumes in the OAM configuration database (OCDB). At the same time, the controller in the IBM 3995 optical library maintains an inventory of the optical volumes, which is called the *outboard inventory*. Usually these two volume inventories match, but there are some unusual conditions where they might go out of synchronization. Use the library audit and remap functions to synchronize the volume inventories in the OCDB and outboard inventory.

If audits or other functions indicate volume locations are incorrect, you can use the REMAP line operator to reconstruct the optical library's outboard inventory, by verifying the identity of each volume in the optical library.

REMAP also detects and corrects any discrepancies between the outboard inventory and the volume table in the OAM configuration database. The REMAP line operator is not available for tape libraries.

You also can use the MODIFY OAM,REMAP operator command to remap an optical library. See "Remapping an optical library" on page 377 for more information on the syntax and parameters for this command.

Enter the REMAP line operator next to a 3995 optical library on the Optical Library List. REMAP checks for the following conditions:

- The CDS name is "ACTIVE".
- The optical library is a real 3995 library. (You cannot use REMAP with a 3995-C3A or a pseudo library of any kind.)
- The optical library is online and operational.
- At least one internal drive is online and operational. The drive cannot be pending offline.

If these checks are successful, the REMAP Confirmation panel is displayed, asking you to confirm the REMAP. To confirm, type in "Y", then press ENTER. If the REMAP is accepted, the message REMAP SCHEDULED is displayed. Once the REMAP process is started, it must run to completion. There is no REMAP cancel function.

Recommendation: REMAP execution can take from 30 to 80 minutes to complete a full library depending on the number of online drives in the library. Therefore, when scheduling a REMAP, be sure to take work load and time factors into consideration. Also, once a REMAP is started, stopping OAM DOES NOT cancel REMAP processing. If you stop OAM in an attempt to cancel or stop a REMAP request, the library controller inventory must be rebuilt, and the REMAP continues to process in the library. There is no option of canceling a REMAP request once it is started.

When you issue a REMAP request for a library, all work in progress for the library is allowed to complete. However, new work requests and work requests scheduled but not yet started are rejected (except for drive vary requests that were queued prior to the REMAP request). Cartridges in the internal drives are demounted. If the library has an operator-accessible drive, upon completion of work in progress, the cartridge is demounted and the drive becomes unavailable for activity until the REMAP processing is complete. If an internal drive cartridge demount fails, REMAP processing fails; however, if the demount fails on the operator-accessible drive, REMAP processing continues on the internal drives.

Once you issue REMAP, the library controller reconstructs its inventory by going to each slot in the library. It verifies the identity of the cartridge in that slot by inserting that cartridge in a drive and reading both volume serial numbers. When the new inventory is complete, the host reads the inventory and compares all the inventory records with the volume records, and the volume records with the inventory. Updates are made to the appropriate host tables or the cartridge is ejected if the table cannot be corrected. Host volume records that indicate a volume is in the library but cannot be found in the outboard inventory are marked as shelf-resident and the volume status is updated as lost. All duplicate cartridges are marked by the library controller, causing the host to schedule the cartridge for eject. You are notified of the start of REMAP, all ejected cartridges, and of the completion of REMAP through a message to your TSO/E logon session.

When you receive a completion message, you can enter the LISTVOL line operator to display the Mountable Optical Volume List for all optical volumes in the remapped library. Check the VOLUME ERROR STATUS column, described in Table 27 on page 187, for error messages issued for errors detected by the REMAP operation.

Table 27. Remap results that appear in the VOLUME ERROR STATUS column

Result	Meaning
NO ERROR	Either no error occurs during the REMAP, or shows the initial status of the VOLUME ERROR STATUS column.
LOST VOLUME	A record in the volume table indicates a volume is in the optical library, but the volume cannot be found in the outboard inventory for that optical library.

The error messages are issued to the ISMF user who invoked the REMAP line operator. The messages contain the serial number (volser) of the volume for which the error was found and text indicating the type of error found. If you log off the TSO/E session before completion, the messages are stored in a broadcast data set and displayed the next time you log on. REMAP also might update the VOLUME LOCATION, VOLUME LIBRARY NAME, and MEDIA TYPE columns.

Types of discrepancies that REMAP resolves

Table 28 on page 188 shows how REMAP resolves various discrepancies.

Table 28. Discrepancies REMAP resolves between the outboard inventory and the OAM configuration database

Cause	Resolution
A cartridge has a media error. Possible causes are damaged media, the volume is a duplicate, or the cartridge is unformatted.	The cartridge is ejected from the optical library to allow it to be inspected.
The row in the volume table in the OCDB indicates the volume should be in another real optical library.	The volume is ejected from this library.
The paired volumes of a cartridge do not match the paired volumes in the OCDB.	The cartridge is ejected.
A volume physically resides in the library, but no row for that volume is found in the OCDB.	The cartridge is ejected.
The row in the volume table in the OCDB indicates the volume is on the shelf, but is physically residing in the library.	The volume location is changed to state the volume is residing in the library. The library name also is corrected if it differs from the library in which the volume resides.
The OCDB has a row for the volume but the outboard inventory does not have the corresponding volume.	The volume is lost. The VOLUME ERROR STATUS column for the volume is updated to say "LOST VOLUME". The volume is given a pseudo library name. The volume location is changed to "SHELF".
The volume media type in the OCDB is not compatible with the library device type.	The media type is corrected.

Monitoring and maintaining optical volumes

You can use the Mountable Optical Volume Application available from the Volume List Selection menu to maintain and verify optical volumes within the optical library.

For information about using ISMF and the Mountable Tape Volume Application with tape libraries and volumes, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

ISMF Mountable Optical Volume Application

A mountable optical volume resides on one side of an optical disk cartridge. The ISMF Mountable Optical Volume Application allows you to maintain optical volumes in an optical library through the use of the AUDIT, EJECT, and RECOVER line operators, or through the use of the AUDIT command. You can use line operators and commands to perform audits against volume lists, single volumes, or full libraries, eject optical disks from the library, and recover data stored on an optical disk that can no longer be read.

Restriction: You cannot use the ISMF RECOVER line operator to recover from second backup copies of objects. To recover a primary volume from the second object backup copy, use the MODIFY OAM command. See "Starting the OAM Volume Recovery utility" on page 270 for further details on this procedure.

Generating a mountable optical volume list

With the Mountable Optical Volume Application, you can generate a volume list of optical volumes for the following types of optical media:

3995 5.25-inch, single-density, WORM
3995 5.25-inch, double-density, WORM
3995 5.25-inch, quad-density, WORM
3995 5.25-inch, 8*x*-density, WORM
3995 5.25-inch, single-density, rewritable
3995 5.25-inch, double-density, rewritable
3995 5.25-inch, quad-density, rewritable
3995 5.25-inch, 8*x*-density, rewritable

Note: Double-, quad-, and 8*x*-density WORM also includes continuous composite WORM (CCW) media. WORM is write-once-read-many media. REWR is rewritable media.

z/OS DFSMS Using the Interactive Storage Management Facility describes all the columns in the Mountable Optical Volume Selection Entry panel.

Completing the Mountable Optical Volume Selection Entry panel

Follow these steps to bring up the Mountable Optical Volume Selection Entry panel.

1. Select option 2, VOLUME, from the ISMF Primary Option menu. ISMF displays the Volume List Selection menu.

   ____

2. Select option 2, MOUNTABLE OPTICAL, to generate a list of mountable optical volumes. ISMF displays the Mountable Optical Volume Selection Entry panel, shown in Figure 20 on page 189.

   ____

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                MOUNTABLE OPTICAL VOLUME SELECTION ENTRY PANEL
Command ===>

Select Source to Generate Volume List . . 2   (1 - Saved list, 2 - New list)

     1 Generate from a Saved List
         List Name  . .

     2 Generate a New List from Criteria Below
         Volume Serial Number . . . *         (fully or partially specified)
         Library Name . . . . . . . *         (fully or partially specified)
         Storage Group Name . . . . *         (fully or partially specified)
         Optical Media Type . . . . ALL       (See help for valid value)



       Enter "/" to select option     _ Respecify View Criteria
                                      _ Respecify Sort Criteria

Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 20. Mountable Optical Volume Selection Entry Panel

This section describes only the fields used to generate a new list (option 2).

GENERATE A NEW LIST FROM CRITERIA BELOW

Complete the following fields if you selected option 2 (default) for SELECT SOURCE TO GENERATE VOLUME LIST:

VOLUME SERIAL NUMBER

Enter a full or partial serial number of the volume or volumes to include in the list. The default value is an asterisk.

To include a single volume, enter a fully qualified volume serial number of 1 to 6 characters:

```
VOLUME SERIAL NUMBER ===> SYS001
```

For a partially qualified volume serial number, use asterisks as global volume serial number characters or percent signs as place holders.

For example, to include a range of volumes, enter a partially qualified volume serial number by using one or two asterisks as global volume serial number characters:

```
VOLUME SERIAL NUMBER ===> SYS*
```

Two asterisks are the maximum number of volume serial number characters allowed.

Use a single asterisk to specify all volumes that fit your other selection criteria:

```
VOLUME SERIAL NUMBER ===> *
```

This field is primed with the last value used. The default is *

LIBRARY NAME
Enter the 1 to 8 character name of an optical library, or a partially qualified name. This field is primed with the last value used. The default value is an asterisk.

STORAGE GROUP NAME
Enter the 1 to 8 character name of an SMS Object or Object Backup storage group, in the same way as you would for a volume serial number. This field is primed with the last value used. The default value is an asterisk.

OPTICAL MEDIA TYPE
Enter the 3 to 8 character name of the optical media type on which the volume resides. ALL is the default value. This field is primed with the last value used. Use the following values for optical media types:

3995WORM
3995 single-, double-, quad-, and 8*x*-density, WORM, optical disk volumes

3995REWR
3995 single-, double-, quad-, and 8*x*-density, rewritable, optical disk volumes

3995-1
3995 single-density, WORM and rewritable, optical disk volumes

3995-1WO
3995 single-density, WORM, optical disk volumes

3995-1RW
3995 single-density, rewritable, optical disk volumes

3995-2
3995 double-density, WORM and rewritable, optical disk volumes

3995-2WO
3995 double-density, WORM, optical disk volumes

3995-2RW
3995 double-density, rewritable, optical disk volumes

3995-4
All 3995 quad-density, rewritable or WORM, optical disk volumes

3995-4RW
All 3995 quad-density, rewritable, optical disk volumes

3995-4WO
All 3995 quad-density, WORM, optical disk volumes

3995-8
All 8*x*-density, rewritable or WORM, optical disk volumes

3995-8RW
: All 3995 8*x*-density, rewritable, optical disk volumes

3995-8WO
: All 8*x*-density, WORM, optical disk volumes

ALL
: Select all available optical media types.

Final step: Generating the list

After entering the information you want on the Mountable Optical Volume Selection Entry panel, you are ready to generate the list. Press the ENTER key to display the volumes that meet your selection criteria. Figure 21 on page 191 and Figure 22 on page 191 show columns 14 through 21 of the Mountable Optical Volume list. Scroll the list to see all the columns.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                        MOUNTABLE OPTICAL VOLUME LIST
Command ===>                                                  SCROLL ===> HALF
                                                       Entries 1-7 of 7
Enter line operators below:                            Data Columns 14-16 of 21
    LINE       VOLUME                           MEDIA
    OPERATOR   SERIAL  SHELF LOCATION           TYPE      VOLUME ERROR STATUS
   ---(1)----  -(2)--  -----(14)--------------  --(15)--  ---------(16)-------
               SYS090  COMPUTER CENTER LIB 7     3995-1WO  NO ERROR------------
               SYS092  COMPUTER CENTER LIB 7     3995-1RW  NO ERROR------------
               SYS093  ------------------------  3995-1RW  --------------------
               SYS095  OFFSITE                   3995-2RW  VOLUME NOT FOUND----
               SYS096  SHELF 2                   3995-2WO  WRONG VOLUME IN SLOT
               SYS097  SHELF 2                   3995-2WO  NO ERROR------------
   ----------  ------  -----------  BOTTOM OF DATA  -----  --------------------




Use HELP Command for Help; Use END Command to Exit.
```

Figure 21. Mountable Optical Volume List, Columns 14 through 16

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                        MOUNTABLE OPTICAL VOLUME LIST
Command ===>                                                  SCROLL ===> HALF
                                                       Entries 1-7 of 7
Enter line operators below:                            Data Columns 17-21 of 21

    LINE       VOLUME  CAPACITY  VOLUME       ENTER OR    PSEUDO    OAM INSTANCE
    OPERATOR   SERIAL  (IN MB)   CREATE DATE  EJECT DATE  LIB NAME  MEMBER NAME
   ---(1)----  -(2)--  --(17)--  ---(18)---  ---(19)---   --(20)--  ---(21)-----
               SYS090       320  1992/07/29  1992/07/29   --------  ------------
               SYS092       320  1992/10/22  1992/10/22   --------  ------------
               SYS093       320  1992/10/22  1992/11/22   --------  ------------
               SYS095       640  1992/11/04  1992/11/04   --------  ------------
               SYS096       640  1992/11/10  1992/11/10   PLIB2     OAMIMEMBER1
               SYS097       640  1992/12/20  1992/12/20   PLIB1     OAMIMEMBER2
   ----------  ------   -------  BOTTOM  OF  DATA  ----   --------  ------------



Use HELP Command for Help; Use END Command to Exit.
```

Figure 22. Mountable Optical Volume List, Columns 17 through 21

For information on columns 1–13 on the Mountable Optical Volume list, see *z/OS DFSMS Using the Interactive Storage Management Facility*.

SHELF LOCATION
: The physical location of the optical volume that resides outside (shelf-resident) of an optical library.

MEDIA TYPE

Displays the type of optical media upon which an optical volume resides. The MEDIA TYPE field applies to all optical volumes. The valid values are:

3995-1WO
3995 single-density, WORM, optical disk media

3995-1RW
3995 single-density, rewritable, optical disk media

3995-2WO
3995 double-density, WORM, optical disk media

3995-2RW
3995 double-density, rewritable, optical disk media

3995-4RW
3995 quad-density, rewritable, optical disk media

3995-4WO
3995 quad-density, WORM, optical disk media

3995-8WO
3995 8*x*-density, WORM, optical disk media

3995-8RW
3995 8*x*-density, rewritable, optical disk media.

UNKNOWN
The REMAP function encountered an unknown MEDIA TYPE when it processed volumes in a 3995 library. "UNKNOWN" only occurs with the display of "LOST VOLUME" in the VOLUME ERROR STATUS column.

????????
If the value cannot be displayed because of an error, the following columns (which depend upon a valid MEDIA TYPE) display question marks: FREE SPACE, %USED, and VOLUME ERROR STATUS

VOLUME ERROR STATUS

Shows the error status of individual optical volumes. The VOLUME ERROR STATUS shows the status after the volume is audited, or the volume status after the remapping of a 3995 optical library.

CAPACITY (IN MB)

Shows the raw unformatted capacity in megabytes of the optical disk volume.

VOLUME CREATE DATE

Shows the date that the volume was created and initially labeled in the form YYYY/MM/DD, where YYYY is the year, MM is the month of the year, and DD is the day of the month.

ENTER OR EJECT DATE

The date that the volume was entered into the optical library if the volume is currently library-resident. If the volume is currently shelf-resident, this date is when the volume was last ejected from an optical library. The format is YYYY/MM/DD, where YYYY is the year, MM is the month of the year, and DD is the day of the month.

PSEUDO LIB NAME

The name of the pseudo library if the volume is a shelf-resident volume. This field should match the library name field when the volume is shelf-resident. If the volume is library-resident, this name is the target pseudo library to which the volume is assigned when it is ejected from the real library.

OAM MEMBER INSTANCE NAME

The instance of OAM in an OAMplex that is currently managing and controlling this volume. If the volume is library-resident, this name is the member associated with the OAM where the optical library is currently physically online. If the volume is shelf-resident and the member name is not blank, the volume is currently mounted on an operator-accessible drive, which is currently online to the OAM identified by this member name.

Viewing and sorting a list

You can sort and tailor a list with the View and Sort options on the Mountable Optical Volume List Selection Entry panel.

z/OS DFSMS Using the Interactive Storage Management Facility discusses the View and Sort options in more detail.

Maintaining and verifying the volume list

You can use the AUDIT, EJECT, and RECOVER line operators, and the AUDIT command, to maintain and verify optical volumes in your optical library, and if errors are found, you can reconstruct an accurate list using the REMAP line operator. The line operators affect individual volumes. In contrast, the commands affect all eligible volumes on the Mountable Optical Volume list, except for the optical volumes that you choose to hide. You can issue an audit from the host application through ISMF Library Management or Mountable Optical Volume Application.

Press PF1 on the Mountable Optical Volume list for help information about the line operators and list commands.

For more information on auditing and remapping optical libraries, see "Remapping an optical library" on page 186 and "Auditing an optical library" on page 185.

Verifying optical volumes using audit

AUDIT can be invoked as an ISMF line operator on the Mountable Optical Volume List panel (Single Volume AUDIT) or the Optical Library List panel (Full Library AUDIT). The storage administrator uses the AUDIT line operator to verify the physical location of an optical volume. The audit compares the volume information maintained in the OAM configuration database and the optical library outboard inventory with the actual location of the optical volume. AUDIT does not just compare the OAM configuration database with the outboard inventory; it actually causes the volume to be mounted and reads the volume serial number to verify that the volume is in its assigned storage location.

Tip: You also can use the MODIFY OAM,AUDIT,VOLUME command to audit a volume. See "Auditing a volume" on page 376 for more information.

When you invoke the AUDIT line operator successfully, AUDIT SCHEDULED is displayed on the Mountable Optical Volume list. If the volume is successfully scheduled for an audit, the volume has ***AUDIT** displayed in the line operator column. Audits that are not successfully scheduled have **¬AUDIT** or **?AUDIT** in the line operator column. ISMF also displays a short message explaining why the audit is not scheduled (see Table 31 on page 196 for more information).

AUDIT can also be invoked as an ISMF command to audit all the eligible optical volumes on the Mountable Optical Volume List (Volume List AUDIT). ISMF is an important part of the AUDIT scheme because it allows you to start with an entire optical volume list, and by using sorting and filtering capabilities, you reduce that list to a subset of volumes; for example, all the volumes in storage group *x*. You can then use the AUDIT command to request an audit of all volumes in that subset list. (The AUDIT command does not affect volumes that you have hidden using the HIDE line operator.)

AUDIT functions help you ascertain the physical location of optical volumes by verifying whether or not a library volume resides in the location that is listed for that volume in the optical library outboard inventory. The library controller maintains the internal library location of the cartridges in the outboard inventory. The host also identifies which library contains each cartridge in the volume table. If the host record or records do not match the controller inventory when OAM performs an audit, then you must correct the records, the inventory, or both. The AUDIT functions do not perform any corrective actions; their purpose is verification only. See "Remapping an optical library" on page 186 for a description of corrective actions that you can take if the audit is unsuccessful.

Note:

1. The AUDIT function is available for the 3995 with the exception of the 3995-C3A models and pseudo optical libraries.

2. AUDIT functions only process volumes that are known to OAM. If an unknown volume is in the library, the AUDIT functions do not detect this condition. REMAP functions are necessary to locate and eject the unknown volume.
3. AUDIT functions from the host application and the 3995 dynamic console are not the same. In a z/OS installation, the AUDIT function that is available from the 3995 dynamic console is intended for use during hardware service; AUDIT functions that are issued from the 3995 dynamic console do not communicate with OAM at the host. The AUDIT function from the 3995 dynamic console performs the same hardware function as a library REMAP. In a z/OS installation, issue the AUDIT functions from the host application and not from the 3995 dynamic console; otherwise, the host and the library controller can differ. This manual describes only the command functions available from the host application through ISMF Library and Volume applications and z/OS operator commands.

   You can find further information about commands from the 3995 dynamic console in the *3995 Model 153, 151 Operator Guide* or the *3995 Operator Guide for C-Series Models*.

ISMF AUDIT provides three scopes:

- Single volume audit (AUDIT line operator)
- Volume list audit (AUDIT command)
- Full library audit (AUDIT line operator)

For each volume audited, three conditions must be present:

- Volume must be obtained from its assigned storage slot.
- Volume must be mounted.
- Internal label of the volume must be read and verified as matching the entry for that storage slot/volume combination in the outboard inventory.

When you receive an audit completion message, you can use the REFRESH command to update the Mountable Optical Volume list with the same selection criteria. The results of the audit are shown in the VOLUME ERROR STATUS column and you also receive a message with the error results. If you log off before the results are obtained, messages are stored in the users broadcast data set and displayed during the next logon process. VOLUME ERROR STATUS contains only the last error found; no history is kept. No attempts are made to fix the problems at the time of detection because, based on the error found, the software is unable to determine exactly what the corrective action should be. Use REMAP, discussed in "Remapping an optical library" on page 186, to correct the problems. Table 29 on page 194 lists possible results of auditing an optical volume.

Table 29. Auditing results that appear in the VOLUME ERROR STATUS column

Result	Meaning
NO ERROR	Either no error occurs during the audit, or shows the initial status of the VOLUME ERROR STATUS column.
NOT IN THE LIBRARY	The volume has an entry in the OAM configuration database, but no corresponding entry in the outboard inventory.
VOLUME NOT FOUND	The volume was not found in its assigned storage slot. Either the slot is empty or another volume was found.
VOLUME IN WRONG SLOT	This volume was found while auditing another volume or when attempting to mount another volume.
MEDIA ERROR	An error occurred when the volume serial number was read.

Note: You cannot audit optical volumes that are on a shelf.

When you request an AUDIT, you must specify all volsers to be audited, even if two of the volsers represent opposite sides of the same cartridge. When you specify a full library AUDIT using the AUDIT line operator, all volume serial numbers listed in the host inventory as residing in that library are audited.

Whenever a full library AUDIT or a volume list AUDIT is requested, a confirmation panel is displayed. This panel gives you the opportunity to confirm or deny the audit request. To confirm, type in Y, then press ENTER. See Figure 23 on page 195 for the Confirm Audit Request panel.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                           CONFIRM AUDIT REQUEST
Command ===>

Number of Volumes to be Audited: 97


Specify the Following:
   Enter "/" to select option      Perform Audit



   Note: If Audit is Performed, Audit Requests will be Interspersed with other
          Requests, with the Audit Request having low priority.






Use ENTER to Perform Operation;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 23. Confirm Optical Volume Audit Panel

When you receive an audit completion message, you can use the REFRESH command to update the Mountable Optical Volume List with the same selection criteria. The results of the audit are shown in the VOLUME ERROR STATUS column, discussed in Table 29 on page 194. If you log off before completion, resulting messages are stored in your broadcast data set and displayed at the next logon.

For information concerning using the AUDIT line operator with tape libraries, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

AUDIT messages

After confirming the audit request, a generic message is displayed on the Mountable Optical Volume List indicating whether all, some, or none of the optical volumes were scheduled for an audit. Table 30 on page 195 explains the meaning of the generic AUDIT messages.

Table 30. Generic AUDIT messages

Generic AUDIT message	**Meaning**
AUDIT SCHEDULED	All optical volumes were successfully scheduled for audit.
SELECTED AUDIT SCHEDULED	Only the eligible optical volumes were scheduled by ISMF for audit. Optical volumes on the shelf or in an optical library are ineligible for audit.
PARTIAL AUDIT SCHEDULED	At least one optical volume was rejected by OAM and at least one optical volume was successfully scheduled for the audit.
NO VOLUMES SELECTED	None of the optical volumes were eligible for the audit by ISMF.
AUDIT FAILED	All the optical volumes were rejected for the audit by OAM.

The AUDIT command shows in the LINE OPERATOR column for each individual optical volume whether the audit was successfully scheduled, as shown in Table 31 on page 196.

Table 31. Specific AUDIT messages	
AUDIT message	**Meaning**
*AUDIT	The optical volume was scheduled for an audit.
¬AUDIT	The optical volume was not eligible for audit.
?AUDIT	The audit request for the optical volume was rejected.

The AUDIT command functions provide a message history for each optical volume, and ISMF log entries about rejected audit requests.

See "Handling OAM scheduling errors" on page 196 for details.

Ejecting optical disks

The EJECT line operator schedules the mountable optical disk to be moved to the input/output station of the optical library. When you submit the line operator successfully, EJECT SUBMITTED is displayed on the Mountable Optical Volume List. After the eject completes, OAM issues a completion message.

When an optical disk is ejected, the operator might be prompted on the system console to supply a shelf location and a pseudo library to assign the ejected volume to. Ejected optical disks are stored according to the optical shelf conventions established at your computer site. The information on the shelf location and the pseudo library is stored in the OCDB for later use when the optical disk needs to be obtained from the shelf and a drive selected for its use. When the optical disk is obtained from the shelf, it is mounted within an operator-accessible drive, or within a library if no outstanding mount exists for the cartridge.

After receiving the EJECT completion message, you can use the REFRESH command to update and view the shelf location and pseudo library information for an optical volume in the SHELF LOCATION column.

See "Ejecting an optical disk" on page 296 for more information on the EJECT command, including command syntax.

Recovering optical disks

The RECOVER line operator allows you to invoke the Volume Recovery utility to recover data stored on an optical disk that can no longer be read. When you successfully enter the RECOVER line operator against an optical volume serial number, RECOVER SUBMITTED is displayed on the Mountable Optical Volume list. Subsequent processing is the same as that performed when an operator command is used to invoke the Volume Recovery utility.

Before you enter the RECOVER line operator, become familiar with the prerequisites and the dialogue that occurs with the system operator.

Restriction: You cannot use the ISMF RECOVER line operator to recover from second backup copies of objects. Use the MODIFY OAM command to recover a primary volume from the second object backup copy. See "Starting the OAM Volume Recovery utility" on page 270 for further details on this procedure.

For more information on the Volume Recovery utility, see "Recovering an entire optical cartridge or tape volume" on page 218.

Handling OAM scheduling errors

The following information provides assistance in handling OAM scheduling errors.

Message history for AUDIT commands

The AUDIT command or line operator provides a message history for each optical volume. Enter the MESSAGE line operator next to any optical volume serial number to display the short message for the specific volume. Press PF1 to display the additional long message.

ISMF log entries about rejected requests

OAM schedules AUDIT, REMAP, and EJECT requests. If OAM rejects the requests, or an error occurs during the scheduling of the request, the OAM errors are recorded as ISMF log entries. An example of a rejected request is an ?AUDIT in the LINE OPERATOR column for a specific optical volume. See the feedback area of the ISMF log entry for the return codes and reason codes for OAM errors. The ISMF log entries are in the ISPF Transaction Log.

For the meaning of the OAM return and reason codes, see *z/OS DFSMSdfp Diagnosis*.

Errors after a request is scheduled

If discrepancies are found after issuing an AUDIT command, they are noted and related to you by the following means:

- Scheduling error messages for full library audits are issued to you by the z/OS SEND interface unless the scheduling error occurred prior to any volumes from the library being successfully scheduled and that error was severe enough to prevent any other volumes in the library from also being scheduled. This early detected severe error is reported on the ISMF help panel. SEND messages contain the volser (if known) of the volume for which the error was found and text that indicates the type of error found in attempting to validate an audit request.

  If the scope of the audit is volume list or single volume, scheduling errors are not reported to you by the z/OS SEND interface. These errors are indicated on return to the ISMF panel from which the AUDIT was initiated and can be interrogated by using the ISMF message and help panels.
- Errors incurred while attempting to perform the physical audit for any of the three audit scopes, single volume, volume list, or full library are reported to you by the z/OS SEND interface.
- After auditing a volume, the error status field (ERRSTAT) of the OCDB volume record is updated. As notification that the audit is complete and the error status fields can be reviewed, a completion message is sent to you through your TSO/E logon session. If you are not logged on when OAM issues these errors or completion messages, they are saved in the SYS1.BRODCAST data set. You will receive these messages the next time you log on to TSO/E.

For more information about the ISMF message and help panels, see *z/OS DFSMSdfp Storage Administration*.

Monitoring and maintaining SMS construct definitions

ISMF Library Management makes it possible to monitor and make changes to the SMS constructs.

For detailed information about using SMS constructs and ISMF options for storage administrators, see *z/OS DFSMSdfp Storage Administration*.

Changing SMS construct definitions

As installation requirements change, it might be necessary to update definitions of the storage classes, management classes, storage groups, data class, and ACS routines. You can modify definitions for these constructs using the ISMF ALTER panels for all OAM object environments.

Perform these updates with caution because objects that do not require processing after the definition is changed are not affected by the change, even though they are assigned to the class to which the new definition applies. The updated definitions are used *only* for objects entering the system, or processed by the system, after the change. This situation is particularly true in changes to management class definitions that affect retention, backup, or class transition.

Maintaining storage class definitions

A new storage class might be needed to define the performance requirements of a new application. In addition, you need to add or modify management class definitions and the ACS routines to manage objects that use the new storage class.

Storage class definitions might be changed to accomplish the following functions:

- Add storage classes.

- Alter the INITIAL ACCESS RESPONSE SECONDS to store data on removable media (optical or tape volumes depending on the Sustained Data Rate) instead of disk space.
- Alter the OAM sublevel to store objects in tape sublevel 2 volumes or disk sublevel 2 (file system).

Maintaining management class definitions

When the need for a new object cycle is recognized, it usually leads to the definition of a new set of management classes for the phases in the new cycle. It becomes necessary to add statements to the ACS routine to process the transitions for the new management classes.

Consider the effect of the changed management class definitions on objects that currently exist. The change of a management class might imply conversion action which is not supported by OSMC. For example, a transition rule could be changed to cause the schedule date to occur a month earlier. OSMC would not process the object until the scheduled date assigned using the previous transition rule, which is a month later than is specified by the updated definition.

Management class definitions might be changed to accomplish the following functions:

- Add management classes.
- Alter backup requirements.
- Change retention criteria.
- Modify class transition parameters.

Maintaining object storage group definitions

New Object storage groups might be needed for physical separation of new types of objects. In addition to defining the new Object storage groups, you might need to change the installation's object naming conventions and modify the ACS routines to use the new naming conventions to assign objects to the new storage groups.

Object and Object Backup storage group construct definitions might be changed to accomplish the following functions:

- Define a different storage management cycle start window.
- Change DRIVE STARTUP threshold for optical.
- Change the criteria used to determine when optical volumes are marked full.
- Add a real or pseudo optical library.
- Remove an optical library (for migration to a newer library or media).
- Change the OSMC processing system name.
- Change system connectivity.

Modifying data class definitions

You might need to define or modify data classes for objects stored in Object and Object Backup storage groups to use new tape devices, for example.

Data class definitions might be changed to accomplish the following functions:

- Specify a media type for a tape device.
- Specify a recording technology for a tape device.
- Specify performance scaling or performance segmentation for a tape device.

Recommendation for 3592 Model J1A tape devices: If you are not using the performance scaling option, run in 3590-1 emulation mode to use the tape volumes effectively because OAM uses the full capacity of the volume. If you run in 3490E emulation mode, OAM might run out of tape blockids before it uses the full capacity of the volume, depending on the capacity value and the size of the objects being written.

Modifying default storage and management classes

Tip: Although IBM strongly advises against direct modification of the DB2 tables, it is sometimes necessary. Back up relevant data before modifying the DB2 tables. Also, quiesce any system activity that might be active for the collection or storage group so the application does not receive errors if it attempts to access the collection data while changes are made.

To determine the current default storage class and management class that is associated with a given collection, you can use a SPUFI SELECT for the collection name row of the DB2 Collection Name Identifier Table in the Object Administration Database.

There are two methods available to modify both the default storage class and default management class that is associated with a given object collection.

The CHGCOL procedure

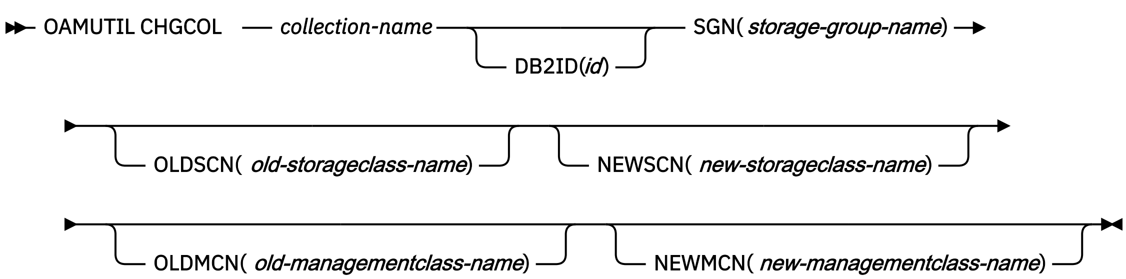
The recommended procedure is to use the OAMUTIL CHGCOL function included in the CBRSAMUT SAMPLIB member. To invoke this utility, modify and run the CBRSAMUT SAMPLIB job or issue a TSO/E OAMUTIL command.

Select this utility to perform the following tasks:

- Change the default Storage Class Name only
- Change the default Management Class Name only
- Change both default Storage Class Name and default Management Class Name.

The following is the syntax for the OAMUTIL CHGCOL command:

The OAMUTIL CHGCOL command syntax



DB2ID
: Specifies the DB2 SSID (for a single OAM) or DB2 group attachment name (for an OAMplex) of the DB2 subsystem that the preferred OAM subsystem (and optionally a started OAM address space) has been configured to use. The *id* can be 1 - 4 characters. In a multiple OAM configuration, the DB2ID keyword is required.

Manual procedure

Use SPUFI UPDATE to change either the storage class, management class, or both in the DB2 Collection Name Identifier Table to the preferred value. Ensure that these values in the SMS CDS are valid.

The new default storage class, or management class, or both only applies to new objects stored after these changes are complete.

For details on using the OSREQ macro, see *z/OS DFSMS OAM Application Programmer's Reference*.

Changing ACS routines

As mentioned above, ACS routines might need to be changed to implement changes in storage group, management class, storage class, or data class definitions. Defining new storage groups, storage classes, management classes, and data classes has no effect unless the ACS routines are changed to select those new constructs.

ACS routines can be changed to accomplish the following functions:

- Provide initial class defaults for new collections.

- Cause an object to move differently in the hierarchy by assigning a different storage class at class transition.

Tip: Defining new classes does not always mean new values for parameters; a new class can have the same parameters as an existing class. A new class might be created to make the relationship between a class and an application more understandable. This action makes it possible to modify parameters later to fit the needs of one application without affecting other applications. For example, adding a new management class that has the same backup parameter as an existing class allows you to change the backup parameter later for the new application's objects without changing the backup requirements for other objects associated with the original class.

Attention: Changing existing constructs might not affect all objects associated with those constructs. Only those objects being stored or encountering a class transition after the construct definition is changed is affected. A change to storage class or management class takes effect at the next storage management cycle only if the object needs management (such as class transition). For example, a change in the INITIAL ACCESS RESPONSE SECONDS parameter in a storage class might not cause any or all objects with that storage class to move within the storage hierarchy.

Monitoring DB2 databases

You can use the following techniques to obtain information about performance and space allocation of DB2 databases, tables, and indexes that are used by OAM:

- DB2 RUNSTATS utility
- DB2 STOSPACE utility
- SQL statements

DB2 RUNSTATS utility

RUNSTATS is a DB2 utility that scans a table space or indexes to gather information about space utilization and index efficiency. The information gathered is stored in the DB2 system tables and used by the SQL optimizer to select the best access paths during the bind process.

Run RUNSTATS to help evaluate the design of the database and determine when the REORG utility should be run for specific table spaces or indexes.

The output from RUNSTATS consists of DB2 updates to any or all of the following tables, depending on whether RUNSTATS was executed for a table space, indexes, or both:

- SYSIBM.SYSCOLUMNS
- SYSIBM.SYSINDEXES
- SYSIBM.SYSTABLES
- SYSIBM.SYSTABLESPACE
- SYSIBM.SYSTABLEPART
- SYSIBM.SYSINDEXPART

By doing a global SELECT on the SYSIBM tables that were updated by RUNSTATS, you can determine what action, if any, should be taken to improve the performance of the system.

Attention:

1. Use caution. When RUNSTATS is active, no requests can be made to the affected tables.
2. After using RUNSTATS, rebind application plans that use the tables or indexes that were the subject of the RUNSTATS. This rebuild allows the DB2 optimizer to take advantage of new information about the structure of indexes.
3. After rebinding, examine the PLAN_TABLE output from the SQL EXPLAIN statement to ensure that all indexes are used. If PLAN_TABLE indicates that indexes are *not* used, override the DB2 catalog

statistics, using the procedure specified in the "Maintaining data organization and statistics" in IMS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPH2).

DB2 STOSPACE utility

The DB2 STOSPACE utility determines the amount of space allocated for DB2 storage groups and their related table spaces and indexes. The utility updates DB2 system tables with the information it gathers.

Output from this utility consists of DB2 updates to the following tables:

- SYSIBM.SYSINDEXES
- SYSIBM.SYSTABLESPACE
- SYSIBM.SYSSTOGROUP

After STOSPACE execution is complete, use an SQL SELECT to view the tables that STOSPACE changed.

For RUNSTATS and STOSPACE utility syntax and usage notes, see the topic "DB2 online utilities" in IMS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPH2).

SQL statements

SQL statements can be used to determine the contents of various tables. You should be familiar with these tables to learn about the OAM databases as they relate to DB2.

For a complete listing of the DB2 catalog tables and what they contain, see the "DB2 reference information" topic in IMS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPH2).

Tuning OAM

Tuning OAM is largely a matter of tuning its various components. Remember that application design, although not under the control of the storage administration team, plays a significant role in OAM performance and efficiency.

Tuning OAM connections to DB2

When tuning OAM, there are a finite number of connections to DB2 from a batch environment that you need to consider. A number of functions, which are initiated by operator commands or automatically (for example, storage management cycle), might each result in multiple connections to DB2. OAM also establishes connections to process application requests. All of these connections established by OAM are in addition to the other necessary batch connections on your system. The total of all these connections at any given point in time must not exceed the DB2 limit because exceeding the limit causes OAM processing to fail.

The amount of concurrent function requests made of OAM control the tuning of OAM connections to DB2. Tuning OAM can involve limiting the number of concurrent functions requested automatically or by operator commands. A storage management cycle, for example, might establish three, four, or more connections to DB2 that persist for a good portion of the processing for each Object storage group. When deriving a value for MAXS, consider the number of connections, because MAXS controls the number of storage groups that the storage management cycle processes concurrently. While other calculations might seem to accommodate a larger number for MAXS, you must remember the DB2 limitation and adjust MAXS accordingly. Each installation is unique and must be tuned independently based upon actual experience; however, as a general guideline, as MAXS is increased above 10, the effectiveness of concurrency is diminished and might severely constrain processing in OAM or cause OAM processing to be unsuccessful. In an OAMplex, contention can increase with DB2 data sharing. When working with this type of environment, consider all the OAMs within the OAMplex when determining the storage group processing cycles and MAXS values for each instance of OAM.

Tuning the DB2 databases

It is important to run the DB2 utility RUNSTATS on all of the databases after a significant number of objects are stored and volumes are defined. Running this utility is likely to decrease the length of the DB2 instruction path and to improve performance.

Performance is generally improved when DB2 uses an index to locate an object or object directory entry in a DB2 table. The index scan access path provides more direct access to the data than the table scan access path.

For example, if the DB2 utility RUNSTATS is run on a storage group with only one collection-name and object-name or only one collection-name and pending-action-date, DB2 can choose the table scan access path for operations such as OSREQ DELETE of an object and OSMC object processing. On the other hand, if the DB2 utility RUNSTATS is run after there are a significant number of objects in the Object storage group, the index scan access path can be chosen by DB2.

After running DB2 utility RUNSTATS, rebind the OAM application plans.

OAM databases can use the following facilities:

- REORG utility—reorganizes table spaces and indexes
- DB2 trace facilities—report on various internal system events
- Index, table space, and buffer pool tuning options—allow control of performance-related factors
- Concurrency control mechanisms (locks)—can be manipulated to increase concurrency or to improve performance

DB2 performance can be significantly affected by providing channel separation and DASD device separation when allocating DB2 logs, the directory, and object databases for each Object storage group.

Table spaces are created with primary and secondary space. The secondary space is used when there is no more primary space. The secondary space is allocated from the DB2 storage group containing the primary space.

You must monitor the extension of table spaces into the secondary allocation to determine when to reorganize the individual database table spaces. It might be necessary to add volumes to the DB2 storage group so that additional extensions of the table spaces can occur.

For more information about tuning DB2 databases, see the "DB2 administration" topic in IMS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPH2).

Segmented table spaces

OAM specifies segmented table spaces in sample jobs and instructions for creating tables. This takes advantage of capabilities of segmented table spaces when doing INSERTs. The space maps for segmented table spaces provide a "guaranteed space" capability to find space in a table. Partitioned and simple table spaces lack this ability; however, there are instances where partitioned table spaces have definite advantages (see “Partitioning table spaces” on page 203 for more information).

OAM stores objects on DASD in DB2 tables. In most environments, a relatively static quantity of data is stored overall. New objects are usually stored in both a time-sequenced manner and with object names which cause inserts to occur in timestamp order. Data is constantly being deleted from DASD as it is moved by class transition to optical or tape, or simply expiring. In general, large blocks of space become free in the object tables during each OSMC cycle. This space is best reused when segmented table spaces are used as DB2 does not have to search on a target page to determine if there is space prior to insertion. The space maps are able to pinpoint available and sufficient space. This permits managing tables at a predetermined size with greatly reduced maintenance.

Some installations experience significantly longer processing time when segmented table spaces are not used, which causes increased search activity and more time consuming searches for available space. This reduced performance, primarily when using simple table spaces, can only be managed by constant management of the table space (reorganization activity and reallocation) to assure that there is always both space in the current extent of the data set that contains the DB2 table and sufficient extents available to guarantee space for storing objects. The use of segmented table spaces reduces the need for such manual management.

Restriction

Do not use simple table spaces with OAM object tables or object directory tables. Space searches and space reuse in simple table spaces consume a much greater proportion of processing time.

Partitioning table spaces

Partitioned table spaces permit large tables to be split into smaller entities which are managed more easily using DB2 utilities. Operations such as IMAGE COPY and REORG are more efficient and consume less total aggregate processing time when performed on smaller entities when tables are larger than 2 GB.

Partitioned table spaces are recommended when:

- Tables become very large
- Data might be relatively static for long periods of time
- DB2 maintenance must be minimized
- Any combination of these reasons

Backup and recovery actions for DB2 tables and table spaces are necessary under all circumstances. Regular IMAGE COPY operations and proper safeguard of DB2 logging is necessary to provide contingency for outages of any type.

Reorganizing tables is a different matter. Under circumstances where an object table or object directory table can be managed at a stable total allocation, segmented table spaces nearly eliminate any need to reorganize tables using the REORG operation. OAM uses DB2 indexes for all SELECTs and INSERTs as a consequence of its underlying design. The use of indexes removes the requirement for the tables to be in strict cluster index order. When a table is relatively new and is loaded with data, the RUNSTATS utility should be used to be certain that DB2 has good information on the order within the table in its catalog tables. Following RUNSTATS, a BIND with the EXPLAIN parameter should be performed to determine if DB2 is using the indexes. After this initial use of RUNSTATS, avoid the further use of the RUNSTATS utility. Over time with deletes of older objects and reuse of space for new objects, the object directory and object tables tend not to be in strict cluster sequence. It is not important that OAM object and object directory tables be in cluster sequence and regularly reorganized. OAM access to data is entirely though DB2 indexes. The initial "decision" by DB2 to use indexes when a table is created is maintained, and indexes are used for access, as long as RUNSTATS utility is not used when tables are not maintained in cluster index sequence. The use of RUNSTATS without reorganizing a table could result in DB2 discontinuing use of indexes.

The advantages described here are best used when segmented table spaces are used for objects and object directory entries. As stated, simple table spaces should not be used for OAM. There are circumstances when the INSERT performance differences between segmented and partitioned table spaces are not as important as minimizing the work load of DB2 maintenance activity. It is the decision of the installation whether to accept less possible performance and use partitioned table spaces based on their unique operating circumstances.

Partitioning object storage tables

Because of the large amount of data that can be stored in the OAM object storage tables (the 4 KB object storage table, the 32 KB object storage table and the LOB storage structure) associated with each Object storage group, you might choose to partition the DB2 table spaces containing each of these tables. The 4 KB object storage table, the 32 KB object storage table, and LOB storage structure are each stored in separate DB2 table spaces. You might partition the DB2 table space containing any or all of these tables.

For information about the advantages and disadvantages of partitioned table spaces, see the "DB2 administration" topic in IMS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPH2).

Partitioning the 4 KB object storage tables

During OAM installation and customization (using the default SAMPLIB members provided with the product), OAM creates a nonpartitioned unique clustered index on the 4 KB object storage table using a composite key consisting of both:

- Collection ID column (OTCLID) in ascending order
- Object name column (OTNAME) in ascending order

You might change the DB2 table space containing the 4 KB object storage table into a partitioned table space. If you do so, decide which column or columns on the 4 KB object storage table to use for the partitioning key. The following two examples of columns in the 4 KB object storage table might be used for the partitioning key:

- Collection ID column (OTCLID)
- Object name column (OTNAME)

OAM does not update the value of the collection ID column (OTCLID) in a row in the 4 KB object storage table, so this column might be used in a partitioning key. OAM does not update the value of the object name column (OTNAME) in a row in the 4 KB object storage table, so this column might be used in a partitioning key. If you use the object name column (OTNAME) as the partitioning key, remember that DB2 only uses the first 40 bytes of the partitioning key to actually partition the data.

Another option to partition the 4 KB table is by adding a column that you use only as the partitioning key.

1. Determine the data type of the column and the value to be inserted into the column.

   ____

2. Alter the 32 KB table to add this column before the OTOBJ column.

   - For example, one option could be to make this column a timestamp that is set when a row is inserted into the table when the object is created. You would then partition the table based on the selected time periods.
   - Another option could be to make the partitioning column a partition ID. You could specify a DB2 trigger that calculates the value of the partition ID.

   ____

3. If you add a new column to the 4 KB table, you must drop and recreate the views that were created at installation. To drop and recreate the views, run sample job CBRISQL0, as the following example shows:

   ```
   DROP VIEW osg_hlq.V_OSM_04K_OBJ_TBL;
   CREATE VIEW osg_hlq.V_OSM_04K_OBJ_TBL AS SELECT ALL * FROM
   osg_hlq.OSM_04K_OBJ_TBL;
   ```

   The *osg_hlq* in the DROP and CREATE statements is the high-level qualifier for the object storage group for the 4 KB table.

   ____

To create the DB2 table space containing the 4 KB object storage table as a partitioned table space, you must modify the CREATE TABLESPACE SQL statements by adding a NUMPARTS clause for the HLQ.OSMOTS04 table spaces in the CBRISQL0 job in SYS1.SAMPLIB.

If you create the DB2 table space containing the 4 KB object storage table as a partitioned table space, you must define a partitioned index on the 4 KB object storage table. The partitioned index can be created by adding a CREATE INDEX SQL statement to the CBRISQL0 sample job in SYS1.SAMPLIB.

If you create the DB2 table space containing the 4 KB object storage table as a partitioned table space, the partitioned index must also be the clustering index. Therefore, the default index that OAM creates on the 4 KB object storage table (HLQ.OBJT04X1) cannot be a clustering index. In this case, you must change the default index that OAM creates on the 4 KB object storage table (HLQ.OBJT04X1) to a nonclustered index by removing the CLUSTER keyword from the CREATE INDEX SQL statement for the HLQ.OBJT04X1 index in the CBRISQL0 sample job in SYS1.SAMPLIB.

If you create the DB2 table space containing the 4 KB object storage table as a partitioned table space, there must still be a unique nonpartitioned index on the composite key in order for OAM to function properly that consists of both:

- Collection ID column (OTCLID) in ascending order
- Object name column (OTNAME) in ascending order

Note: Having a partitioned index and a nonpartitioned index on the 4 KB object storage table might diminish some of the benefits of partitioning the 4 KB object storage table.

In addition to changing the SQL statements contained in the CBRISQL0 sample jobs in SYS1.SAMPLIB, also update the CBRIALC0 job in SYS1.SAMPLIB. Include IDCAMS (access method services) DEFINE CLUSTER commands to preallocate a VSAM linear data set (LDS) for each of the partitions that you plan on having for each partitioned table space containing the 4 KB object storage table associated with each Object storage group. Also use the IDCAMS DEFINE CLUSTER command to preallocate a VSAM linear data set for each partition comprising the partitioned index that you plan to create. The data set names associated with each VSAM linear data set must conform to DB2 data set naming conventions as specified in the "Data set naming conventions" topic in IMS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPH2).

DB2 free space search algorithms are not as efficient for partitioned table spaces as they are for segmented table spaces. As a result of partitioning the DB2 table spaces that contain the OAM 4 KB object storage tables, you might impact the performance when small objects are being stored to the DB2 sublevel.

By partitioning the DB2 table space containing the OAM 4 KB object storage table, you are accepting the following responsibilities:

- That OSREQ STORE performance, when storing small objects to the DB2 sublevel, might not be as fast as when using a segmented table space for the OAM 4 KB object storage table.
- That OSMC transition of small objects to the DB2 sublevel might not be as fast as when using a segmented table space for the OAM 4 KB object storage table.
- To create and manage each of the underlying VSAM linear data sets associated with each partition of the partitioned table space.
- To create and manage each of the underlying VSAM linear data sets associated with each partition of the partitioned index.

Partitioning the 32 KB object storage tables

During OAM installation and customization (using the default SAMPLIB members provided with the product), OAM creates a nonpartitioned unique clustered index on the 32 KB object storage table using a composite key consisting of both:

- Collection ID column (OTCLID) in ascending order
- Object name column (OTNAME) in ascending order

You might change the DB2 table space containing the 32 KB object storage table into a partitioned table space. If you do so, you need to decide what column or columns on the 32 KB object storage table to use for the partitioning key. The following two examples of columns in the 32 KB object storage table might be used for the partitioning key:

- Collection ID column (OTCLID) or
- Object name column (OTNAME)

Because OAM does not update the value of the collection ID column (OTCLID) in a row in the 32 KB object storage table, this column might be used in a partitioning key. OAM does not update the value of the object name column (OTNAME) in a row in the 32 KB object storage table, so this column might be used in a partitioning key. If you use the object name column (OTNAME) as the partitioning key, remember that DB2 only uses the first 40 bytes of the partitioning key to actually partition the data.

Another option to partition the 32 KB table is by adding a column that you use only as the partitioning key.

1. Determine the data type of the column and the value to be inserted into the column.

   ___

2. Alter the 4 KB table to add this column before the OTOBJ column.

   - For example, one option could be to make this column a timestamp that is set when a row is inserted into the table when the object is created. You would then partition the table based on the selected time periods.
   - Another option could be to make the partitioning column a partition ID. You could specify a DB2 trigger that calculates the value of the partition ID.

   ___

3. If you add a new column to the 32 KB table, you must drop and recreate the views that were created at installation. To drop and recreate the views, run sample job CBRISQL0, as the following example shows:

   ```
   DROP VIEW osg_hlq.V_OSM_32K_OBJ_TBL;
   CREATE VIEW osg_hlq.V_OSM_32K_OBJ_TBL AS SELECT ALL * FROM
   osg_hlq.OSM_32K_OBJ_TBL;
   ```

   The *osg_hlq* in the DROP and CREATE statements is the high-level qualifier for the object storage group for the 32 KB table.

   ___

To create the DB2 table space containing the 32 KB object storage table as partitioned table space you must modify the CREATE TABLESPACE SQL statements by adding a NUMPARTS clause for the HLQ.OSMOTS32 table spaces in the CBRISQL0 job in SYS1.SAMPLIB.

If you create the DB2 table space containing the 32 KB object storage table as a partitioned table space, you must define a partitioned index on the 32 KB object storage table. The partitioned index can be created by adding a CREATE INDEX SQL statement to the CBRISQL0 sample job in SYS1.SAMPLIB.

If you create the DB2 table space containing the 32 KB object storage table as a partitioned table space, the partitioned index must also be the clustering index. Therefore, the default index that OAM creates on the 32 KB object storage table (HLQ.OBJT32X1) cannot be a clustering index. In this case, you must change the default index that OAM creates on the 32 KB object storage table (HLQ.OBJT32X1) to a nonclustered index by removing the CLUSTER keyword from the CREATE INDEX SQL statement for the HLQ.OBJT32X1 index in the CBRISQL0 sample job in SYS1.SAMPLIB.

If you create the DB2 table space containing the 32 KB object storage table as a partitioned table space, there must still be a unique nonpartitioned index on the composite key in order for OAM to function properly that consists of the following:

- Collection ID column (OTCLID) in ascending order
- Object name column (OTNAME) in ascending order
- Object segment (OTSEG) in ascending order

Note: Having a partitioned index and a nonpartitioned index on the 32 KB object storage table might diminish some of the benefits of partitioning the 32 KB object storage table.

In addition to changing the SQL statements contained in the CBRISQL0 sample job in SYS1.SAMPLIB, also update the CBRIALC0 job in SYS1.SAMPLIB. Include IDCAMS DEFINE CLUSTER commands to preallocate a VSAM linear data set (LDS) for each of the partitions that you plan on having for each 32 KB object storage table associated with each Object storage group. Also use the DEFINE CLUSTER command to preallocate a VSAM linear data set for each partition comprising the partitioned index that you plan to create. The data set names associated with each VSAM linear data set must conform to DB2 data set naming conventions as specified in the "Data set naming conventions" topic in IMS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPH2).

DB2 free space search algorithms are not as efficient for partitioned table spaces as they are for segmented table spaces. As a result of partitioning the DB2 table spaces that contain the OAM 32 KB object storage tables, you might impact the performance when large objects are being stored to the DB2 sublevel.

By partitioning the DB2 table space containing the OAM 32 KB object storage table, you are accepting the following responsibilities:

- That OSREQ STORE performance when storing large objects to the DB2 sublevel might not be as fast as when using a segmented table space for the OAM 32 KB object storage table
- That OSMC transition of large objects to the DB2 sublevel might not be as fast as when using a segmented table space for the OAM 32 KB object storage table
- To create and manage each of the underlying VSAM linear data sets associated with each partition of the partitioned table space
- To create and manage each of the underlying VSAM linear data sets associated with each partition of the partitioned index

Partitioning the LOB storage structures

During OAM installation and customization (using the default SAMPLIB members provided with the product), OAM creates a nonpartitioned unique clustered index on the LOB storage structure using a composite key consisting of both:

- Collection ID column (OTCLID) in ascending order
- Object name column (OTNAME) in ascending order

You might change the DB2 table space containing the LOB storage structure into a partitioned table space. If you do so, decide which column or columns on the LOB storage structure to use for the partitioning key. The following two examples of columns in the LOB storage structure might be used for the partitioning key:

- Collection ID column (OTCLID)
- Object name column (OTNAME)

OAM does not update the value of the collection ID column (OTCLID) in a row in the LOB storage structure, so this column might be used in a partitioning key.

OAM does not update the value of the object name column (OTNAME) in a row in the LOB storage structure, so this column might be used in a partitioning key. If you use the object name column (OTNAME) as the partitioning key, remember that DB2 only uses the first 40 bytes of the partitioning key to actually partition the data.

Another option to partition the LOB storage structure is by adding a column that you use only as the partitioning key.

1. Determine the data type of the column and the value to be inserted into the column
2. Alter the LOB storage structure to add this column before the OTOBJ column.
   - For example, one option could be to make this column a timestamp that is set when a row is inserted into the table when the object is created. You would then partition the table based on the selected time periods.
   - Another option could be to make the partitioning column a partition ID. You could specify a DB2 trigger that calculates the value of the partition ID.
3. If you add a new column to the LOB storage structure, you must drop and recreate the views that were created at installation. To drop and recreate the views, run excerpt from sample job CBRILOB, as the following example shows:

```
DROP VIEW osg_hlq.V_OSM_LOB_BASE_TBL;
CREATE VIEW osg_hlq.V_OSM_LOB_BASE_TBL AS SELECT ALL * FROM
osg_hlq.OSM_LOB_BASE_TBL;
```

The *osg_hlq* in the DROP and CREATE statements is the high-level qualifier for the object storage group for the LOB storage structure.

To create the DB2 table space containing the LOB storage structure as a partitioned table space, you must modify the CREATE TABLESPACE SQL statements by adding a NUMPARTS clause for the HLQ.OSMLBTS and table spaces in the CBRILOB job in SYS1.SAMPLIB.

If you create the DB2 table space containing the LOB storage structure as a partitioned table space, you must define a partitioned index on the LOB storage structure. The partitioned index can be created by adding a CREATE INDEX SQL statement to the CBRILOB sample job in SYS1.SAMPLIB.

If you create the DB2 table space containing the LOB storage structure as a partitioned table space, the partitioned index must also be the clustering index. Therefore, the default index that OAM creates on the LOB storage structure (HLQ.OBJT04X1) cannot be a clustering index. In this case, you must change the default index that OAM creates on the LOB storage structure (HLQ.OTLOBX1) to a nonclustered index by removing the CLUSTER keyword from the CREATE INDEX SQL statement for the HLQ. OTLOBX1 index in the CBRILOB sample job in SYS1.SAMPLIB.

If you create the DB2 table space containing the LOB storage structure as a partitioned table space, there must still be a unique nonpartitioned index on the composite key in order for OAM to function properly that consists of both:

- Collection ID column (OTCLID) in ascending order
- Object name column (OTNAME) in ascending order

Note: Having a partitioned index and a nonpartitioned index on the LOB storage structure might diminish some of the benefits of partitioning the LOB storage structure.

In addition to changing the SQL statements contained in the CBRILOB sample job in SYS1.SAMPLIB, also include IDCAMS (access method services) DEFINE CLUSTER commands to preallocate a VSAM linear data set (LDS) for each of the partitions that you plan on having for each partitioned table space containing the LOB storage structure associated with each object storage group. Also use the IDCAMS DEFINE CLUSTER command to preallocate a VSAM linear data set for each partition comprising the partitioned index that you plan to create. The data set names associated with each VSAM linear data set must conform to DB2 data set naming conventions as specified in the "Data set naming conventions" topic in IMS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPH2).

DB2 free space search algorithms are not as efficient for partitioned table spaces as they are for segmented table spaces. As a result of partitioning the DB2 table spaces that contain the OAM LOB storage structures, you might impact the performance when small objects are being stored to the DB2 sublevel.

By partitioning the DB2 table space containing the OAM LOB storage structure, you are accepting the following responsibilities:

- OSREQ STORE performance, when storing small objects to the DB2 sublevel, might not be as fast as when using a segmented table space for the OAM LOB storage structure.
- OSMC transition of small objects to the DB2 sublevel might not be as fast as when using a segmented table space for the OAM LOB storage structure.
- To create and manage each of the underlying VSAM linear data sets associated with each partition of the partitioned table space.
- To create and manage each of the underlying VSAM linear data sets associated with each partition of the partitioned index.

Tuning object retrieval response time

The OAM response time for retrieval of objects might be a key factor in the performance of your application, especially if the application is intended for interactive use. The minimum time to retrieve an object from an unmounted cartridge in an optical library is typically 15 - 30 seconds. These times increase when the resource is attached to a different OAM in an OAMplex and cross-system processing is required.

Understand the retrieval response time requirements for your applications and monitor the actual response time achieved.

The key to providing the required response time is to assign objects to storage classes that have an adequate INITIAL ACCESS RESPONSE SECONDS value that is defined and to management classes that do not cause a transition to a slower storage class until the frequency of retrieving the objects is reasonably low. The primary attributes of a management class that can be used to control transition are TIME SINCE CREATION and TIME SINCE LAST USE.

Restriction: Do not use TIME SINCE LAST USE if the UPD=N option is used on the OAM*n*,CBRINIT statement in IEFSSN*xx* PARMLIB member.

Use the OSREQ QUERY function to obtain the estimated retrieval response time for an object. The OSREQ QUERY function also provides descriptive information concerning objects residing on the following storage media:

- DB2 sublevel storage on a direct access storage device (DASD)
- An optical disk volume inside of an optical library
- An optical disk volume on the shelf
- A tape volume inside an IBM Automated Tape Library Dataserver
- A shelf-resident tape volume (a tape volume that resides outside an IBM Automated Tape Library Dataserver)
- File system sublevel storage, which could be a zFS file system on DASD or an NFS file system on a wide variety of internal storage within an NFS server.

OAM returns this descriptive information, along with the primary, backup, and secondary backup retrieval order keys, in the Query Element List. QUERY searches the object directory for a match on the specific name in the NAME keyword and returns a single query element. You can perform a generic search for each object whose name matches the partially qualified name that is specified in the NAME keyword. The search returns a query element for each object found. The output of a QUERY might be used as the input to an OSREQ RETRIEVE request.

Additionally, for library-resident objects whose retrieval can be predicted in advance, you might want to fetch the objects before they are needed for interactive use. This can be done by performing the following activities:

- Using the OSREQ macro to change objects' storage classes to storage classes having a nonzero value for INITIAL ACCESS RESPONSE SECONDS (stored on removable media—tape or optical)
- Running the storage management cycle to move the objects to the proper level of the object storage hierarchy

Tip: You can use the PERIODIC attribute of the management class to prefetch the objects to DASD to improve performance on subsequent retrievals. For example, on the first day of each month, you can assign the objects to a management class that moves the objects to a faster storage class just before you use the objects. See "PERIODIC" on page 177 for a discussion of the attributes for this parameter.

Related reading:

- For more information concerning the Query Element List, see *z/OS DFSMS OAM Application Programmer's Reference*.
- For a discussion of how drive availability between libraries can affect object retrieval response time, see "Balancing library usage" on page 212.

Recalling objects to disk

When objects residing on optical or tape media are retrieved, they can be recalled to the DB2 sublevel or file system sublevel for up to 255 days beyond the day of retrieval, thereby providing improved performance for subsequent retrieves of those objects. If you anticipate increased demand for an object, you can specify, through OSREQ or SETOSMC keywords, how long to keep it on the DB2 sublevel or file system sublevel when retrieving it. After the specified number of days have elapsed the objects are restored back to their original location as part of the OSMC storage management cycle.

For each OSREQ RETRIEVE request for an object residing on optical or tape media, OAM determines whether or not a recall to the DB2 sublevel or file system sublevel is required, either explicitly through the RECALL keyword on the OSREQ RETRIEVE, or implicitly through SETOSMC keywords in the CBROAM*xx* PARMLIB member. If a recall is specified, OAM initiates a recall request to the OSMC component to write a full copy of the object to the DB2 sublevel or file system sublevel at the same time the OSR component of OAM services the object retrieval. The location field for an object will be updated to "R" (if recalled to the DB2 sublevel) or to "2" (if recalled to the file system sublevel). The objects pending action date will be set to the current date + the number of days the object is to be recalled. Because these OSMC recall tasks can consume considerable resources, you may wish to use the MAXRECALLTASKS keyword on the SETOSMC statement to limit the number of recalls that can run simultaneously.

Automatic class selection (ACS) routines are not invoked when the recalled object is copied to the DB2 sublevel or file system sublevel. ACS may, however, be invoked when the object is restored to its original location and the DB2 or file system copy is deleted after the specified number of days. After the specified number of days, the recalled object is subsequently processed by OSMC, which deletes the disk copy of the object and updates the object's location and pending action date based on the object's current management class. If the calculated pending action date is the current date or earlier, object processing will occur at this time for that object, including object expiration, the invocation of the ACS routines for transitioning the object, or both.

Recalls are processed asynchronously with the actual RETRIEVE request so that the RETRIEVE is not delayed while recall processing takes place. For each object that is recalled, an OSMC task is started in the background. This can result in subsequent RETRIEVES for an object with recall processing still pending. In most cases the cartridge will still be mounted on the drive, reducing any delays on subsequent retrievals. To increase the chances of the volume still being mounted, use the TAPEDISPATCHERDELAY function described in "SETOAM keyword definitions for global level" on page 112 and the and OPTICALDISPATCHERDELAY function described in "SETOPT keyword definitions" on page 132.

See "Updating SETOSMC values" on page 371 and "Updating SETOAM values" on page 367 for information on specifying recall to the DB2 sublevel or file system sublevel.

Refer to *z/OS DFSMS OAM Application Programmer's Reference* for information on using the optional RECALL keyword on an OSREQ RETRIEVE request to initiate an explicit recall.

Note: Regardless of the setting for UPD=, the ODLREFDT is not updated when:

- OSREQ RETRIEVE results in RECALL being scheduled, or 2.
- OSREQ RETRIEVE of object currently in RECALL mode.

Tuning the storage management cycle

The storage management cycle moves objects between disk, optical, and tape media, writes backup copies of objects, deletes expired objects, and expires tape volumes and optical cartridges when all the expiration criteria have been set. It should be run when the application work load is at a minimum.

You can obtain the shortest storage management cycle by making the best use of the library drives or operator-accessible drives in your SMS configuration. Your intent should be to process as many Object or Object Backup storage groups concurrently and use as many drives concurrently as possible without introducing contention for drives by different storage groups (which causes unnecessary mounts and demounts of cartridges).

OAM provides the following controls for tuning the storage management cycle:

- The DRIVE STARTUP threshold attribute of each Object storage group definition and each Object Backup storage group definition determines the number of optical drives that are used concurrently for that storage group.
- The TAPEDRIVESTARTUP threshold optional subparameter of the STORAGEGROUP parameter on the SETOAM statement, determines when OAM is to start additional tape drives for writing object data to

tape volumes that belong to the Object and Object Backup storage groups. Consider the MAXTAPESTORETASKS keyword (for the OAM global level) and SGMAXTAPESTORETASKS keyword (for the storage group level) with TAPEDRIVESTARTUP threshold for further controls within the storage management cycle.

Restriction: Do not specify a number greater than the number of tape drives available to OAM for the combined MAXTAPESTORETASKS, MAXTAPERETRIEVETASKS, SGMAXTAPESTORETASKS, and SGMAXTAPERETRIEVETASKS subparameters. This specification can cause a system to go into allocation recovery and attempt to allocate tape drives after all tape drives are in use, causing system problems.

- The CYCLE START TIME and CYCLE END TIME attributes of each Object or Object Backup storage group control the window in which the storage management cycle begins processing and (optionally, stops processing) the storage group.
- The MAXS parameter of the OAM cataloged procedure controls the number of storage groups that the storage management cycle processes concurrently.
- The UPD=N option on the OAM*n* statement in IEFSSN*xx* member of PARMLIB reduces unnecessary retrieval and update of objects' directory entries during the OSMC cycle if your installation's management classes do not use the TIME SINCE LAST USE or EXPIRE AFTER DAYS USAGE parameters.
- In an OAMplex, the OSMC processing system name for each Object or Object Backup storage group controls where OSMC processing is done for that storage group. Using this parameter and separating hardware between storage groups can balance workload across systems for OSMC processing. Globalize hardware and highest usage to reduce XCF usage.

If your processing includes making object backups on operator-accessible optical disk drives, run as many Object storage groups concurrently as there are usable operator-accessible optical drives. If backup copies are not being made, you might be able to run as many Object storage groups concurrently as there are usable library drives in the configuration. Remember, in determining the number of Object storage groups to run concurrently, there are DB2 limitations which must be taken into account. See "Tuning OAM connections to DB2" on page 201 for additional information on these DB2 limitations.

Remember, OSMC functions other than the storage management cycle you start (for example, Volume Recovery utility, Move Volume utility, and others) are consumers of resources as well and need to be considered in your usage of the MAXS and DRIVE STARTUP threshold controls.

To avoid contention for drives within any one library, analyze the windows for processing each Object storage group. For example, if volumes for eight storage groups all exist in the same single optical library with four drives and MAXS=10, then the windows for the groups should be set so that no more than four overlaps at any given time.

Recommendation: Consider all OAMs within an OAMplex when making decisions that affect those resources.

Attention: If you plan on using the CBRHADUX installation exit that is shipped with the SAMPLIB or plan on editing or creating your own CBRHADUX installation exit that does not allow expiration of objects, using this exit can cause OSMC performance problems if you have not properly established your expiration criteria in your SMS management classes.

If you do not plan on expiring objects and have established your CBRHADUX installation exit to return with an indication that no expiration is allowed, you must make sure that your SMS management class has expiration criteria that does not cause OSMC to continually pick objects to be expired. Always be sure that your management class sets the expiration criteria to NEVER expire if you do not plan to expire OAM objects.

Considerations for larger data objects

If larger objects are spread across multiple storage groups, you can avoid virtual storage shortages in OAM address space when running your OSMC storage management cycle by limiting the number of storage groups that can be run concurrently. For example, if the maximum object size (specified in the IEFSSN*xx* with the MOS=*nnn* keyword) is 100MB, processing a maximum of 10 object storage groups at a time is recommended. For a maximum object size of 256MB, a maximum of four storage groups for concurrent processing is recommended. The maximum number of storage groups processed can be set using MAXS= in the PARM field on the JCL EXEC statement in the OAM cataloged procedure. The default value is MAXS=2.

You should also determine the maximum object size of any other storage groups you have and adjust the MAXS= parameter accordingly. If, for example, you have only five storage groups with the potential to contain 100MB objects, you could specify more than five additional storage groups.

In general, OSMC will process concurrently up to 15MB of data per storage group with the possibility of processing 15MB plus the largest object size in the storage group. When processing multiple storage groups concurrently, ensure that the total amount of storage across all storage groups stays around 1GB. This will allow the remaining 1GB to be used for other OAM address space functions. The use of a monitoring tool during the initial OSMC storage cycles can help you avoid problems resulting from virtual storage constraints.

OAM exploits 64–bit virtual storage when objects greater than 256MB are written to or read from the file system sublevel or tape. If your installation will implement objects greater than 256MB on the file system sublevel or tape, you must ensure that OAM is configured properly to access virtual storage above the 2G bar and you must plan for the significantly increased storage utilization on your system. For more information see "MEMLIMIT for OAM" on page 35, "Auxiliary storage and real storage considerations" on page 69, and "System paging" on page 72.

Note: Objects greater than 256MB are supported on the DB2 sublevel, file system sublevel, tape sublevel 1, and tape sublevel 2 in the OAM storage hierarchy. OSMC will not attempt to transition or write backup copies of objects greater than 256MB to the optical level. All other object processing for these objects continue.

Balancing library usage

OAM attempts to balance drive use within a given library; however, in a system with multiple libraries, the level of activity can vary greatly among libraries. Activity can be affected by the assignment of Object or Object Backup storage groups to libraries, the frequency of retrieval of objects from optical or tape cartridges, and the availability of scratch cartridges.

If one library has much more activity than the others, the response time for work on that library can be lengthened. You should monitor the number of cartridge mounts in each library by examining the console log.

To balance the work load, reassign Object or Object Backup storage groups to other libraries, and eject and move the cartridges to the corresponding libraries. Also, for an Object storage group that is defined to be resident in multiple libraries, cartridges in the Object storage group can be ejected from the overly active library and entered into another library assigned to the Object storage group.

When determining which volume to use for a write request, OAM attempts to find the volume that is available to the storage group and that has the least amount of free space, but enough space for the object that is to be written. If a new library is added to a configuration and the new library is added to an existing Object or Object Backup storage group definition, it is a good idea to move some of the partially used volumes that belong to the Object or Object Backup storage group instead of populating a new library only with scratch or unused grouped volumes. This will help to distribute the read and write workload across the libraries.

Using appropriate transport classes within XCF

In an OAM Parallel Sysplex environment, cross-coupling facility (XCF) message services send requests and data between instances of OAM within an OAMplex. XCF messaging services, a function within XCF, uses transport classes to send XCF messages and data through the coupling facility between the various systems within a sysplex environment.

Transport classes are used by XCF messages service to group messages that are to be sent between systems within a Parallel Sysplex. Messages are assigned to transport classes based on the group name (defining specific transport classes to OAM, for example), the message size, or both. Each transport class has its own resources that consist of buffers and one or more inbound and outbound paths. In most cases, it is more efficient to pool the resources and have the transport classes based on message size, rather than on group name.

XCF message buffers are managed by correctly selecting the size of the messages most frequently sent from specific buffer pools and by specifying an adequate upper limit for the size of the buffer pool. Multiple default transport classes of various sizes are assigned to the multiple buffers in the coupling facility. XCF determines which transport class is to be used depending on the size of the message or object in the buffer being transported.

XCF attempts to optimize the use of transport classes by selecting a class that is large enough to handle all the data being transported. For example, if a message or data is being sent from one instance of OAM to another OAM system within an OAMplex through the coupling facility using XCF message services that is 5 KB in size, XCF might use the 5 KB default transport class to send the message. XCF tries to select the best fit transport class for the buffer size. However, XCF might also choose the 5 KB transport class to handle a 2 KB message if that is the best fit available at the time. The 5 KB default transport would be large enough to handle the request; however, the buffer is not being used efficiently.

It is possible to create customized transport classes based on message size (or object size) to use specifically for your own data by defining transport classes based on message or object size and assigning them based on the OAM XCF group name along with the default transport classes. Defining your own transport classes allows you to determine the best fit for your objects to optimize the use of the transport class for your group class buffer size.

Recommendation: Customizing your own transport classes works best if your installation has standard object sizes. Additionally, you would want to create smaller transport classes for OAM to handle the smaller messages used to communicate configuration updates. In this case, you would have the best fit transport classes for your data and messages.

Perform the following steps to determine if there is a legitimate need for user-defined transport classes:

1. Use the XCF default transport classes assigned to the buffers first to determine if they sufficiently accommodate the size of your data and are being used in an efficient manner.
2. Modify your configuration (storage group, library, and drive definitions) to best utilize your resources and reduce unnecessary XCF messaging for processing transactions. Some libraries span several storage groups, which might increase the need to send messages using XCF. Update your configuration to minimize the frequency and the amount of data that needs to be transported. For OSMC processing, try to run OSMC on the OAM that is managing and controlling the hardware associated with both the Object storage group being processed and the Object Backup storage group. If different storage groups are being processed on multiple OAMs concurrently and backup copies of objects need to be written, you should have hardware available to the Object Backup storage groups on each OAM that is doing the processing.

Recommendation: If your average object size is larger, consider using a CTC direct connection for transporting XCF messages and data to improve performance.

If you determine, after you have performed the previous steps, that there is sufficient need to define specific transport classes to the OAMplex to optimize the use of system resources, defined transport classes can be used along with the XCF defaults.

Related reading:

- For more information about an OAMplex, see "OAMplex" on page 8.
- For more information about Parallel Sysplexes, see Parallel Sysplex Overview (www.ibm.com/systems/z/advantages/pso/sysover.html).
- For information on how to calculate message buffer space, see *z/OS MVS Setting Up a Sysplex*.

Measuring OAM transaction performance using SMF

OAM uses system management facility (SMF) recording for gathering OAM statistical information to allow customers to measure the performance of OAM at the application programming interface level (the OSREQ macro interface).

The OAM SMF record allows the collection of statistical information about OAM usage for planning and diagnosis purposes such as:

- Information system usage accounting and charge back to user departments
- Performance analysis and monitoring to make certain that their information systems are as finely tuned as possible
- Capacity planning to determine when to procure additional hardware resources, such as storage devices (DASD, tape, optical) and media
- Potential problem determination data

The OAM SMF record subtypes are assigned to almost all OAM activities. The MVS system operator or system programmer can dynamically select the OAM SMF record subtypes to be recorded. The following activities have associated subtype records:

- Invocations of OSR functions:
  - ACCESS
  - CHANGE
  - DELETE
  - QUERY
  - RETRIEVE
  - STORE
  - STOREBEG
  - STOREPRT
  - STOREEND
  - UNACCESS
- Invocations of OSMC storage management activities:
  - Storage group processing
  - DASD space management processing
  - Volume recovery utility
  - Single object recovery utility
  - Library space management
  - Move volume utility
  - Single object recall utility
  - Immediate Backup Copy
  - Tape Recycle
- Library control system (LCS) optical library activities
  - Optical library varies online
  - Optical library varies offline
  - Optical drive varies online
  - Optical drive varies offline
  - Optical cartridge entry
  - Optical cartridge eject
  - Optical cartridge label
  - Optical cartridge audit
  - Optical cartridge mount
  - Optical cartridge demount
  - Optical write

- Optical read
- Optical logical delete
- Optical physical delete
- LCS object tape activities:
  - Object tape write
  - Object tape read
  - Object tape demount
  - Object tape logical delete
- LCS file system sublevel activities:
  - File system write
  - File system read
  - File system physical delete (delete)
  - File system physical delete (uncommitted store)

Related reading:

- For more information regarding the OSREQ macro, see *z/OS DFSMS OAM Application Programmer's Reference*.
- For more information about SMF, see Appendix E, "OAM System Management Facility records," on page 515, and *z/OS MVS System Management Facilities (SMF)*.

OAM SMF record subtypes

OAM records SMF records in the SMF data sets to account for OAM activity. The OAM SMF record is a type 85 (X'55'). The OAM SMF record begins with a standard 48-byte SMF record header. Each OAM SMF record contains three sections:

- Standard 48-byte SMF record header
- 112-byte OAM product section
- Variable length OAM data section

Table 32 on page 215 lists the OAM SMF record subtypes.

Table 32. Record subtypes and descriptions

Record subtype	Record size	Description
1	372	OSREQ Access
2	372	OSREQ Store
3	372	OSREQ Retrieve
4	372	OSREQ Query
5	372	OSREQ Change
6	372	OSREQ Delete
7	372	OSREQ Unaccess
8	372	OSREQ STOREBEG
9	372	OSREQ STOREPRT
10	372	OSREQ STOREEND
32	664	OSMC Storage Group Processing

Table 32. Record subtypes and descriptions (continued)		
Record subtype	**Record size**	**Description**
33	664	OSMC DASD Space Management
34	664	OSMC Volume Recovery Utility
35	664	OSMC Move Volume Utility
36	298	OSMC Single Object Recovery Utility
37	184	OSMC Library Space Management
38	288	OSMC Single Object Recall Utility
39	296	OSMC Immediate Backup
40	428	OSMC Tape Recycle
64	256	LCS Optical Drive Vary Online
65	256	LCS Optical Drive Vary Offline
66	256	LCS Optical Library Vary Online
67	256	LCS Optical Library Vary Offline
68	284	LCS Optical Cartridge Entry
69	284	LCS Optical Cartridge Eject
70	284	LCS Optical Cartridge Label
71	284	LCS Optical Volume Audit
72	284	LCS Optical Volume Mount
73	284	LCS Optical Volume Demount
74	variable (min = 416, max = 32 664)	LCS Optical Write Request
75	416	LCS Optical Read Request
76	416	LCS Optical Logical Delete Request
77	variable (min = 416, max = 32 664)	LCS Optical Physical Delete Request
78	variable (min = 416, max = 32 664)	LCS Object Tape Write Request
79	416	LCS Object Tape Read Request
87	228	LCS Object Tape Volume Demount (OAM usage)
88	416	LCS Object Tape Logical Delete Request
90	324	LCS File System Write Request
91	324	LCS File System Read Request
92	324	LCS File System Physical Delete Request
93	324	LCS File System Physical Delete Request (Uncommitted Application Store Cleanup)

For more information about SMF records, see Appendix E, "OAM System Management Facility records," on page 515.

OAM SMF start and end time accuracy

Each OAM SMF record has a function start and end time in the common OAM product section. The start and end times are in fields R85PSTRT and R85PENDT respectively, and they are presented in z/Architecture® STORE CLOCK (STCK) instruction format.

The starting time of the OAM function is obtained as soon as possible so that the elapsed time of the function includes as much OAM processing time as possible.

The ending time of the OAM function is obtained as soon as possible to the end of the function so that the elapsed time of the function includes as much OAM processing time as possible. However, because the function end time (in field R85PENDT) is set and the elapsed time (in field R85PRESP) is calculated prior to passing the record to SMF for recording, the elapsed time does not include the time required to invoke SMF and the time required to copy the record into the SMF buffers.

Establishing recovery procedures

As part of your disaster recovery plan, establish and test procedures for recovering these entities:

- DB2 object storage databases and the OAM configuration database
- File system directories
- Single objects
- An entire optical or tape volume
- Automatic access to back up

Recovering DB2 databases

The recoverable structure of data in DB2 is the table space. To ensure recoverability, make an image copy when creating each table space in the OAM configuration database and all the table spaces in each of the object storage databases. Your installation determines how often to make backup copies, based on the usage of each table space. Use this original image copy as a base, and make subsequent periodic incremental image copies of each table space.

At specified intervals (best defined based on the usage of each table space), perform a MERGECOPY on the base (original, full-image copy) and subsequent incremental-image copies to establish a new base. After creating the new base level, perform subsequent incremental image copies in relation to this new base.

The main benefit of periodically using MERGECOPY to create a new base is the time savings at the time of the failure. Merge copies can be time-consuming, so it is best to do them on a timely, convenient basis.

To recover a table space, merge the contents of the DB2 recovery log with the most recent full-image copy of the table space. Because each change that is made to the database is recorded in the DB2 recovery log, the merge restores the table space to its last point of consistency before system failure.

Note:

1. In DB2, *point of consistency* is a term that designates a time when all recoverable data that is accessed by an application program is consistent with other data. It is also known as *syncpoint* or *commit point*.

For directions regarding how to make these image copies, see the topic "Plans for backup and recovery" in IMS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPH2).

Recovering file system directories

When the file system sublevel is used, the recoverable structure is the file system, so you should perform regular backups of the file system. Refer to the z/OS Unix publications for information on the various mechanisms for backing up file system data.

Ensure that the backup of the file system is coordinated with other OAM backup activities, including:

- DB2 object storage tables

- OAM configuration database
- any symbolic links that are used in the file system directory
- CBROAMxx PARMLIB member

Whenever OAM is running, the file system must remain mounted for use by OAM. OAM file system deletion activity can occur at any time that OAM is running. OAM must be stopped before a shut down of Unix System Services. These considerations must be taken into account for backup and recovery activities, as well as for continued proper operation of OAM.

Recovering single objects from removable media

OAM contains a single object recovery utility for recovering a single object from removable media. You can use either the first or the second backup copy for single object recovery as determined by the settings of the SETOSMC statements in the active CBROAM*xx* member of PARMLIB. The system creates a new primary copy from a backup copy (if one exists) using the following criteria:

- If the primary object resides on optical disk, OAM uses a backup copy (on either optical disk or tape) to create a new optical primary copy.
- If the primary object resides on tape, a backup copy (on either optical disk or tape) is used to create a new tape primary copy.
- If the primary object resides on the DB2 sublevel, OAM uses a backup copy (on either optical disk or tape) to create a new DB2 sublevel primary copy.
- If the primary object resides on the file system, OAM uses a backup copy (on either optical disk or tape) to create a new file system primary copy.

The operator starts the single object recovery utility to copy the object. For further information on this procedure, see "Starting object recovery for single objects" on page 276.

Recovering an entire optical cartridge or tape volume

OAM contains the Volume Recovery utility that recovers the objects from an unusable optical or tape volume to a usable volume. The Volume Recovery utility is used if an optical or tape volume is rendered unreadable, either because of physical damage, or because the volume cannot be found. The Volume Recovery utility is used for two types of volume recovery:

- Volumes containing primary objects belonging to an Object storage group can be recovered from the first or second backup copies of the objects (optical or tape).
- Backup volumes belonging to an Object Backup storage group can be recovered from the primary copies of the objects (disk (DB2 or file system), optical, or tape). All storage groups that contain objects that need to be recovered must be defined as part of the ACDS configuration.

To recover a primary optical or tape volume, all of the backup volumes containing either the first or the second backup copies of the objects on the primary volume are needed, whether they are optical or tape. The media from which the backup copy is retrieved for the recovery depends on the volume on which the selected backup copy resides.

If you specified the DELETE option, you can delete recovered tape and optical volumes after all data on those volumes has been recovered successfully.

Example: OAM might have run one storage management cycle for the storage group after OAM was initialized with a CBROAM*xx* PARMLIB member that contained a particular SETOAM statement. This SETOAM statement specified a tape unit name for an Object Backup storage group that caused OAM to write backups for that group to tape. Another time, OAM might have run a storage management cycle for the storage group after OAM was initialized, and the system invoked the START OAM command with one of these three options:

- Without a CBROAM*xx* PARMLIB member
- With a CBROAM*xx* PARMLIB member that contained no SETOAM statements

- With a SETOAM statement that did not specify a tape unit name that was associated with an Object Backup storage group

Any one of these options causes OAM to write backups to an optical volume.

When recovering a backup volume, every Object storage group must be searched for primary objects having backup copies residing on the backup volume being recovered. The primary copy for each of these objects can be on disk (DB2 or file system), optical, or tape. As a result, the Volume Recovery utility must identify the optical volumes as well as the tape volumes that are needed for recovery. If both optical and tape volume are requested for the recovery, the operator must reply that both types are available for the recovery to continue. The operator starts the recovery utility to copy the data.

The Volume Recovery operation is similar to that of Movevol, and, as such, it's performance can also be enhanced by employing the same techniques used to tune the Movevol environment. Refer to "Analyzing resources and tuning OAM for MOVEVOL usage" on page 225.

For more information on the Volume Recovery utility, see "Starting the OAM Volume Recovery utility" on page 270 and "Stopping a volume recovery that is in progress" on page 380.

Accessing backup objects automatically

OAM allows your application to automatically obtain the first or second backup copy of an object when the primary copy of the object is in one of the following situations:

- On removable media that is marked not readable (possibly damaged or destroyed)
- On removable media that is in a library that is offline or pending offline
- On removable media that is in a library that is not operational
- If a DB2 error is encountered attempting to retrieve the object data (4 KB or 32 KB table rows) and a backup copy of the object exists, OAM attempts to retrieve the backup copy of the object to satisfy the request.
- On removable media that is marked lost or is not-defined
- If a file system error is encountered attempting to retrieve the object data.

When you activate this function for one or more of the above conditions, and that condition exists when the OSREQ application interface retrieves an object, OAM attempts to obtain the first or second backup copy, if one exists. The backup copy that is selected for retrieval is determined by the BACKUP1 | BACKUP2 keyword that is specified on the MODIFY OAM,START,AB command. If this automatic access to backup objects function is inactive and the primary copy of the object is not available for any of the above reasons, the OSREQ API passes error return and reason codes back to the application. If no backup copy exists and the function is active, the OSREQ API passes error return and reason codes back to the application. Automatic access to backup does not automatically force all retrieves to be initiated from backup. Automatic access to backup copies of objects can be activated and deactivated by operator commands, or by using access backup keywords in a SETOPT statement in the CBROAMxx PARMLIB member.

For further information on accessing backup copies, see "Starting automatic access to backup copies of objects" on page 287 and "SETOPT Statement Syntax: OAM Global Level Parameters" on page 131.

Using the Move Volume utility

OAM provides a utility, the Move Volume utility (MOVEVOL), capable of moving objects from a primary or backup source volume (a tape volume or one side of an optical disk) to one or more target volumes, or deleting scratch tape volumes from the OAM tape inventory and specifying a scratch volume with the delete option specified. The set of eligible target volumes is determined by the drives eligible to write data into the Object or Object Backup storage group containing the source volume. The set of drives eligible to write data into the storage group containing the source volume is defined by the SMS storage group definition and the corresponding SETOAM statement if it exists.

If the source volume is an optical volume belonging to an Object storage group, the target volumes are optical volumes belonging to the same storage group. If no optical drives are available to write to the Object storage group to which the optical source volume belongs, the MOVEVOL request fails.

If the source volume is a tape volume belonging to an Object storage group, the target volumes are tape volumes belonging to the same storage group. If there is not a valid tape unit name associated with the Object storage group to which the tape source volume belongs, the MOVEVOL request fails.

If the source volume is either an optical or a tape volume belonging to the Object Backup storage group, the type of media used for writing the objects is derived from the definition of the Object Backup storage group. For example, if there is a valid SETOAM statement for the Object Backup storage group to which the source volume belongs, the target volumes chosen are tape volumes. If there is no valid SETOAM statement for the Object Backup storage group belonging to the source volume, the target volumes chosen are optical disk volumes.

The Move Volume utility performs the following functions:

- Writes objects using the optical drives defined in the Object storage group when moving objects off a primary optical source volume.
- Writes objects using the tape drives allocated to the Object storage group (using the tape unit name specified in the SETOAM statement for the Object storage group) when moving objects off a primary tape source volume.
- Writes objects using optical drives defined to, or tape drives allocated to the Object Backup storage group when moving objects off of a backup optical or tape source volume.
- Recycles optical and rewritable tape volumes after all the objects have been successfully moved off of the volumes, if you have specified the optional RECYCLE keyword. For a WORM tape volume, the MOVEVOL with RECYCLE command is rejected.
- Deletes tape and optical volumes from the OAM inventory after all the objects have been moved off of the volumes when you have specified the optional DELETE keyword.

The Move Volume utility enables you to migrate object data in the Object storage group from the following:

- One optical media type to another optical media type
- One tape media type to another tape media type

The Move Volume utility enables you to migrate object data in the Object Backup storage group from the following:

- One optical media type to another optical media type
- One tape media type to another tape media type
- Optical media to tape media
- Tape media to optical media

The Move Volume utility enables you to migrate objects from tape sublevel 1 source volumes to tape sublevel 1 target volumes, and objects from tape sublevel 2 source volumes to tape sublevel 2 target volumes.

Note: MOVEVOL cannot be used to move objects from tape sublevel 1 volumes to tape sublevel 2 volumes, or from tape sublevel 2 volumes to tape sublevel 1 volumes. For example, to change an object’s residence from a tape sublevel 1 volume to a tape sublevel 2 volume, change the object's storage class and start an OSMC cycle to affect the transition to another media type.

The intent of the utility is to facilitate migration from "older" technology media to "newer" technology media (for example, from 12-inch media to 5.25-inch media). However, because the set of drives eligible to write data is defined by the definition of the storage group, the following examples of data movement in Table 33 on page 221 are also valid.

Table 33. Examples of data movement with the Move Volume utility		
Scenario	**Move from**	**Move to**
Movement from one optical media type to another optical media type	single-density WORM single-density REWR double-density WORM double-density REWR quad-density WORM quad-density REWR 8*x*-density WORM 8*x*-density REWR	single-density WORM single-density REWR double-density WORM double-density REWR quad-density WORM quad-density REWR 8*x*-density WORM 8*x*-density REWR
Movement from one tape media type to another tape media type	Cartridge System Tape Enhanced Capacity High Performance Extended High Performance Enterprise Tape Cartridge	Cartridge System Tape Enhanced Capacity High Performance Extended High Performance Enterprise Tape Cartridge
Movement from any optical media type to a tape volume belonging to the Object Backup storage group	single-density WORM single-density REWR double-density WORM double-density REWR quad-density WORM quad-density REWR 8*x*-density WORM 8*x*-density REWR	Cartridge System Tape Enhanced Capacity High Performance Extended High Performance Enterprise Tape Cartridge
Movement from any tape media type to optical media type belonging to the Object Backup storage group	Cartridge System Tape Enhanced Capacity High Performance Extended High Performance Enterprise Tape Cartridge	single-density WORM single-density REWR double-density WORM double-density REWR quad-density WORM quad-density REWR 8*x*-density WORM 8*x*-density REWR
Note: 1. WORM = write-once-read-many 2. REWR = rewritable		

Once the data is successfully moved from the source volume, the data is no longer accessible from the original source volume.

Recommendation: If the intent is to migrate from one tape media type to another tape media type (for example, from IBM Cartridge System Tape to IBM Enhanced Capacity Cartridge System Tape), it might be necessary to modify tape volume records in the DB2 TAPEVOL table (see "Updating fields in the DB2 Volume Table and the Tape Volume Table" on page 374). If you change the TAPEUNITNAME on the SETOAM statement that assigns the storage group, OAM recognizes the change; however, OAM only uses it for new scratch allocations. If there are available usable tape volumes that belong to this storage group (the storage group to which the source volume belongs), OAM continues to use these volumes until they are all full. In order to force writes to go to a new media type (honoring the changed TAPEUNITNAME), any available usable volumes in that storage group must be either marked not writable or marked full with their percent full set to 100. This causes OAM to believe that the storage group is out of space and to request a scratch allocation using the new SETOAM TAPEUNITNAME specified for that group.

Preparation of the Move Volume utility environment

Before you invoke the Move Volume utility (also called MOVEVOL), perform the following tasks to prepare the environment in which the utility runs.

Subtask	See Procedure
"Preparing object and object backup storage groups" on page 222	"Preparing object and object backup storage groups" on page 222
"Updating the SETOPT OPTICALREINITMODE or SETOAM TAPERECYCLEMODE keyword" on page 222	"Updating the SETOPT OPTICALREINITMODE or SETOAM TAPERECYCLEMODE keyword" on page 222
"Updating and activating an SCDS" on page 222	"Updating and activating an SCDS" on page 222
"Minimizing or quiescing contending system activity" on page 224	"Minimizing or quiescing contending system activity" on page 224
"Analyzing resources and tuning OAM for MOVEVOL usage" on page 225	"Analyzing resources and tuning OAM for MOVEVOL usage" on page 225

Each of these tasks is explained in further detail below.

Preparing object and object backup storage groups

Update storage group definitions in ISMF and the CBROAM*xx* PARMLIB member to ensure the following desired results for the MOVEVOL command:

- A valid SETOAM statement exists for the Object storage group to which the source volume belongs if the source volume is a tape volume.
- An optical library and drives are defined to the Object storage group to which the source volume belongs if the source volume is an optical volume.
- A valid SETOAM statement exists for the Object Backup storage group if you plan to use the Move Volume utility to move objects from the Object Backup storage group to a tape volume.
- An optical library and drives are defined to the Object Backup storage group if you plan to use the Move Volume utility to move objects from the Object Backup storage group to an optical volume.

Updating the SETOPT OPTICALREINITMODE or SETOAM TAPERECYCLEMODE keyword

If you plan to use the Move Volume utility with the RECYCLE keyword, perform the following steps:

- For optical volumes, update the SETOPT OPTICALREINITMODE keyword in the CBROAM*xx* PARMLIB member.
- For object tape volumes, update the SETOAM TAPERECYCLEMODE keyword in the CBROAM*xx* PARMLIB member.

Updating and activating an SCDS

The Move Volume utility processes only objects that are associated with Object and Object Backup storage groups that are defined and enabled in the active SCDS. In your preparation for invocation of the Move Volume utility, update the SCDS, if necessary:

1. If the source volume is a backup volume from the Object Backup storage group, the SCDS must include *all* Object storage groups that contain objects that have a backup copy on the backup source volume. The Move Volume utility uses the list of Object storage groups defined in the SCDS to identify the objects to be moved.

   Failure to include all Object storage groups containing objects that have a backup copy on the backup source volume causes non-valid results. For example, although not all of the objects have been moved from the source volume, the utility might issue messages indicating that all objects are moved from the source volume because all of the objects identified in the SCDS have been moved. This situation occurs because the SCDS excludes some of the Object storage groups that contain objects having a backup copy on the backup source volume.

   ____

2. You might need to update the storage group that contains the source volume to define the set of drives eligible to write data into the storage group. Make sure that the storage group containing the source volume includes libraries capable of writing to the target optical media type. Use one of the following methods to update the storage group:

   a. Remove libraries from the storage group definition that contains volumes that you do not want to be used as target volumes for MOVEVOL.

   b. Add libraries that contain volumes that you do want to be used as target volumes for MOVEVOL.

   c. Leave the libraries with the undesired media assigned to the storage group, but for the duration of MOVEVOL, mark them as unwritable or full.

   d. If you use multifunction device types in the storage group definition, update the media type attribute in the library definition to direct writes to a specific media type.

   e. If you use multifunction device types in the storage group definition, update the library's read-only and write capable function based on the source and target media types.

   ____

3. Activate the updated SCDS.

   ____

For more information on updating and activating the SCDS, see *z/OS DFSMSdfp Storage Administration*.

The examples in Table 34 on page 223 show how to migrate data from an older media type to a newer media type using various methods.

Table 34. How to migrate from old media types to new media types

Method	**Scenario**	**Desired Result**	**Tasks**
Removing libraries from storage group definitions.	You have libraries with drives capable of writing to old media types defined to a storage group.	Ensure that the target volumes for the MOVEVOL request are new media types.	• Remove any libraries with drives capable of writing to old media types from the storage group definition. • Add libraries with drives capable of writing to new media types to the storage group definition, if some libraries are currently not defined.
Migrating without modifying the SCDS.	VOL1 is an old media type and belongs to SG1. Libraries containing drives capable of writing to both old media types and new media types exist in the SCDS.	Ensure that the target volumes for the MOVEVOL request for VOL1 are new media types.	• Mark full or unwritable any scratch volume in libraries associated with SG1 that are capable of writing to old media types. • Mark full or unwritable any volume belonging to SG1 in libraries that are capable of writing to old media types.

Table 34. How to migrate from old media types to new media types (continued)			
Method	**Scenario**	**Desired Result**	**Tasks**
Using the library default media type	All libraries associated with the source volume's storage group are multifunction 3995 optical libraries that are capable of writing to both old media types and new media types.	Ensure that the target volumes for the MOVEVOL request are new media types.	Modify the library default media type to a media type that includes new media types, but not old media types, for each library associated with the source volume's storage group. This allows read requests for old media types to still occur inside the libraries, while any new requests to that storage group will be satisfied with new media types.
Using multifunction libraries without modifying the SCDS	You are using multifunction 3995 optical libraries that use old media types as read-only.	Ensure that the target volumes for the MOVEVOL request are new media types.	Make certain that no libraries associated with the source volume's storage group can write to old media types. The reads from the source volume are still supported and all writes are directed to new media types without making any modification to the SCDS.

For more information regarding updating, validating, and activating an SCDS, see *z/OS DFSMSdfp Storage Administration*.

Minimizing or quiescing contending system activity

Before and during the execution of the Move Volume utility, you should minimize or quiesce the following types of system activity, which might interfere with the utility:

- Any activity involving *reading* from the source volume. Or, if the source volume is an optical volume any activity involving *reading* from the volume on the opposite side of the optical disk. If read activity on the source volume cannot be quiesced, investigate using the UPD=N option on the CBRINIT statement in the IEFSSN*xx* member of PARMLIB during initial program load (IPL). This will avoid conflicting updates to the objects' directory entries.
- Any activity involving *inserting, selecting, updating, or deleting* operations on the OAM DB2 object directory table for the storage group.

System activity includes, but is not limited to the following:

- Scheduled or operator-initiated OSMC storage group management cycles involving the:
  - Storage group that contains the source volume when moving objects from a primary volume.
  - Storage groups that contain objects that have a backup copy on the source volume when moving objects from a backup optical volume.
- Operator-initiated invocations of the Move Volume utility for other volumes requiring the same OAM DB2 object directory tables.
- Operator-initiated invocations of the Volume Recovery utility for other volumes requiring the same OAM DB2 object directory tables.

- Operator-initiated Tape Recycle commands affecting volumes requiring the same OAM DB2 object directory tables.
- Applications using the OSREQ interface to read data from or write data to the storage groups referenced by the Move Volume utility, or both.

Failure to minimize or quiesce this type of system activity might cause operations in the other system activities to complete unsuccessfully (for example, due to the unavailability of the source volume during the execution of the utility) or might interfere with the operation of the utility (for example, causing contention in the DB2 databases required for the utility to operate).

Analyzing resources and tuning OAM for MOVEVOL usage

At a minimum, plan to use the Move Volume utility when contending system activity is low.

When planning to use the Move Volume utility, the same considerations used for planning the number of OSMC storage group management cycles to run concurrently apply. The Move Volume utility is a long-running process that requires a drive for reading and at least one drive for writing for its duration. You might have to tune your existing implementation of OAM for these considerations using the MAXS and or DRIVE STARTUP threshold, or both controls.

For more information, see:

- "MAXS considerations for MOVEVOL processing" on page 225
- "DRIVE STARTUP threshold considerations for MOVEVOL processing" on page 225
- "Tuning OAM" on page 201

MAXS considerations for MOVEVOL processing

Review your specification for MAXS. The Move Volume utility is not controlled by the MAXS value; however, it is a consumer of resources. It is not recommended that you run the Move Volume utility at the same time as an OSMC storage management cycle because this causes a conflict for resources. However, if you plan to run the Move Volume utility at the same time as one or more OSMC storage management cycles, you might need to reduce the number of concurrent OSMC storage management cycles to make resources available for the Move Volume utility. Consider and account for the resources that the Move Volume utility requires first and then distribute the remaining resources for the OSMC storage management cycle with MAXS.

Also, because the MAXS value does not control the Move Volume utility, take resource considerations into account when planning for concurrent executions of the utility (for example, multiple Move Volume utilities running concurrently for different volumes).

DRIVE STARTUP threshold considerations for MOVEVOL processing

Always review your specifications for DRIVE STARTUP threshold and TAPEDRIVESTARTUP for your Object storage groups. The Move Volume utility causes objects to be written to the storage group of which the source volume is a member.

For example, if the Move Volume utility is running at the same time as one or more OSMC storage management cycles for the same storage group, the write activity in the storage group is increased. As a result, you must consider and account for the drives that the Move Volume utility requires in addition to the drives that are required by the OSMC storage management cycle when determining the appropriate value for the threshold. You must perform this action for each Object storage group containing primary source volumes that are processed by the Move Volume utility as well as for any Object Backup storage group for backup source volumes that are processed by the Move Volume utility.

Attention: Running the Move Volume utility concurrently with the OSMC processing for the storage group that the volume belongs to can cause DB2 contention.

DB2 index considerations

There are several factors that influence the performance of the Move Volume utility. The following are items to consider when making processing decisions to maximize the performance of the utility:

- Average object size
- Source and target optical device types
- Number of active utilities
- Primary or backup source volume
- Number of rows in the object directory table for the storage group to which the source volume belongs
- Additional DB2 indexes defined (depending on activity currently in progress)
- Level of the system workload and I/O to the source and target devices

The Move Volume utility performs DB2 SELECTs from the object directory tables (OSM_OBJ_DIR) to prepare the list of objects to be moved. Depending on the current status of the table, DB2 might do a table space scan or an index scan. With large DB2 tables, it is recommended that you verify that DB2 is to use index scans to enhance the efficiency and performance of the Move Volume utility.

The object selection is not overlapped with the read from optical or tape, write to optical or tape, or directory update processes. This means that the elapsed time to complete the MOVEVOL processing consists of the time to select the objects and the time to update the object directory tables after the object movement is complete.

The object selection process consists of executing DB2 SELECT statements from the object directory tables. This process is limited by the DB2 indexes available and the chosen access path. Without the additional recommended indexes, the object selection process requires:

- At least one table space scan of the object directory table for a single storage group associated with the primary volume processed
- A table space scan of *all* object directory tables when a backup volume is processed.

Recommendation: Creating these additional DB2 indexes enhances the performance for the Move Volume utility; however, these indexes can negatively affect other OSMC process. After MOVEVOL processing has completed, drop these additional indexes and rebind the modified plans.

To improve performance for the Move Volume utility, perform the following steps:

1. Create an index on ODLSLOC and ODCLID for each storage group that contains a primary volume that is being processed. For first backup volumes, create an index on ODBKLOC and ODCLID for each storage group that has a significant number of objects (more than 10 000). For second backup volumes, create an index on ODBK2LOC and ODCLID for each storage group that has a significant number of objects (more than 10 000). Use the following SQL statements to define additional DB2 indexes.

   ```
    CREATE INDEX GROUPhlq.OBJDIRX4
           ON    GROUPhlq.OSM_OBJ_DIR
           (
            ODLSLOC   ASC,
            ODCLID    ASC
           )
           USING      VCAT cat_name
           CLOSE      NO
           SUBPAGES   1
          BUFFERPOOL BP1
          PCTFREE    10;
   CREATE INDEX GROUPhlq.OBJDIRX5
          ON    GROUPhlq.OSM_OBJ_DIR
          (
           ODBKLOC   ASC,
           ODCLID    ASC
          )
          USING      VCAT cat_name
          CLOSE      NO
          SUBPAGES   1
          BUFFERPOOL BP1
          PCTFREE    10;
   ```

   Where: *cat_name* is the name of the catalog used under DB2. OAM does not create these indexes using the UNIQUE keyword.

Because moving data from a backup volume requires access to all of the defined storage group directories, creation of the index on ODBKLOC (instead of ODLSLOC as in above example) is important for all storage groups that contain large numbers of objects (whether there are any backup objects on the volume being processed or not).

2. Execute RUNSTATS to collect data on the new indexes. See “Tuning the DB2 databases” on page 202 for more information.

   ____

3. Modify and run CBRPBIND to rebind the following DB2 plans to their packages for the affected storage group: CBRHSVOL, CBRHSBKV, CBRHSBCC, CBRHSPCC, CBRHORCL, and CBRHCNTL. See “CBRPBIND” on page 461, “CBRHBIND and CBRHGRNT” on page 461 for information on binding these plans.

   ____

4. Define data sets for the additional indexes. See “CBRIALC0” on page 424 for more information on defining these data sets.

```
//STEPxx  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSDUMP  DD SYSOUT=*
//SYSIN    DD *
 DELETE
 cat_name.DSNDBC.GROUPhlq.OBJDIRX4.I0001.A001
     CLUSTER
     PURGE
 DELETE
 cat_name.DSNDBC.GROUPhlq.OBJDIRX5.I0001.A001
     CLUSTER
     PURGE
 SET LASTCC=0
 SET MAXCC=0
 DEFINE CLUSTER
 (NAME(cat_name.DSNDBC.GROUPhlq.OBJDIRX4.I0001.A001)
          LINEAR
          SHAREOPTIONS(3 3)
          VOLUMES(vol_ser)
          CYLINDERS(pri_alloc sec_alloc)
          UNIQUE   )
        DATA
 (NAME(cat_name.DSNDBD.GROUPhlq.OBJDIRX4.I0001.A001))
 DEFINE CLUSTER
 (NAME(cat_name.DSNDBC.GROUPhlq.OBJDIRX5.I0001.A001))
         LINEAR
         SHAREOPTIONS(3 3)
         VOLUMES(vol_ser)
         CYLINDERS(pri_alloc sec_alloc)
         UNIQUE    )
       DATA
 (NAME(cat_name.DSNDBD.GROUPhlq.OBJDIRX5.I0001.A001))
/*
```

   Where: *vol_ser* is the volume serial the data set should be placed on; *pri_alloc*, *sec_alloc* is the number of primary and secondary allocations for the data set; *cat_name* is the name of the catalog used under DB2. These indexes are not created using the UNIQUE keyword.

   ____

5. Run the Move Volume utility.

   ____

6. Drop the DB2 indexes used during MOVEVOL processing.

   ____

7. Execute RUNSTATS again.

   ____

8. Rebind the following DB2 plans: CBRHSVOL, CBRHSBKV, CBRHSBCC, CBRHSPCC, CBRHORCL, and CBRHCNTL.

____

Migrating objects from one media to another takes a significant amount of time. With this in mind, there are some things that you should consider that could save time and effort in the future.

- Define expiration criteria for all objects. This should reduce the amount of data and time needed to perform migration to another media.
- Adopt higher capacity media into your installation as soon as it is introduced. If you write all objects to the highest capacity media available, you might be able to avoid a media migration.

See “CBRISQL0” on page 432 for sample job control language (JCL).

Reusing recycled tape and optical volumes

You can use the MOVEVOL command with the RECYCLE option to recycle optical and rewritable tape volumes. After all the objects have been moved off of the tape volume or both sides of an optical platter, OAM processes the volumes as specified in the SETOAM TAPERECYCLEMODE and SETOPT OPTICALREINITMODE keywords in the CBROAM*xx* PARMLIB member. When a tape volume is released from DFSMSdfp, DFSMSrmm also releases the volume.

The main reason to recycle a tape or optical volume is to reuse the volume for new data and to save money on buying new media. Many partially full tapes or optical volumes using slot space in a library is considered wasted space. You can use MOVEVOL with the RECYCLE option to consolidate data from many partially full volumes and move that data to one volume, freeing up volumes that you can use for new data. The MOVEVOL command with the RECYCLE option overrides the SETOAM TAPERECYCLEMODE or SETOPT OPTICALREINITMODE options.

For optical volumes, the MOVEVOL command with the RECYCLE option moves objects on both sides of the optical platter (both volumes). In contrast, the MOVEVOL command without any options moves objects on only one side of the optical platter (one volume).

To recycle an optical or rewritable tape volume, issue the following command:

```
MODIFY OAM,START,MOVEVOL,volser,RECYCLE
```

When a volume is released from DFSMSdfp, DFSMSrmm also releases the volume.

Attention: If the MOVEVOL with RECYCLE process is interrupted, all three of the following conditions are true:

- For tape media, the **logical kilobytes deleted** field in the tape volume is not updated, so it would be inaccurate.
- For rewritable optical media, the deleted objects are not inserted in the Deleted Objects table, so orphaned objects remain on the media and the space is not reclaimed until the volume is reformatted. The number of deleted objects and deleted space are not updated, so it would be inaccurate.
- The tape volume or both sides of the optical platter are left in non-writable mode.

Related reading:

- For more information about the RECYCLE option of the MOVEVOL command , see “Moving objects and recycling the source volume” on page 281 and “Updating the SETOPT OPTICALREINITMODE or SETOAM TAPERECYCLEMODE keyword” on page 222.
- For more information about enabling and managing the TAPE RECYCLE function , see “SETOAM keyword definitions for global level” on page 112.
- For more information about expiring tape and optical volumes, see “Expiring tape and optical volumes” on page 230.
- For more information about using DFSMSrmm to manage tape volumes, see *z/OS DFSMSrmm Managing and Using Removable Media*.

Recycling tape volumes

When you recycle a rewritable tape volume, you can leave the volume in its current storage group, reassign the volume as an OAM scratch volume, or remove the volume from OAM control. After OAM moves the objects off of the tape volume, it recycles the tape volume using the parameters that are specified in the SETOAM TAPERECYCLEMODE statement in the CBROAM*xx* member of PARMLIB. You can immediately reuse the tape volume.

Recycling optical volumes

When you recycle an optical volume, OAM removes the objects from both sides of the optical platter. After OAM removes the objects from the optical platter, it recycles the optical volume using the parameters that are specified in the SETOPT OPTICALREINITMODE statement in the CBROAM*xx* member of PARMLIB. Specify the GROUP parameter to leave the volume in its current storage group, or specify the OAMSCRATCH parameter to reassign the volume as an OAM scratch volume.

You can reuse optical media, as follows:

- OAM reformats the rewritable optical volume to reclaim the used space the next time that it is mounted on a drive. You can reuse the entire rewritable optical volume.
- OAM does not reformat WORM optical volumes because the used space cannot be reclaimed. However, you can recycle WORM media and use the remaining available space on the media.

Example: In this example, SETOPT OPTICALREINITMODE(OAMSCRATCH) is specified in the CBROAM*xx* PARMLIB member. Suppose you have a WORM volume, VOL123, that is 67 percent full. VOL123 belongs to storage group GROUP01. After you recycle VOL123, all the objects are moved off of it and VOL123 is changed to a scratch volume. VOL123 is 100 percent empty, but it still has 33 percent of its original capacity available to write to. This recycled volume is available to any Object or Object Backup storage group that needs it. If this volume had been a rewritable optical volume, you could write to 100 percent of the original capacity.

Deleting recycled tape and optical volumes from OAM

You can use the MOVEVOL command with the DELETE option to delete tape and optical volumes, and remove them from the OAM inventory. The rows that are associated with these volumes are deleted from the TAPEVOL or VOLUME table in DB2. OAM does not reclaim space on rewritable optical platters that have been deleted. If the deleted volume resides in an optical library, the volume is ejected. Then OAM issues message CBR2153I for deleted optical platters or message CBR2165I for deleted tape volumes.

You can also use the MOVEVOL command with the DELETE option to delete a scratch volume from the OAM inventory. When a MOVEVOL command is specified for a scratch volume and the DELETE option is specified, OAM issues CBR9883I to indicate MOVEVOL is processing a scratch volume. The volume is deleted from the OAM inventory without any data movement taking place.

For optical volumes, the MOVEVOL command with the DELETE option moves objects on both sides of the optical platter (both volumes). In contrast, the MOVEVOL command without any options moves objects on only one side of the optical platter (one volume).

The MOVEVOL command with the DELETE option overrides the SETOAM TAPERECYCLEMODE or SETOPT OPTICALREINITMODE options.

Releasing tape volumes using DFSMSrmm: If you are using DFSMSrmm as your tape management system, DFSMSrmm Tape Volume Exit (EDGTVEXT) is invoked after OAM issues the CBR2165I message for a rewritable tape volume, or the CBR2173I message issued for a WORM tape volume. For DFSMSrmm Tape Volume Exit (EDGTVEXT) to return the tape volume to MVS scratch status, the installation must use RACF commands to define the appropriate level of authorization for OAM. Determine the appropriate level of RACF authorization required for OAM to release its own tape volumes. Refer to Authorizing DFSMShsm to DFSMSrmm Resources in *z/OS DFSMSrmm Implementation and Customization Guide* for information about DFSMShsm authorization requirements, which can be used as a base for setting up OAM. Regardless whether the DFSMSrmm exit successfully returns the tape volume to the MVS scratch pool, the tape is deleted from the OAM inventory.

For more information about the DELETE option of the MOVEVOL command, see "Moving objects and deleting the source volume" on page 282. For more information about using DFSMSrmm to manage tape volumes, see *z/OS DFSMSrmm Implementation and Customization Guide*.

Starting tape recycle

Use the F OAM,START,RECYCLE command to select full tape volumes as recycle candidates, based on user defined specifications, and initiate MOVEVOL with RECYCLE processing for those volumes.

The command is as follows:

```
F OAM,START,RECYCLE,scope,DISPLAY|LIM=nn
```

When option DISPLAY is specified, a list of candidate volumes that meet your criteria are written to the hardcopy log. When option LIM=nn is specified, the MOVEVOL with RECYCLE process begins on candidate volumes until either the limit you specified is reached or no more volumes meeting criteria are available.

During Recycle processing, the F OAM,START,RECYCLE command lists the remaining candidate volumes to be processed in the hardcopy log.

Note:

- You can only use this START,RECYCLE command for the recycle of full tape volumes associated with Object or Object Backup storage groups. Optical volumes and tape volumes that are not full continue to be recycled as usual with one cartridge per MOVEVOL command.
- You can only run one RECYCLE command at a time. Your attempt to issue the RECYCLE command fails if a previous RECYCLE command is processing. You can issue a RECYCLE command with the DISPLAY option at any time. This is because the DISPLAY option indicates that a list of candidate volumes meeting the user-defined criteria is to be displayed and no recycle activity actually occurs.
- You can use tape recycle to move objects from tape sublevel 1 source volumes to tape sublevel 1 target volumes, and objects from tape sublevel 2 source volumes to tape sublevel 2 target volumes.

For more information about enabling and managing the TAPE RECYCLE function, and specifying the criteria for candidate volume selection , see "SETOAM keyword definitions for global level" on page 112.

Expiring tape and optical volumes

This section describes expiring tape and optical volumes in Object and Object Backup storage groups.

Expiring tape volumes in object or object backup storage groups

OAM expires objects based on the management class criteria that you specify. When an OAM object is expired, all copies of that object are deleted. The OSMC shelf manager expires object tape volumes that are assigned to Object or Object Backup storage groups when the tape volumes have been marked as full, but no longer contain any valid data.

You can define how OAM processes expired tape volumes. Perform the following steps to set up tape volumes for expiration processing:

1. Use the SETOAM TAPERECYCLEMODE GROUP | OAMSCRATCH | MVSSCRATCH statement in the CBROAM*xx* member of PARMLIB to specify how OAM manages expired tape volumes.
   - Specify the GROUP parameter to leave the tape volume in the currently assigned Object or Object Backup storage group.

     When this tape volume is expired, OAM writes message CBR2166I to the hardcopy log only.

     Note: For WORM tape volumes, the GROUP parameter, if specified, is handled like the MVSSCRATCH parameter because WORM tape volumes cannot be reused. However, once a logical WORM volume expires and the tape management system returns the logical WORM volume to scratch (to reuse the volume's VOLSER), the logical volume can be reused as either WORM or rewritable (R/W). If the volume is subsequently reused as logical WORM, the library will create a new instance of the volume.

- Specify the OAMSCRATCH parameter to return the tape volume to OAM scratch status to be available for use by any Object or Object Backup storage group.

  When this tape volume is expired, OAM writes message CBR2164I to the hardcopy log only.

  Note: For WORM tape volumes, the OAMSCRATCH parameter, if specified, is handled like the MVSSCRATCH parameter because WORM tape volumes cannot be reused. However, once a logical WORM volume expires and the tape management system returns the logical WORM volume to scratch (to reuse the logical volume's VOLSER), the logical volume can be reused as either WORM or rewritable (R/W). If the volume is subsequently reused as logical WORM, the library will create a new instance of the volume.

- Specify the MVSSCRATCH parameter to return the tape volume to the MVS scratch pool and remove it from the OAM inventory.

  When this tape volume is expired and purged from the inventory, OAM writes message CBR2165I for a rewritable tape volume or CBR2173I for a WORM tape volume to the hardcopy log only. However, once a logical WORM volume expires and the tape management system returns the logical WORM volume to scratch (to reuse the logical volume's VOLSER), the logical volume can be reused as either WORM or rewritable (R/W). If the volume is subsequently reused as logical WORM, the library will create a new instance of the volume.

  In an OAMplex, the OAM member that initially releases the tape volume issues the appropriate CBR21xxI message; then the other OAM members in the OAMplex each issue a CBR7404I message.

  OAM issues the CBR2165I or CBR2173I message to indicate that all knowledge of the specified tape volume is being removed from the OAM tape volume inventory. If your installation uses DFSMSrmm to manage OAM object tapes, determine the appropriate level of RACF authorization required for OAM to release its own tape volumes. For DFSMSrmm Tape Volume Exit (EDGTVEXT) to return the tape volume to MVS scratch status, the installation must use RACF commands to define the appropriate level of authorization for OAM. Refer to Authorizing DFSMShsm to DFSMSrmm Resources in *z/OS DFSMSrmm Implementation and Customization Guide* for information about DFSMShsm authorization requirements, which can be used as a base for setting up OAM. Installations that use other tape management systems might need to perform similar actions.

____

2. In the management class, specify the following expiration criteria for the objects that reside on tape volumes:

   - Retention Limit
   - Expire After Days Non-Usage
   - Expire After Date/Days
   - RETPD/EXPDT

____

3. The OSMC cycle expires tape volumes that belong to Object or Object Backup storage groups. If the expiration date for a volume is reached and no active objects exist on the volume, OSMC expires the volume.

____

Result: Eligible tape volumes in an Object or Object Backup storage group, which have expiration dates of less than today, have no more active objects and are full, are expired during the OSMC cycle. These expired volumes are recycled based on the criteria that is specified in the SETOAM TAPERECYCLEMODE statement.

Related reading:

- For more information about the SETOAM parameters for expiring tape volumes, see "SETOAM keyword definitions for global level" on page 112 and "SETOAM keyword definitions for STORAGEGROUP subparameters" on page 121.

- For more information about setting expiration attributes in the management class, see *z/OS DFSMSdfp Storage Administration*.
- For more information about setting the appropriate level of authorization to intercept messages, see *z/OS DFSMSrmm Implementation and Customization Guide*.

Expiring optical volumes in object or object backup storage groups

The OSMC shelf manager expires rewritable optical platters that are assigned to the Object or Object Backup storage group when valid data no longer exists on either side of the platter. The OSMC shelf manager expires WORM optical platters that are assigned to the Object or Object Backup storage group when both sides of the platter have been marked full, but no valid data exists on either side of the platter.

Rewritable optical volumes: You can specify how OAM processes rewritable optical cartridges that have expired. These volumes are either retained in the current Object or Object Backup storage group, or returned to OAM scratch status.

WORM optical volumes: Because WORM optical media cannot be reclaimed, all knowledge of a WORM cartridge is removed from the OAM inventory when the cartridge expires. If the WORM volume is in an optical library, it is ejected from the library. The volume records are removed from the OAM VOLUME table. OAM writes message CBR2153I to the hardcopy log that indicates which optical volumes have expired. You can dispose of the WORM optical volume according to your installation's regulations.

Perform the following steps to set up optical volumes for expiration processing:

1. Use the SETOPT OPTICALREINITMODE statement in the CBROAM*xx* member of PARMLIB to specify how OAM manages expired optical volumes.
   - Specify the GROUP parameter to leave expired rewritable optical volumes in the currently assigned Object or Object Backup storage group.
   - Specify the OAMSCRATCH parameter to return expired rewritable optical volumes to the *SCRTCH* storage group to be available for use by any Object or Object Backup storage group.

   ____

2. In the management class, specify the following expiration criteria for the objects that reside on optical volumes:
   - Retention Limit
   - Expire After Days Non-Usage
   - Expire After Date/Days
   - RETPD/EXPDT

   ____

3. The OSMC cycle expires optical volumes that belong to Object or Object Backup storage groups. If the expiration date for a volume is reached and no active objects exist on the volume, OSMC expires the volume.

   ____

Result: Eligible rewritable optical platters in the Object or Object Backup storage group, which have expiration dates of less than today and have no more active objects, are expired during the OSMC cycle. These expired rewritable volumes are recycled based on the criteria that is specified in the SETOPT OPTICALREINITMODE statement. Eligible WORM optical platters in the Object or Object Backup storage group, which have expiration dates of less than today, have no more active objects, and have been marked full. These volumes are expired during the OSMC cycle. These expired WORM volumes are removed from the OAM volume inventory.

Related reading:

- For more information on the SETOPT parameters for optical volumes, see “SETOPT statements for options” on page 129.

- For more information about setting expiration attributes in the management class, see *z/OS DFSMSdfp Storage Administration*.

Using the OAM Object Tape Volume Return to MVS Scratch exit routine

The OAM Object Tape Volume Return to MVS Scratch exit routine, CBRUXTVS_EXIT, can be used to notify the installation's tape management system that all knowledge of a given tape volume has been removed from OAM's tape volume inventory. This is a notification only exit; OAM does not change it's tape volume expiration processing regardless of the return code supplied by the user exit. The exit is invoked after OAM issues the CBR2165I message indicating that OAM has removed the tape volume from the OAM inventory and released it for return to the MVS scratch pool.

The exit point uses Dynamic Exit Services, CSVDYNEX, which allows multiple exit routines to be simultaneously associated with a single exit point. No additional steps are required for you to use this dynamic exit with DFSMSrmm. The DFSMSrmm tape exit (EDGTVEXT) is always invoked independently of the dynamic exit routines. If your installation manages tape volumes with any tape management systems other than DFSMSrmm, you can write the exit routine(s), or use the same load module used for the ARCTVEXT exit and add them to this exit using the MVS dynamic exits facility.

This CBRUXTVS_EXIT exit point can provide a tape management system, through user-written or system exit routines that have been defined to the dynamic exits facility, with the information required to maintain an accurate inventory of OAM tapes. However, you do not have to use this exit point to use tapes with OAM.

The exit routine is called when OAM is returns an object tape to MVS scratch after it no longer contains valid files. The tape volume is deleted from the OAM TAPEVOL database before the exit is called. This is a notification only exit. A non-zero return code or an abend in the exit module has no effect on OAM's object tape expiration processing. No error message is issued for a non-zero return code, but an error message is issued for an abend in the exit module.

Dynamic exits facility

The dynamic exits facility is a set of services that you can use through any of the following methods:

- The EXIT statement of the PROGxx parmlib member. The EXIT statement allows an installation to add exit routines to an exit, change the state of an exit routine, delete an exit routine for an exit, undefine an implicitly defined exit, and change the attributes of an exit.

  The PROGxx EXIT statement interacts with the PROG=xx parameter of IEASYSxx and the SET PROG=xx command. At IPL, operators can use PROG=xx to specify the particular PROGxx parmlib member the system is to use. During normal processing, operators can use the SET PROG=xx command to set a current PROGxx parmlib member. See *z/OS MVS Initialization and Tuning Reference* for information about the PROGxx parmlib member.
- The SETPROG EXIT operator command. This command performs the same functions as the EXIT statement of the PROGxx parmlib member. See *z/OS MVS System Commands* for information about the SETPROG EXIT command.
- The CSVDYNEX macro. The CSVDYNEX macro can be used to define exits to the dynamic exits facility, control their use within a program, and associate one or more exit routines with those exits. It can also be used to associate exit routines with the existing SMF and allocation exits, which have been defined to the dynamic exits facility. The CSVDYNEX macro provides a superset of the functions available through the SETPROG EXIT operator command and the EXIT statement of the PROGxx parmlib member. See *z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN* for information about the CSVDYNEX macro.

An installation can use any of these methods to control dynamic exits. For example, an exit routine can be associated with the CBRUXTVS_EXIT exit point using the CSVDYNEX ADD request, the SETPROG EXIT,ADD operator command, or the EXIT statement of PROGxx.

Adding the installation exit routines to CBRUXTVS_EXIT

Users of DFSMSrmm need not provide an exit routine for this exit point because OAM always invokes the DFSMSrmm EDGTVEXT exit routine directly whenever an OAM tape volume is returned to MVS scratch. However, if you are using a tape management system other than DFSMSrmm, then you need to add the module name of the installation exit routine provided by the tape management system with the OAM dynamic exit point, CBRUXTVS_EXIT. For example, if you are using the DFSMShsm ARCTVEXT module, you could specify the following EXIT statement in the PROGxx PARMLIB member:

```
EXIT ADD     EXITNAME(CBRUXTVS_EXIT)
                               MODNAME(ARCTVEXT)
                               STATE(ACTIVE)
```

You can associate multiple exit routines to the CBRUXTVS_EXIT exit point by specifying multiple EXIT ADD statements. See *z/OS MVS Initialization and Tuning Reference* for information about the PROGxx parmlib member.

Alternatively, you can add the exit routines to this exit point with the SETPROG EXIT command. See "Adding an Exit routine to an Exit" in *z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN* for details, or see *z/OS MVS System Commands* for information about the SETPROG EXIT command.

Modifying the installation exit routine

To update exit modules associated with the OAM Object Tape Return to MVS Scratch dynamic exit (CBRUXTVS_EXIT) while OAM is up running; perform the following steps:

1. Change and recompile the exit module
2. Update the library resident copy of the exit module such as SYS1.LINKLIB
3. Perform LLA refresh command for the exit module
4. Modify the current copy of the exit module to be inactive or Delete the current copy of the exit module, by using the MVS operator command:

   ```
   SETPROG EXIT,MODIFY,EXITNAME=CBRUXTVS_EXIT,MODULENAME=m,STATE=INACTIVE
   ```

   or

   ```
   SETPROG EXIT,DELETE,EXITNAME=CBRUXTVS_EXIT,MODULENAME=m
   ```

5. Modify the new copy of the exit module to be active or Add the new copy of the exit module, by using the MVS operator command:

   ```
   SETPROG EXIT,MODIFY,EXITNAME=CBRUXTVS_EXIT,MODULENAME=m,STATE=ACTIVE
   ```

   or

   ```
   SETPROG EXIT,ADD,EXITNAME=CBRUXTVS_EXIT,MODULENAME=m
   ```

Displaying the installation exit routine

The installation can use the DISPLAY PROG,EXIT command to display the exit routines associated with the OAM object tape return to MVS scratch dynamic exit (CBRUXTVS_EXIT). The syntax for the DISPLAY PROG,EXIT command is:

```
DISPLAY PROG,EXIT,
                        {EXITNAME|EX|EN}=CBRUXTVS_EXIT
                        [,L={a|cc|cca|name|name-a}]
```

See *z/OS MVS System Commands* for more information about the DISPLAY PROG,EXIT command. See *z/OS MVS Installation Exits* for more information on dynamic exits.

Processing requirements for the installation exit

The installation exit allows your tape management system to accurately reflect OAM's object tape usage. You must use the tape management system facilities to notify itself that OAM has deleted the tape from its database. After the exit sends the notification to the tape management system, it returns control to OAM. Tape deletions cannot be canceled.

Writing the exit routine

If you are using the DFSMShsm ARCTVEXT exit or an exit supplied with a tape management system, include the coding required for the CBRUXTVS_EXIT exit in your existing exit. The coding considerations for this exit routine are similar to those for the DFSMShsm ARCTVEXT exit or the DFSMSrmm EDGTVEXT exit, including entry and exit linkage. A non-zero return code or an abend in the exit module has no effect on OAM's object tape expiration processing. No error message is issued for a non-zero return code, but an error message is issued for an abend in the exit module.

The exit routine must be reentrant and run in either the 24-bit or 31-bit addressing mode (AMODE). You should always return to the caller in the caller's AMODE.

Registers on entry to the exit routine

Table 35. Registers on entry to the exit routine

Register	Content
1	Address of the parameter list
13	Address of standard 72-byte save area
14	Return address of the caller
15	Entry point address of the exit routine

Note: The 72-byte save area is provided solely to enable exit routines to save the registers on entry. Its contents are not used by the Dynamic Exit Services nor by the exit.

Registers on exit

There is no requirement to return any values in registers from the exit routines, so it is not necessary to restore any registers on exit from the exit routines.

Parameter list

The CBRUXTVS_EXIT is modeled to interface directly with the ARCTVEXT exit, and therefore supports the ARCTVEXT parameter list. Refer to *z/OS MVS Installation Exits* for more information on the ARCTVEXT parameter list. On entry to the exit routines, the Register 1 contains the address of the parameter list which contains the following information:

Table 36. ARCTVEXT parameter list

Offset	Length or Bit Pattern	Description
0	4	The address of an 8-byte data area, where the first 6 bytes contain the volume serial number of the volume having expired, followed by a 2-byte field
4	4	The address of a full word binary return code described below

Note: The high order bit is set to 1 to indicate the end of the parameters

Data area

The 8-byte data area is passed to the exit routine, none of them may be modified:

Table 37. ARCTVEXT parameter list

Offset	Length or bit pattern	Description
0	6	The volume serial number of the expired tape volume
6	1	Flag byte one
	1...	Tape volume is being purged
	.1..	Data sets on tape volume are expiration date protected
7	1	Flag byte two

Return codes

The exit routine may indicate return codes as follows:

Table 38. ARCTVEXT return codes

Return Code	Meaning
0	The processing was successful
non-zero	The processing failed

Related reading:

- *z/OS DFSMS Installation Exits*
- "Support Use Information" in *z/OS MVS Installation Exits*
- "CSVDYNEX Terminology" in *z/OS MVS Programming: Authorized Assembler Services Guide*

Synchronizing OAM scratch tape

OAM provides the ability for tape volumes to be returned to OAM scratch status. When a tape is returned to OAM scratch status, it is available to be claimed and used by any Object or Object Backup storage group with the same unit name and data class. For supported levels of OAM, tape volumes introduced into the OAM volume inventory will have valid unit name and data class values in the TAPEVOL table.

However, tape volumes that were introduced into the OAM inventory before z/OS V1R5 and not reused since will have blank unit names and data classes associated with them by default. Therefore, these volumes are never selected for reuse after being returned to OAM scratch status. To make such volumes eligible for reuse when they are returned to OAM scratch status, you must either manually update their OUNITNAM and DATACLAS fields in the TAPEVOL table, or enable SETOAM OAMSCRATCHSYNCH mode in the CBROAM*xx* PARMLIB member.

Enabling SETOAM OAMSCRATCHSYNCH mode

You can use the SETOAM OAMSCRATCHSYNCH statement to specify how OAM is to react to blank DATACLAS and OUNITNAM fields in the DB2 TAPEVOL table that are associated with a tape volume that is being returned to OAM scratch status. In installations where the original unit name of the tape is not changed because the tape is mounted in an IBM automated tape library, you can enable the SETOAM OAMSCRATCHSYNCH statement to fill in the data class and unit name for the tape volume. Thus, these tape volumes are eligible to be reused.

If the DATACLAS field is blank, OAM sets DATACLAS to the same value as the data class that was associated with the storage group for the volume before it was returned to OAM scratch. If the OUNITNAM field is blank and the value in UNITNAME is a generic device, OUNITNAM and UNITNAME are both set to the same value as the UNITNAME for the storage group. If OUNITNAM is blank and the value in UNITNAME is an esoteric device name, both OUNITNAM and UNITNAME remain unchanged.

Restriction: This option cannot be used if your object tape volumes reside inside an IBM automated tape library.

Manually updating the OUNITNAM and DATACLAS fields in the TAPEVOL table

If you manually update the OUNITNAM and DATACLAS fields in the TAPEVOL table, follow these steps:

1. OAM must be down before you make any changes to the UNITNAME or OUNITNAM fields in the DB2 table.
2. If you are using OAM object tape in an IBM automated tape library and opt to return these tape volumes to OAM scratch upon expiration, determine the appropriate esoteric name to insert into the OUNITNAM field.
3. If you are not using OAM object tape in an IBM automated tape library, you can copy the value that is currently in the UNITNAME field to the OUNITNAM field.
4. Set the DATACLAS value based on the Object or Object Backup storage group to which the tape volume belongs.

Related topics: For more information about the SETOAM OAMSCRATCHSYNCH statement, see "SETOAM keyword definitions for global level" on page 112 .

Processing object expiration

Each object stored by OAM is assigned an expiration date. This expiration date is derived using the retention period (RETPD) keyword on the OSREQ STORE macro when the object was stored, the expiration rules in the SMS management class assigned to the object, or both.

When a class transition occurs, the SMS storage class and management class ACS routines are invoked. The SMS storage class ACS routine can assign a new management class to the object. Input to the SMS storage class and management class ACS routine indicates that the reason the routine is invoked is for an OAM object class transition. As a result of a new SMS storage class being assigned to the object, the physical location of the object in the OAM storage hierarchy might change. As a result of a new SMS management class assignment to the object, the expiration date of the object might change as well as its backup requirements.

If an OSREQ CHANGE request is performed and a new retention period is specified with the RETPD keyword on the OSREQ CHANGE macro, then the expiration date of the object is recalculated based on the period specified with the RETPD keyword on the OSREQ CHANGE macro. This is true regardless of the media type of the primary or backup copy.

Note: If the requested expiration date is earlier than the retention date for a retention-protected object, the expiration date is set to object's retention date.

When the OSMC storage management cycle determines that an object has reached or passed its expiration date and is not in deletion-hold status, it invokes the Auto-Delete Installation Exit to approve the deletion of the object. If the exit approves the deletion of the object, the object is expired. When an object expires, all copies of the object are deleted, the row for the object in the Object Directory Table is deleted and any reusable resources are reclaimed. For objects residing on tape volumes, the number of logical KB deleted from the tape volume is incremented for each object deleted.

If at any time an object's management class results in the object's expiration date being set to 9999/12/31 while that object is on removable media, that volume's expiration date is set to 9999/12/31. The volume never expires, even if the object's management class changes at a later date allowing the object to expire. Be aware of the affects of expiration dates that can be set by a management class, even if it is being used as an interim management class for an object.

For primary copies in the DB2 sublevel, OSMC deletes the copy from the DB2 sublevel. For primary copies in the file system or on optical or tape, OSMC makes a delete request to LCS to delete the copy from the file system, optical, or tape.

For backup copies on optical or tape, OSMC makes a delete request to delete all the copies from optical or tape.

For both optical and object tape volumes, the number of logical KB of data that is deleted from an OAM optical or tape volume containing objects is calculated and stored in the VOLUME or TAPEVOL table. Each time LCS receives a request to delete an object from an optical or tape volume, LCS updates the number of logical KB deleted for that volume. Because an application could choose to do a DB2 ROLLBACK after requesting a delete, the count of the logical KB deleted in the VOLUME or TAPEVOL table is an approximation.

Destroying and deleting expired data

Deleting an optical-disk-resident object from WORM media removes the directory entry for that object; however, the data itself remains on the disk, because of the write-once characteristic of the media. To ensure that confidentiality is maintained, it might be necessary to physically destroy disks that contain expired sensitive material. Follow the manufacturer's directions to safely dispose of optical media.

When a delete request is issued for an object that resides on rewritable optical media, and the directory table entry for the object is deleted upon approval of the Auto-Delete Installation Exit, a row is added to the Deleted Object table in DB2 to indicate the object is to be deleted (physically erased). If the volume where the object resides is selected for a write request, LCS deletes all objects on the volume indicated within the Deleted Object table. LCS also deletes groups of objects when a drive completes a request and there are no other higher priority requests to perform.

When a delete request or transition to another level is completed for an object that resides on the filesystem, a row is added to the Filesystem Delete Table (FSDELETE) in DB2 to indicate that the object needs to be deleted (physically erased). A dedicated OAM task will run periodically to remove these objects from the filesystem.

[NOT Programming Interface Information] When OAM processes physical deletes of objects as background work in periods of inactivity, the order of volume preference is as follows:

- Deletions from the mounted volume
- Deletions from the opposite side of the mounted volume
- Deletions from the volume which has the largest amount of space to be deleted (valid only when considering library resident drives)

[End NOT Programming Interface Information]

Diagnosing nondeletion/expiration of objects during OSMC processing

This information provides suggestions for handling the situation of objects not deleting or expiring during the OSMC processing cycle. This information allows you to identify, investigate, and overcome these problems.

If, during OSMC processing, objects that should have expired do not actually get deleted from the database, check if the auto-delete installation exit (CBRHADUX) is in SYS1.LINKLIB to be available to OAM to LOAD it when it starts. If the EXIT is not used, the OBJECTS will need to be deleted manually. One of the following situations is probably causing the problem:

- The auto-delete installation exit (CBRHADUX) has been invoked and completed with a return code of either 0 or 8.
- The object has not been selected for expiration processing by OSMC.

To diagnosis the problem, examine the system console log from the start of the OSMC processing of the storage group containing the object through the completion of the OSMC storage group processing. Look for any CBR*xxxxx* messages that might indicate a problem.

The CBR*xxxxx* messages are documented in *z/OS MVS System Messages, Vol 4 (CBD-DMO)*.

Checking CBRHADUX

If you have not successfully deleted objects during OSMC in the past, or you have just moved to a new release of OAM, a good place to start is by examining the Auto-Delete Installation Exit, CBRHADUX. To allow objects to be deleted during OSMC processing, perform the following steps:

1. Modify CBRHADUX to allow deletes.

   CBRHADUX returns with a return code 12 if the exit cannot dynamically allocate the group X.OBJECT.DELETE.NOTIFY data set. Check the syslog for CBR, IGD or IEF messages following the start of OSMC processing for the storage group.
2. Refresh LINKLIB.
3. Stop and start OAM so that the new copy of CBRHADUX is available to OAM.

For more information on the CBRHADUX installation exit, see "Auto-delete installation exit (CBRHADUX)" on page 573.

Objects not selected for expiration processing by OSMC

There might be a number of reasons why an object is not selected for expiration during OSMC processing. The expiration date assigned to an object might not be correct, the object might be in a management class that has different expiration attributes assigned to it than you think it has, a collection entry might not be found within the collection table, or the Deletion-Hold flag in the ODSTATF field might be set on. The first step in investigating why OSMC does not select the object for expiration processing is to examine the object directory table (HLQ.OSM_OBJ_DIR). The following SQL command retrieves the object's directory table entry:

```
SELECT * FROM HLQ.OSM_OBJ_DIR
WHERE ODNAME = 'object_name';
```

Examine the ODSTATF field in the object directory to verify that deletion-hold (ODSTATF_DELHOLD) is not turned on for this object. See Table 53 on page 488 for more information about the ODSTATF field.

Verify that the pending action date (ODPENDDT) is the current date or earlier.

If the expiration date (ODEXPDT) in the object directory table is less than or equal to the current date, then the object should expire. If the expiration date is the special value 9999-12-31, the object never expires. If the expiration date is 0001-01-01, OAM uses the management class attributes to determine the expiration date. If the expiration date is 0002–02–02, the object is in event-based-retention mode and cannot be deleted until OAM receives notification by the EVENTEXP keyword on an OSREQ API request.

During OSMC processing, a management cycle called the shelf manager runs after storage management has completed for each Object or Object Backup storage group. Shelf manager examines all the volumes in an Object or Object Backup storage group to determine if there are any volumes on which all of the objects are expired. If all of the objects on a tape volume are expired and the tape volume is full, the tape volume is dispositioned as specified in the SETOAM TAPERECYCLEMODE keyword of the CBROAM*xx* PARMLIB member.

If all objects on both sides of a rewritable optical platter are expired, the platter is reinitialized and dispositioned as specified in the SETOPT OPTICALREINITMODE keyword of the CBROAM*xx* PARMLIB member. If all objects on both sides of a WORM optical platter are expired and both sides of the platter are full, the WORM platter is scheduled for ejection and the operator is prompted to remove it from the input/output station of the optical library. Knowledge of the WORM volume is removed from the OAM inventory.

Management class and expiration attribute definitions

To determine the management class name of the object, use the following SQL statement:

```
SELECT * FROM OAMADMIN.CBR_MGT_CLASS_TBL M, HLQ.OSM_OBJ_DIR D
WHERE   ODNAME = 'object_name'
AND     D.ODMCNUM = M.ODMCNUM;
```

List the ISMF definition of the management class. Be sure to specify 'ACTIVE' as the SCDS name. Examine the expiration attributes. If both EXPIRE AFTER DAYS NON-USAGE and EXPIRE AFTER DATE/DAYS are NOLIMIT, the object never expires. The value of EXPIRE AFTER DAYS NON-USAGE is added to the last referenced date (ODLREFDT) in the object directory table to calculate the expiration date. If the last referenced date is not set, the creation date is used. If the value of EXPIRE AFTER DATE/DAYS is an explicit date, that date is used as the expiration date. If the value is a number of days, the expiration date is calculated based on the number of days added to the creation date (ODCREATS) in the object directory table.

For additional information on expiration attribute processing, see *z/OS DFSMSdfp Storage Administration*.

Collection entry not found in the collection table

Another common cause of an object not being selected by OSMC is the collection entry not being found in the collection table (OAMADMIN.OBJ_COLLECTION_TBL) for the collection ID (ODCLID) in the object directory table for the storage group being processed.

You can verify that the collection table entry exists for the object with the following SQL statement:

```
SELECT * FROM OAMADMIN.CBR_COLLECTION_TBL C, HLQ.OSM_OBJ_DIR D
WHERE   ODNAME = 'object_name'
AND     D.ODCLID = C.ODCLID;
```

One row should be returned identifying the collection that you expect to be associated with the object.

Documentation for your IBM representative

If you have not identified the reason for objects not being deleted during OSMC, you might want to contact the IBM support center. Make sure that you have the following documentation available:

- The contents of the following tables:
  - OAMADMIN.CBR_MGT_CLASS_TBL
  - OAMADMIN.CBR_STO_CLASS_TBL
  - OAMADMIN.CBR_COLLECTION_TBL
- The contents of an object directory table entry for one of the objects that you expected would be deleted.
- A screen print of the storage class and management class definitions using `ACTIVE` as the SCDS name.
- The management class and storage class ACS routines.
- The syslog from the time the OSMC message CBR9200I is issued indicating the start of processing for the Object storage group until message CBR9201I is issued indicating processing has completed.

Diagnosing unexpected results of object movement during OSMC processing

During OSMC processing, objects might not always transition within the storage hierarchy as expected, in spite of the performance objectives of the assigned storage class and management class for the object. Should an installation experience differences between its expectations and the actual location of the object within the hierarchy after OSMC processing, the following information might assist in diagnosing why objects are not moved as expected.

Objects not moved to a new storage level during OSMC processing

This information allows you to identify, investigate, and overcome the problem when objects do not move to the expected level within the hierarchy as defined by the ISMF storage class and management class definitions.

To determine where an object resides within the storage management hierarchy (on disk, optical, or tape), query the directory table entry for the object using the following SQL statement:

```
SELECT * FROM HLQ.OSM_OBJ_DIR
WHERE ODNAME = 'object_name';
```

The content of the ODLOCFL field indicates the location of the primary copy of the object as follows:

- blank—primary copy resides on an optical disk volume
- D—primary copy resides on disk sublevel 1 (DB2)
- E—primary copy resides on disk sublevel 2 (file system)
- T—primary copy resides on a tape sublevel 1 volume
- U—primary copy resides on a tape sublevel 2 volume
- R—primary copy has been recalled to disk sublevel 1 (DB2)
- 2—primary copy has been recalled to disk sublevel 2 (file system)

If the object is not found on the expected medium, it might be because an error has occurred during the OSMC cycle or because the SMS environment has not been properly defined to allow objects to make a transition between storage classes.

To determine if there has been an error during the OSMC cycle, examine the system console log from the start and through the completion of OSMC processing of the Object storage group containing the object.

Any CBR*nnnn* messages found in this log might indicate a problem.

Note: Objects greater than 256MB are supported on the DB2 sublevel, file system sublevel, tape sublevel 1, and tape sublevel 2 in the OAM storage hierarchy. OSMC will not attempt to transition or write backup copies of objects greater than 256MB to the optical level. CBR9226I will be issued once per collection if any objects greater than 256M (268,435,456 bytes) were encountered in a collection during OSMC processing that required writing the object or an object backup copy optical level of the OAM storage hierarchy.

For more information concerning these messages and how to resolve the problems that prompted these messages, see *z/OS MVS System Messages, Vol 4 (CBD-DMO)*.

Objects not selected for class transition processing

If once the system console log is examined no error messages are found, the possibility of OSMC not selecting the object for processing needs to be investigated.

The first step in determining if OSMC did not perform a class transition against the object, and why, is to examine the object's directory table entry. Use the following SQL statement to query the object:

```
SELECT * FROM HLQ.OSM_OBJ_DIR
WHERE   ODNAME = 'object_name';
```

Verify that the pending action date (ODPENDDT) is the current date or earlier. An object with a pending action date that is assigned a date in the future will not be selected for processing. For example, if today's date is 01/05/2017, and the pending action date for the object is 01/05/2017 or earlier, the object should be chosen for OSMC processing. If the pending action date for the object is 03/05/2017, the object is not chosen for OSMC processing until today's date is the same or later than the pending action date for the object.

Check the expiration date (ODEXPDT). If the object's expiration date has been reached, OSMC has no reason to move it to another level in the hierarchy.

For more information concerning expiration dates, see "Processing object expiration" on page 237.

Collection entry not found in the collection table

Another cause for nonselection of an object during OSMC processing might be that the collection entry for the collection ID associated with the storage group being processed cannot be found in the collection table of the object directory.

To verify that the collection table entry exists for the object, use the following SQL statement:

```
SELECT * FROM OAMADMIN.CBR_COLLECTION_TBL C, HLQ.OSM_OBJ_DIR D
WHERE   ODNAME = 'object_name'
AND     D.ODCLID = C.ODCLID;
```

One row should be returned from the collection table identifying the collection that is expected to be associated with the object. If there is no collection table entry, the object is not selected for processing.

Check management class definitions

To determine the storage class and management class names associated with the object, use the following SQL statement:

```
SELECT * FROM OAMADMIN.CBR_STO_CLASS_TBL S,
         OAMADMIN.CBR_MGT_CLASS_TBL M,
         HLQ.OSM_OBJ_DIR D
WHERE   ODNAME = 'object_name'
AND     D.ODSCNUM = S.ODSCNUM
AND     D.ODMCNUM = M.ODMCNUM;
```

If no rows are produced from the select statement, the object is not processed.

List the ISMF definition of the management class name returned from the select statement. This is the management class currently assigned to the object. Specify ACTIVE as the SCDS name.

Calculate the date of the next class transition as follows:

- If TIME SINCE CREATION has been used, add the values for time since creation to the creation date of the object (ODCREATS in the object's directory table entry).
- If TIME SINCE LAST USE has been specified, add the values for time since last use to the last referenced date of the object (ODLREFDT in the object’s directory table entry).
- If PERIODIC has been specified, see the period definitions in the *z/OS DFSMSdfp Storage Administration*.

In all cases, if the date calculated is in the future, no class transition occurs.

Check management class and storage class ACS routines

If the ISMF management class definition indicates it is time for a class transition, the storage class ACS routine is invoked followed by the management class ACS routine. The ACS input variable &ACSENVIR is set to CTRANS. In order for an object to be moved to a different level of the hierarchy, a new storage class must be assigned when &ACSENVIR is CTRANS.

Use the ISMF ACS test panels to determine what storage class is being assigned. Possible sources of error are as follows:

- CTRANS logic is not executed.
- The current storage class is assigned, or a new storage class is assigned, but the new storage class definition places the object at the same level of the hierarchy.

Check storage class definitions

Display the ISMF storage class definition of the storage class assigned by the ACS routines. If INITIAL ACCESS RESPONSE SECONDS is zero, then the object is stored on DASD. If a nonzero value is specified, and SUSTAINED DATA RATE is greater than or equal to 3, then the object is stored on tape. Otherwise, the object is stored on optical media. If the media assigned by the storage class is unexpected, correct the storage class definition. If the wrong storage class is being assigned, correct the ACS routine. In either case, validate and activate the new configuration.

Documentation for your IBM representative

If you have not identified the reason for objects not moving to different levels of the hierarchy during OSMC, you might want to contact the IBM support center. You should have the following documentation available:

1. Output from the select statements above.
2. The contents of the following tables:
    - OAMADMIN.CBR_MGT_CLASS_TBL

- OAMADMIN.CBR_STO_CLASS_TBL, and
- OAMADMIN.CBR_COLLECTION_TBL.

3. The contents of an object directory table entry for one of the objects that are expected to make a class transition.
4. A screen print of the storage group, storage class, and management class definitions using "ACTIVE" as the SCDS name.
5. Management class and storage class ACS routines.
6. The syslog from the time OSMC message CBR9200I is issued to indicate the start of processing for the Object storage group until message CBR9201I is issued indicating processing has completed.

Invoking the OSREQ macro through the OSREQ TSO/E command processor

The OSREQ command is a TSO/E command processor that closely resembles the OSREQ macro, a programming interface, provided by OAM. The OSREQ macro is used for the storage, retrieval, query, deletion of objects, and comparing primary data to backup data. The OSREQ TSO/E command is used to exercise the OSREQ macro interface and OAM without having to explicitly use the OSREQ ACCESS and UNACCESS macros to connect and disconnect the macro to OAM.

Each time the OSREQ command is issued, the OSREQ TSO/E command processor performs the OSREQ ACCESS macro between itself and OAM. If the OSREQ ACCESS macro is successful, the OSREQ command processor continues and performs the requested function. After the function is performed, the OSREQ command processor disconnects itself from OAM through an OSREQ UNACCESS macro command.

The following functions can be performed against objects: store, retrieve, query, delete, compare, and change the management class, storage class, and retention period.

Note: Check with your application provider for any corequisite support that may be required in order to store objects larger than 50MB.

Note: For OSREQ generic QUERY, the percent ('%') and underscore ('_') wildcard characters introduced in z/OS Version 1 Release 11 cannot be used with the TSO/E command processor. Only the asterisk ('*') wildcard character is valid for use with the TSO/E OSREQ command processor. Refer to the *z/OS DFSMS OAM Application Programmer's Reference* for detailed information on OSREQ QUERY.

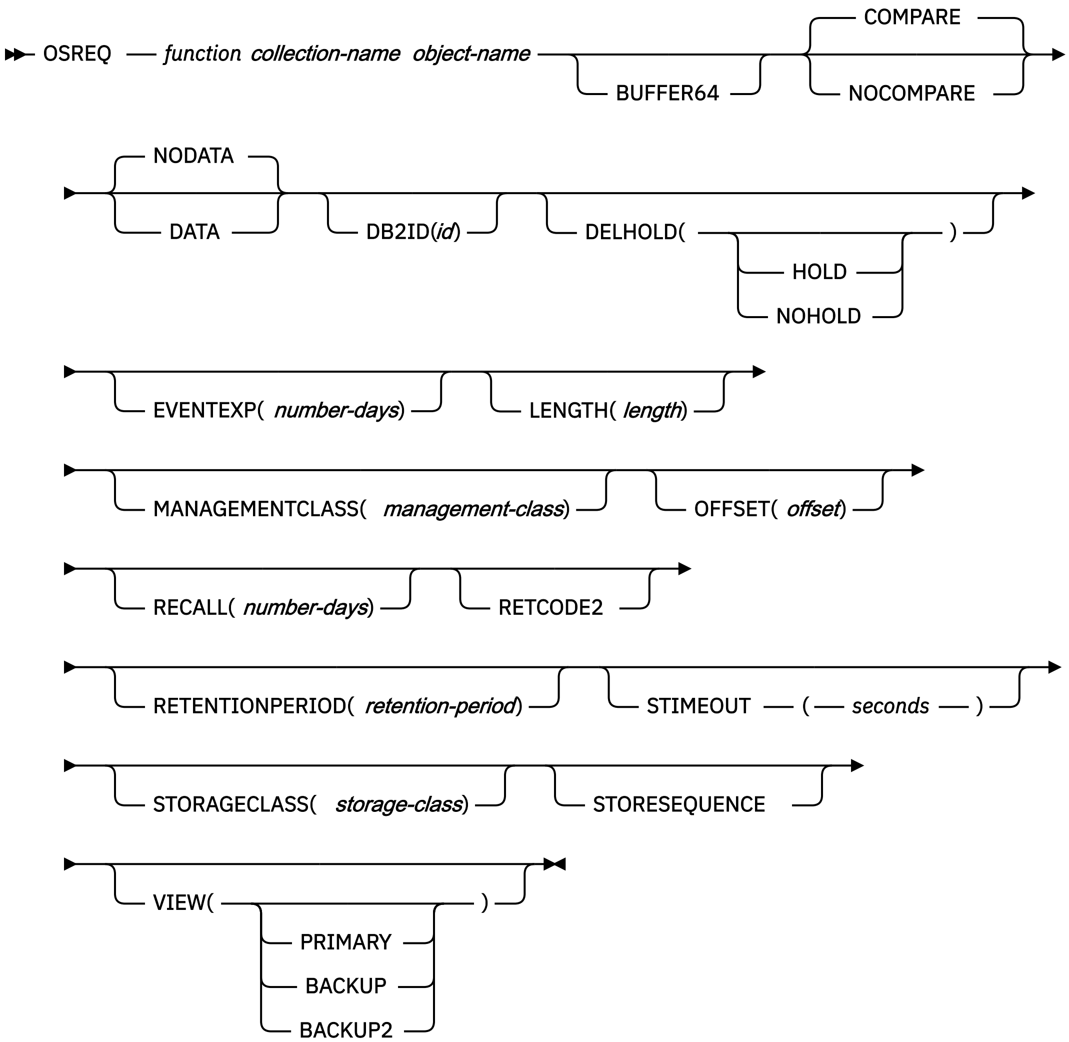
Recommendation: This command processor verifies object support after product installation. You might also use it as a tool to assist in recreating a customer problem in a controlled environment. You cannot use this tool to store actual data, because the STORE function creates "dummy" data.

For more information on using the OSREQ macro, see the *z/OS DFSMS OAM Application Programmer's Reference*.

OSREQ TSO/E command syntax

The syntax for the OSREQ TSO/E command follows:

The OSREQ TSO/E command syntax



The OSREQ TSO/E command processor requires three positional parameters:

1. The first positional parameter is the *function* to be performed.
2. The second positional parameter is the *collection-name*.
3. The third positional parameter is the *object-name*.

All three positional parameters are required and must be supplied in the order specified. If any of the positional parameters are missing, the system prompts the user for the missing parameter.

The first positional parameter following the OSREQ command name is the OSREQ function to be performed. Valid OSREQ functions are described in Table 39 on page 244, and each one is discussed in detail further in this section.

Table 39. OSREQ TSO command processor functions

Function	**Description**
CHANGE	The OSREQ CHANGE macro is started to change one or more of the following associated with specified object: management class, retention period, storage class, deletion-hold status, event-based retention status.
COMPARE **COM**	The OSREQ RETRIEVE macro is started to retrieve both the primary object and the backup copy. The primary copy data is then compared with the backup data to ensure the data is the same.

Table 39. OSREQ TSO command processor functions (continued)	
Function	**Description**
DELETE	The OSREQ DELETE macro is started to delete the specified object.
QUERY	The OSREQ QUERY macro is started for the specified collection name and object name.
RETRIEVE **RET**	The OSREQ RETRIEVE macro is started to retrieve the specified object.
STORE **ST**	The OSREQ STORE macro is started to store an object with the specified collection name and object name.

The following table identifies the optional keywords that can be used with the OSREQ TSO/E Command Processor for each function. Refer to the OAM Application Programmer's Reference for more detail about each keyword.

Table 40. OSREQ TSO/E command processor optional keywords		
Keyword	**Comment**	**Functions supported**
BUFFER64	Use 64-bit addressable virtual storage above the 2G "bar" for the object data buffer	RETRIEVE STORE
COMPARE **COM**	Checks portion of an object that is retrieved for a predefined pattern of data.	RETRIEVE
DATA	Displays the actual data comprising the object in DUMP notation.	COMPARE RETRIEVE
DB2ID	The DB2 subsystem ID (for a single OAM) or DB2 group attachment name (for an OAMplex) that is configured for an OAM subsystem and optionally associated OAM address space to be used to process this request.	CHANGE, COMPARE DELETE RETRIEVE STORE QUERY
DELHOLD **DH**	Object is placed in/released from deletion-hold mode. Valid values are: HOLD NOHOLD	CHANGE STORE
EVENTEXP **EEX**	Specifies the number of days that are elapsed before this event-based-retention object expires. Valid values are: 0 – 93000	CHANGE
LENGTH	Length of object.	COMPARE, RETRIEVE STORE
MANAGEMENTCLASS **MC**	The SMS Management Class for the object.	CHANGE, STORE

Table 40. OSREQ TSO/E command processor optional keywords (continued)

Keyword	Comment	Functions supported
NOCOMPARE **NC**	Suppresses check of an object that is retrieved for a predefined pattern of data.	RETRIEVE
NODATA	Suppresses the display of object data.	COMPARE, RETRIEVE
OFFSET	Offset of object.	COMPARE, RETRIEVE
RETENTIONPERIOD **RP**	The retention period for the object. Equates to RETPD keyword on OSREQ macro. Valid values are: X'FFFFFFFF' (-1) X'FFFFFFFE' (-2) 0-93000 X'7FFFFFFF' (2 147 483 647)	CHANGE, STORE
RECALL **RCL**	Requests a temporary copy of object to be written to DB2 DASD and retained there for the specified number of days. Valid values are: 0 - 255	RETRIEVE
RETCODE2 **RC2**	For a RETRIEVE, request an additional return code be provided to indicate if this RETRIEVE request resulted in the scheduling of a RECALL of the object to the DB2 sublevel or file system sublevel. For a STORE, indicates whether the STORE request results in scheduling an immediate backup copy for this object	RETRIEVE STORE
STIMEOUT **STO**	Timeout value to use when storing objects to tape that are greater than 256 MB. Valid values are: 0 - 9999	STORE
STORAGECLASS **SC**	The SMS Storage Class for the object.	CHANGE, STORE
STORESEQUENCE	Indicates that Store Sequence processing (STOREBEG, STOREPRT, and STOREEND) should be used even if the object size is 256 megabytes or less. If STORESEQUENCE is not specified, Store Sequence processing will only be used if the object size exceeds 256 megabytes. STORESEQUENCE is not valid if LENGTH is 50 megabytes or less or if the object is being written to an optical device.	STORE
VIEW(BACKUP) **V(BU)**	The first backup copy of the object is retrieved.	RETRIEVE
VIEW(BACKUP2) **V(B2)**	The second backup copy of the object is retrieved.	RETRIEVE
VIEW(PRIMARY) **V**	The primary copy of the object is retrieved.	RETRIEVE

CHANGE

An OSREQ CHANGE command results in an OSREQ CHANGE macro invocation to change the management class, retention period, storage class, deletion-hold status and event-based-retention status associated with the specified object. The following optional keywords are valid for an OSREQ CHANGE command:

- DB2ID
- DELHOLD
- EVENTEXP
- MANAGEMENTCLASS
- RETENTIONPERIOD
- STORAGECLASS

Although there are no *required* keywords for an OSREQ CHANGE command, if none of the optional keywords are specified, the OSREQ CHANGE macro that is issued by the OSREQ TSO/E command processor has no effect on any of these keyword attributes currently associated with the object.

New attributes are assigned to the object dependent upon the attributes that are indicated on the keywords that are associated with the OSREQ CHANGE command.

COMPARE

An OSREQ COMPARE command results in the issuance of the OSREQ RETRIEVE macro to retrieve the primary and backup copies of the specified object. The data for the primary object then is compared with the data from the backup copy.

There are no required keywords on an OSREQ COMPARE command.

The following optional keywords are valid on an OSREQ COMPARE command:

- DB2ID
- DATA or NODATA
- LENGTH
- OFFSET

The DATA keyword causes the DUMP notation display of the actual data comprising the object. Each line of object data consists of a message ID, offset within the object, and 16 bytes of object data in both hexadecimal notation and EBCDIC format.

The LENGTH keyword specifies the length of the object, or portion thereof, for comparison. If the LENGTH keyword is omitted on an OSREQ COMPARE command, the OSREQ command processor issues an OSREQ QUERY macro for the specified object to obtain the length of the object. The portion of the object that is compared is also affected by the OFFSET keyword. If both the LENGTH and the OFFSET keywords are omitted, the entire object is compared using the length that is returned from the OSREQ QUERY macro. If the LENGTH keyword is supplied and the OFFSET keyword is omitted, the number of bytes specified with the LENGTH keyword starting at offset zero is compared.

The OFFSET keyword specifies the offset of the first byte of the object to be compared by the OSREQ COMPARE command. The first byte of an object has an offset of zero (0). The second byte of an object has an offset of one (1), and so on. If the OFFSET keyword is omitted from an OSREQ COMPARE command, the portion of the object that is compared is the portion starting with the first byte of the object (OFFSET=0). The number of bytes compared is specified with the LENGTH keyword.

Tip: The compare function of the OSREQ command is different from the compare keyword on the retrieve function.

DB2ID

Specifies the DB2 subsystem ID (for a single OAM) or DB2 group attachment name (for an OAMplex) that an OAM subsystem (and optionally a started OAM address space) has been configured to use. The *id* can be 1 - 4 characters. In a multiple OAM configuration, the DB2ID keyword is required for all OSREQ TSO/E Command Processor functions.

DELETE

An OSREQ DELETE command results in an OSREQ DELETE macro invocation to delete a specified object. There are no required keywords for this command. The DB2ID optional keyword is valid on this command.

QUERY

An OSREQ QUERY command results in an OSREQ QUERY macro invocation for the specified collection name and object name. There are no required keywords for this command. The DB2ID optional keyword is valid on this command.

You might specify the object name for the OSREQ QUERY command as a generic object name by specifying an asterisk (*) for the low-level qualifier of the object name. In the case where the low-level qualifier of the object name is an asterisk, the OSREQ QUERY macro that is issued is a "generic query" that might result in the directory information of multiple objects being displayed. In the case of a generic query, the query buffer that is obtained by the OSREQ TSO/E command processor is sufficient to hold the directory information for 10 000 objects. If there are more than 10 000 objects, only the directory information for the first 10 000 objects is listed, and the OSREQ TSO/E command processor ends with a return code 4.

The following OAM directory information is listed for each object through the OSREQ TSO/E command processor:

- Collection name
- Object name
- Object size
- Creation date
- Creation timestamp
- Last reference date
- Expiration date
- Management class
- Storage class
- Retrieve Keys and Estimated Retrieve Time
- Object location
- Pending action date
- Object status flags for event-based-retention, deletion-hold status, and retention-protection
- Retention protect date (meaningful only for a retention-protected object)
- Deletion-protection status

RETRIEVE

An OSREQ RETRIEVE command results in an OSREQ RETRIEVE macro invocation to retrieve specified objects. There are no required keywords on this command.

The following optional keywords are valid on an OSREQ RETRIEVE command:

- BUFFER64
- DB2ID
- DATA or NODATA
- COMPARE or NOCOMPARE
- LENGTH
- OFFSET
- RECALL
- RETCODE2
- VIEW(PRIMARY | BACKUP | BACKUP2)

The BUFFER64 keyword specifies that 64-bit virtual storage above the 2G "bar" should be used for the OSREQ TSO/E command processor request. Use of the BUFFER64 keyword requires a MEMLIMIT specification to provide storage above the 2G "bar" for your TSO/E session. For more information on MEMLIMIT see *z/OS Migration* and *z/OS MVS Programming: Extended Addressability Guide*.

The COMPARE keyword checks the portion of the object that is retrieved for a predefined pattern of data. The predefined pattern is a combination of the collection name, object name, and the 4-byte binary counter. If the portion of the object that is retrieved contains the expected data, a comparison successful message (CBR0420I) is issued. If the portion of the object that is retrieved does not contain the expected data, a comparison unsuccessful message (CBR0421I) is issued and the OSREQ TSO/E command processor ends with a nonzero return code. The NOCOMPARE keyword suppresses the checking of the predefined pattern. COMPARE is the default if no keyword is specified with this command.

The DATA keyword displays the actual data comprising the object in a DUMP notation. Each line of object data consists of a message ID, offset within object, 16-bytes of object data is both hexadecimal notation and EBCDIC format. The NODATA keyword suppresses the display of object data on the OSREQ RETRIEVE command. NODATA is the default if no keyword is specified on this command.

The LENGTH keyword specifies the length of the object, or portion of the object to be retrieved. If the LENGTH keyword is omitted on the OSREQ RETRIEVE command, the OSREQ TSO/E Command Processor issues an OSREQ QUERY macro for the specified object to obtain the length of the object. The portion of the object that is retrieved is also affected by the OFFSET keyword. If both the LENGTH and the OFFSET keywords are omitted from the command, the entire object is retrieved using the length that is provided by the OSREQ QUERY macro. If the LENGTH keyword is supplied and the OFFSET keyword is omitted, the number of bytes specified with the LENGTH keyword, starting at offset zero, is retrieved.

Note: When retrieving an object greater than 50 MB (52428800 bytes) and less than or equal to 256 MB (268435456 bytes), if the first data buffer supplied is not large enough to contain the requested or partial object, the request fails with OSREQ return code 8, reason code 2403080B.

The OFFSET keyword specifies the offset of the portion of the object that is retrieved by the OSREQ RETRIEVE command. The first byte of an object has an offset of zero. The second byte of an object has an offset of one and so on. If the OFFSET keyword is omitted from the command, the portion on the object that is retrieved is the portion starting with the first byte of the object (OFFSET=0). The number of bytes retrieved is specified with the LENGTH keyword.

The RECALL keyword specifies an explicit request for a temporary copy of this object is to be written to the DB2 sublevel or file system sublevel and retained there for the specified number of days. The value that is specified as the number of days is a full word ranging 0 – 255. An invalid value results in failure of the RETRIEVE request. Regardless whether the RETRIEVE request is for a full or partial object, the RECALL keyword always results in a copy of the full object to the DB2 sublevel or file system sublevel.

The RETCODE2 keyword requests that an additional return code be provided to indicate whether this RETRIEVE request resulted in the scheduling of a RECALL of the object to the DB2 sublevel or file system sublevel. This keyword is an optional parameter. This value that is provided for this additional return code is not valid when the RETRIEVE is unsuccessful; when the RETRIEVE is successful, it returns the information shown in Table 42 on page 252.

Table 41. RETCODE2 values for OSREQ RETRIEVE

RETCODE2	Field Definition	Meaning
0	OSM_RECALL_OK_EXP	Either: RECALL not specified with RETRIEVE; no attempt to schedule RECALL or RECALL specified with RETRIEVE and successfully scheduled
4	OSM_RECALL_OK_IMP	RECALL not specified with RETRIEVE, but RECALL successfully scheduled due to CBROAMxx PARMLIB member specifications
8	OSM_RECALL_NO_OSMC	An attempt to schedule a RECALL was not successful because OSMC=NO was specified on OAM started procedure
10	OSM_RECALL_MRT	An attempt to schedule a RECALL was not successful because MAXRECALLTASKS(0) was specified in CBROAMxx PARMLIB
12	OSM_RECALL_RO	An attempt to schedule a RECALL was not successful because RECALLOFF(ON) was specified in CBROAMxx PARMLIB
14	OSM_RECALL_ERR	An attempt to schedule a RECALL was not successful because of a scheduling error
16	OSM_RECALL_XCF_NS	An attempt to schedule a RECALL was not successful because the RETRIEVE was performed on a previous OAMplex member that does not support RECALL processing

The VIEW keyword is valid only on an OSREQ RETRIEVE request and is ignored on all other requests. If you specify VIEW(PRIMARY) on the RETRIEVE request, the primary copy of the object is retrieved. If the primary copy of the object is on an unreadable tape or optical volume, and the automatic access to backup facility is activated, OAM then retrieves the backup copy of the object. If you specify VIEW(BACKUP) on the RETRIEVE request, then OAM retrieves the first backup copy of the object. If you specify VIEW(BACKUP2) on the RETRIEVE request, then OAM retrieves the second backup copy of the object. If you omit the VIEW keyword, then OAM uses VIEW(PRIMARY) as the default for the request then retrieves the primary copy of the object.

STORE

An OSREQ STORE command results in either:

- An OSREQ STORE macro invocation, for:
  - Objects whose length is equal to or less than 256 megabytes and STORESEQUENCE is not specified
  - Objects of any size if BUFFER64 is specified
  - Objects that are stored on an optical device
- OSREQ STOREBEG, STOREPRT, and STOREEND macro invocations, for:
  - Objects whose length is greater than 256 megabytes and BUFFER64 is not specified.
  - Objects whose length is greater than 50 megabytes and STORESEQUENCE is specified.

The LENGTH keyword is a required keyword on an OSREQ STORE command. The value that is specified with the length keyword is the length, in bytes, of the object to store. If the LENGTH keyword or value is omitted, an error message is issued.

The OSREQ command processor creates an object of the specified length and fills the object with a predefined pattern. The predefined pattern is a combination of the collection name, object name, and the 4-byte binary counter. The binary counter in the data pattern is incremented by one for each replication of the data pattern within the object.

The following keywords are valid on an OSREQ STORE command:

- BUFFER64 (optional)
- DB2ID
- DELHOLD (optional)
- LENGTH (required)
- MANAGEMENTCLASS (optional)
- RETCODE2 (optional)
- RETENTIONPERIOD (optional)
- STIMEOUT (optional)
- STORAGECLASS (optional)
- STORESEQUENCE (optional)

New attributes are assigned to the object dependent upon the attributes that are indicated on the keywords that are associated with the OSREQ STORE command. If these attributes are not specified, the defaults for the collection or the ACS overrides are assigned.

The second positional parameter following the OSREQ command name is the *collection name*. The collection name must be a fully qualified OAM collection name for each OSREQ function.

The third positional parameter to follow the OSREQ command name is the *object-name*. The object name should be a fully qualified object name.

Note: When storing an object greater than 50 MB (52428800 bytes), if multiple data buffers are supplied but they are not in contiguous storage, the request fails with OSREQ return code 8, reason code 2402080A.

The BUFFER64 keyword specifies that 64-bit virtual storage above the 2G "bar" should be used for the OSREQ TSO/E command processor request. On a STORE, BUFFER64 is mutually exclusive with STORESEQUENCE. Use of the BUFFER64 keyword requires a MEMLIMIT specification to provide storage above the 2G "bar" for your TSO/E session. For more information on MEMLIMIT see *z/OS Migration* and *z/OS MVS Programming: Extended Addressability Guide*.

RETCODE2 is an optional keyword that indicates whether the STORE request results in scheduling an immediate backup copy for this object. *return_code2_word* is valid only when the STORE is successful, and the system provides the following information:

Table 42. RETCODE2 values for OSREQ STORE

RETCODE2	Field Definition	Meaning
0	OSM_IMBKP_OK	Immediate backup copy request successfully scheduled.
4	OSM_IMBKP_NOT_RQD	Immediate backup copy request not required.
8	OSM_IMBKP_NO_OSMC	An attempt to schedule an immediate backup for this object is not successful because OSMC is not up and running.
14	OSM_IMBKP_ERR	An attempt to schedule an immediate backup for this object is not successful because of unexpected scheduling error.
16	OSM_IMBKP_STEND	Immediate backup to optical not supported for STOREEND.

Note: The remaining keywords and parameters are described in the *z/OS DFSMS OAM Application Programmer's Reference*

OSREQ TSO/E command processor return codes

The OSREQ TSO/E command processor returns to the TSO/E terminal monitor program (TMP) with a return code in register 15. This return code can be tested using the &LASTCC variable in a TSO/E CLIST. In all cases, except one, the return code in register 15 following the OSREQ command is the return code that was returned by the OSREQ macro in register 15. A return code of 20 indicates that the OSREQ TSO/E command processor encountered an error unrelated to the OSREQ macro which it invokes.

The following return codes are returned from the OSREQ TSO/E command processor:

Return code
Description

0
The requested OSREQ function successfully completed.

4
The requested OSREQ function was completed with a warning condition.

8
The requested OSREQ function was not completed due to an application programming error.

12
The requested OSREQ function was not completed due to an environmental error.

20
The OSREQ TSO/E command processor encountered an error during its processing. The following errors will cause a return code of 20:

Error
Description

1
A nonzero return code was received from the TSO/E parse service routine (IKJPARS) in register 15. Error message CBR0402I is issued.

2
A nonzero return code was received from the STORAGE OBTAIN macro when storage was requested for a data buffer for an OSREQ QUERY operation. Error message CBR0403I is issued.

3
: The LENGTH keyword was not specified on an OSREQ STORE request. Error message CBR0406I is issued.

4
: A nonzero return code was received from the STORAGE OBTAIN macro when storage was requested for a data buffer for an OSREQ STORE operation. Error message CBR0403I is issued.

5
: A nonzero return code was received from the STORAGE OBTAIN macro when storage was requested for a data buffer for an OSREQ RETRIEVE operation. Error message CBR0403I is issued.

6
: An invalid length was specified on an OSREQ RETRIEVE or COMPARE command. If the DATA option was specified, the length must be 268 435 456 or less. Error message CBR0407I is issued.

7
: An invalid offset was specified on an OSREQ RETRIEVE or COMPARE command. Error message CBR0408I is issued.

8
: An OSREQ RETRIEVE command was issued with the COMPARE keyword and the data comparison did not match. Or an OSREQ COMPARE command was issued and the data comparison did not match. Error message CBR0421I is issued.

9
: Either the OFFSET keyword, the LENGTH keyword, or both the OFFSET and LENGTH keywords were not specified. Since the DATA option was specified and the object's size is greater than 268 435 456, both the OFFSET and LENGTH keywords are required in order to perform the OSREQ RETRIEVE or COMPARE command. Error message CBR0442I is issued.

10
: An OSREQ STORE command was issued with both the STORESEQUENCE and BUFFER64 keywords specified. Error message CBR0450I is issued.

11
: A nonzero return code was issued from the IARV64 macro when storage was requested for a 64-bit addressable virtual storage buffer above the 2G “bar”. Error message CBR0451I is issued.

12
: The DB2ID keyword was not specified, but is required in a multiple OAM configuration. Error message CBR0453I is issued.

3995 optical service information messages

The 3995 optical library microcode provides a 3995 Service Information Message (SIM) to the host when a component within the 3995 optical library needs service.

If the appropriate bit is on in the SIM, OAM issues an action message (CBR3309E) to the MVS console. Additionally, if the appropriate bit is on in the SIM, OAM logs the SIM message as a type x‘A3’ Asynchronous Notification Record (ANR) in SYS1.LOGREC.

The following restrictions apply:

- The 3995 optical library dataservers must be defined in the IODF as 3995 devices (not CTC or 3088 devices) in order for the Environmental Recording Editing and Printing program (EREP) to produce formatted reports for any ANRs generated for those libraries.
- AMRF must be active in order to recall CBR3309E messages that have rolled off the screen. To display the message identification numbers and text of all immediate action and eventual action messages, as well as OAM issued messages awaiting replies, use the following command:

  ```
  DISPLAY R,L,KEY=OAM
  ```
- OAM must be active when SIM attentions from the 3995 are sent to the host or they are lost.

- The optical libraries indicated in the SIM attention messages must be known to OAM, and they must be part of the active SMS configuration or they are lost.

Chapter 5. Operating OAM and OTIS address spaces and OSMC functions

This topic helps you use operator commands and describes the tasks to operate the OAM and OTIS address spaces (necessary for using the OSREQ functions) and OSMC functions.

Overview of operator tasks

You can perform the following operator tasks:

Subtask	Associated procedure (see . . .)
Start: • OTIS • OAM • OSMC storage management cycle • Library management cycle • DASD space management cycle • Automatic access to backup • OAM Volume Recovery utility • Single Object Recovery utility • Move Volume utility • Tape Recycle • Diagnostic messages	See the following topics: • "Starting OTIS" on page 259 • "Starting OAM" on page 259 • "Starting OSMC functions" on page 263 • "Starting the library space management cycle" on page 268 • "Starting the DASD space management cycle for an individual storage group" on page 270 • "Starting automatic access to backup copies of objects" on page 287 • "Starting the OAM Volume Recovery utility" on page 270 • "Starting object recovery for single objects" on page 276 • "Starting the Move Volume utility" on page 278 • "Starting the tape recycle utility" on page 278 • "Starting temporary diagnostic messages for file system errors originating from an OSREQ request" on page 286
Vary optical drives and optical libraries online and offline.	"Varying optical drives and libraries" on page 290
Enter unlabeled and labeled optical disks into an optical library.	"Entering an optical disk into an optical library" on page 292
Eject an optical disk from an optical library.	"Ejecting an optical disk" on page 296
Mount an optical disk on an operator-accessible optical drive.	"Mounting an optical disk on an operator-accessible drive" on page 298
Demount an optical disk on an operator-accessible optical drive.	"Demounting and removing an optical disk cartridge from an operator-accessible drive" on page 298
Label an optical disk on an operator-accessible optical drive.	• "Labeling an optical disk on a 3995 operator-accessible drive" on page 299
Relabel an optical disk volume.	"Relabeling a 3995 optical disk volume" on page 302
Display OAM instance configuration	"Displaying OAM configuration information" on page 313
Remove subsystems from the OAM configuration	"Removing one or all OAM subsystems from the OAM configuration" on page 377

Subtask	Associated procedure (see . . .)
Refresh OTIS DB2 Tasks	"Refreshing OTIS DB2 Tasks" on page 377
Display status for an OAM instance of: • OAM • OAMXCF • OSMC storage management cycle • Optical drive • Optical library • Optical and tape volumes • Lost volumes • Storage group • SETOAM, SETOPT, and SETOSMC statements • OAM messages	See the following topics: • "Displaying OAM Status" on page 306 • "Displaying OAM XCF status" on page 314 • "Displaying OSMC summary status" on page 316 • "Displaying drive online/offline connectivity" on page 322 • "Displaying library online/offline connectivity" on page 327 • "Displaying Storage Group Status" on page 335 • "Displaying volume status" on page 344 • "Displaying volumes that have LOSTFLAG set" on page 352 • "Displaying SETDISK, SETOAM, SETOPT, SETOSMC, and SETTLIB parameters" on page 353 • "Displaying outstanding OAM messages" on page 357
Query active and pending optical and tape requests.	"Querying summary and detail information for pending and active requests" on page 357
Dump the OAM address space.	"Scheduling an SVC dump for the OAM address space" on page 363
Restart OAM.	"Restarting the OAM address space" on page 364
Update SETOAM, SETOPT, SETOSMC, and OAMXCF parameter values.	• "Using the UPDATE command to set SETOAM, SETOSMC, and SETOPT values" on page 366 • "Using the UPDATE command to set OAMXCF values" on page 373
Update fields in the DB2 VOLUME or TAPEVOL table.	"Updating fields in the DB2 Volume Table and the Tape Volume Table" on page 374
Audit volumes.	"Auditing a volume" on page 376
Remap a 3995 library.	"Remapping an optical library" on page 377
Stop: • OAM • OSMC • Move Volume utility • Access backup • Volume recovery • OTIS • Tape Recycle utility • Diagnostic messages	See the following topics: • "Stopping OAM" on page 379 • "Stopping OSMC" on page 379 • "Stopping the Move Volume utility" on page 380 • "Stopping automatic access to backup" on page 381 • "Stopping a volume recovery that is in progress" on page 380 • "Stopping OTIS" on page 383 • "Stopping OAM functions" on page 378 • "Stopping display of diagnostic messages for file system related errors originating from an OSREQ request" on page 383

Message format conventions

The following conventions are used to show message format:

```
CBRnnnnX Message_text
```

where:

CBR
Standard OAM message prefix

nnnn
Four-digit message number

X
Type code

A
Action required

D
Decision needed

E
Eventual action required

I
Information only

Message_text
Text of the message

The following is a sample of an OAM message:

```
CBR2601A Specify shelf location for volume volser.
```

Note: In message text, italicized words indicate a value supplied by the system.

For a description of messages, see *MVS System Messages*.

Overview of operator commands

Many commands use the MVS operator MODIFY command. Throughout this topic, the command syntax is:

```
F OAM,…
```

The command F is the abbreviation of the MVS MODIFY command.

The OAM address space can be defined in one of the following ways:

- As the name of the cataloged procedure in SYS1.PROCLIB that you use to start the OAM address space:

  ```
  START OAM,…
  START OAMA,…
  START OAMB,…
  ```

- As the task ID assigned in the address space START command:

  ```
  START procname.OAM,…
  START OAM.OAMA,…
  START OAM.OAMB,…
  ```

In a multiple OAM configuration, it is recommended that a naming convention be established to distinguish one OAM Object address space from another using a prefix of “OAM” and a lettered suffix (OAMA, OAMB). This helps to differentiate OAM subsystems (recommended to be named using a prefix of “OAM” and a numbered suffix) from OAM address spaces. This naming convention can also ease operations:

- Commands can be directed to a specific OAM address space:

  ```
  F OAMA,…
  F OAMB,…
  ```

- Commands can also be directed to all OAM address spaces using wildcard characters:

```
F OAM*,…
```

Tip: If your system programmer chooses to use a name other than *OAM*, use that name in the place of *OAM*.

Two SMS operator commands are related to the OAM address space:

- **DISPLAY SMS**—for determining the status of OAM, OSMC, storage groups, optical volumes, optical disk drives, and optical libraries. See "Displaying status" on page 306 for specific information on the DISPLAY command.
- **VARY SMS**—for varying optical disk drives or optical libraries online or offline. See "Varying optical drives and libraries" on page 290 for specific information on the VARY command.

The following operator commands for the OAM address space are based on the MVS MODIFY command:

- **AUDIT**—for auditing library resident volumes. See "Auditing a volume" on page 376 for specific information on the AUDIT command.
- **DISPLAY**—for displaying the current SETDISK, SETOAM, SETOPT, and SETOSMC settings. See "Displaying SETDISK, SETOAM, SETOPT, SETOSMC, and SETTLIB parameters" on page 353 for specific information on the DISPLAY command.
- **DUMP**—for capturing data used for diagnostic purposes. See "Scheduling an SVC dump for the OAM address space" on page 363 for specific information on the DUMP command.
- **LABEL**—for labeling the two volumes of an optical disk on an operator-accessible optical drive. For specific information on the LABEL command, see the following topic:
  - "Labeling an optical disk on a 3995 operator-accessible drive" on page 299
- **QUERY**—for displaying summary, detail, or both types of information regarding active and pending tape and optical requests. See "Querying summary and detail information for pending and active requests" on page 357 for specific information on the QUERY command.
- **RELABEL**—for changing the volume serial number for an existing optical disk volume. See "Relabeling a 3995 optical disk volume" on page 302 for specific information on the RELABEL command.
- **REMAP**—for initiating a REMAP on a 3995 optical library. See "Remapping an optical library" on page 377 for specific information on the REMAP command.
- **RESTART**—for restarting the OAM address space without issuing a STOP and START command. See "Restarting the OAM address space" on page 364 for specific information on the RESTART command.
- **START**—for starting an OSMC storage management cycle, an OSMC library space management cycle, an OSMC DASD space management cycle, an OSMC Volume Recovery utility, an OSMC Single Object Recovery utility, an OSMC Move Volume utility, an OSMC Tape Recycle command, or the automatic access to backup function. For specific information on the START command, see the following topics:
  - "Starting OSMC functions" on page 263
  - "Starting automatic access to backup copies of objects" on page 287
- **STOP**—for stopping OAM, all OSMC processing, OSMC processing for an individual storage group, an OSMC Move Volume utility, an OSMC Volume Recovery Utility, an OSMC Tape Recycle command process, or the Automatic Access to Backup recovery function. See "Stopping OAM functions" on page 378 for specific information on the STOP command.
- **UPDATE**—for updating fields in the DB2 Volume Table and the Tape Volume Table with the F OAM command. This command also dynamically updates the SETOAM, SETOPT, SETOSMC, and OAMXCF settings. For specific information on the UPDATE command, see the following topics:
  - "Using the UPDATE command to set SETOAM, SETOSMC, and SETOPT values" on page 366
  - "Using the UPDATE command to set OAMXCF values" on page 373
  - "Updating fields in the DB2 Volume Table and the Tape Volume Table" on page 374

The following operator commands for the OAM address space are based on the MVS LIBRARY command:

- **EJECT**—for removing an optical disk from an optical library. See "Ejecting an optical disk" on page 296 for specific information on the EJECT command.
- **RESET**—for reenabling the CBRUXSAE user authorization installation exit after it has been disabled or bypassed.

Two MVS operator commands are related to the OTIS address space:

- **START**—for starting or restarting OTIS. See "Starting OTIS" on page 259 for specific information on this use of the START command.
- **STOP**—for stopping OTIS. You must use the MVS MODIFY command when stopping OTIS. See "Stopping OTIS" on page 383 for specific information on this use of the STOP command.

Starting OTIS

The installation of OAM creates an address space called OAM thread isolation support (OTIS). This required address space starts automatically during system IPL.

Requirement: The OTIS address space must be active when any OAM applications are processing objects.

To start the OTIS address space or restart the address space after it has ended, enter the MVS START command. The following MVS command syntax starts the OTIS address space:

```
►► START ── OTIS ─►◄
```

While OTIS is initializing, the system issues the following message:

```
CBR8500I OTIS subsystem is initializing.
```

If your DB2 subsystem is active, the system issues the following message:

```
CBR8571I OTIS subsystem successfully connected to DB2 subsystem.
```

If your DB2 subsystem is *not* active, the system issues the following message:

```
CBR8572I OTIS subsystem unable to connect to DB2 subsystem because DB2
         subsystem is not active.
```

After either of the above messages are issued, the system issues the following message:

```
CBR8501I OTIS subsystem initialization complete.
```

You should not see any messages other than those listed. If any other messages are issued, see *z/OS MVS System Messages, Vol 4 (CBD-DMO)*.

For further information on the MVS START command, see *z/OS MVS System Commands*

Starting OAM

Before starting an OAM address space for Object processing, you must initialize the DB2 subsystem that is associated with the OAM Object address space.

Tip: The system programmer can update the SMS entry in the IGDSMS*xx* member of PARMLIB to automatically start OAM during MVS IPL.

To start the OAM address space manually, or to restart the OAM address space after it has terminated, enter the MVS START command. The following syntax of the MVS START command starts an OAM address space:

```
>>-- START --+-- OAM ------+--+---------------+--+--------------------+-><
  '-- S ----'  '-- procname -'  '-- .identifier --'  '-- ,parameter=value -'
```

OAM
: Name of the IBM-supplied cataloged procedure that starts OAM.

procname
: Name of the user-written cataloged procedure that starts OAM.

identifier
: User-determined name that identifies the OAM address space. If you do not specify an identifier, the system automatically uses the *procname* as the identifier.

Recommendation: You can include any parameter that was defined in the OAM procedure JCL (PROCLIB member OAM, OAMA, etc.) in this command to initialize or override the parameter upon activation of the OAM address space. An example of modifying these parameters follows:

```
START OAM,OAM=XX,OSMC=YES,REST=NO,MAXS=10,EJECT=LRM
```

Where *xx* is the CBROAM*xx* member of PARMLIB that you prefer to have OAM use during initialization.

In a multiple OAM configuration, it is recommended that you use multiple members of PROCLIB each with a different name (OAMA, OAMB) and containing the parameters specific to each instance of OAM. A single member of PROCLIB can be used and started with a unique task identifier for each address space (S OAM.OAMA and S OAM.OAMB) but because unique values for the D= keyword are required for each address space, this approach requires specifying the D= value on the START command.

For details on these parameters, see the discussion concerning modifying and running the CBRAPROC SAMPLIB member in "5h Updating the PROCLIB" on page 146.

Note: REUSASID=YES must not be specified when starting OAM for object support.

When initializing OAM with OSMC, the system issues the following messages:

```
CBR0001I OAM initialization starting.

CBR0016I Successful processing of the OAMXCF commands in CBROAMxx member of PARMLIB. Initialization continues.

CBR0070I OAM XCF member xcf-member-name is the first member defined to
         OAM XCF group xcf-group-name, group successfully defined to
         XCF and member created.

CBR0016I Successful processing of the SETOAM commands in CBROAMxx member of PARMLIB. Initialization continues.

CBR0016I Successful processing of the SETOPT commands in CBROAMxx member of PARMLIB. Initialization continues.

CBR0016I Successful processing of the SETOSMC commands in CBROAMxx member of PARMLIB. Initialization
continues.

CBR0016I Successful processing of the SETTLIB commands in CBROAMxx
member of PARMLIB. Initialization continues.

CBR0016I Successful processing of the SETDISK commands in CBROAMxx
member of PARMLIB. Initialization continues.

CBR0016I Successful processing of the ONLYIF commands in CBROAMxx member
of PARMLIB. Initialization continues.

CBR9000I OSMC initialization starting.

CBR9001I OSMC initialization completed.

CBR0002I OAM initialization completed.
```

Notes about starting OAM:

1. CBR0016I displays only if OAM=*xx* is indicated.
2. If you use a CBROAM*xx* PARMLIB member to initialize OAM, OAM displays CBR03*xx*I messages that might be normal for your environment.
3. You must restart OAM to recognize any changes that were made to the CBROAM*xx* member of PARMLIB. Additionally, to dynamically change the SETOAM, SETOPT, or SETOSMC parameters, use the

UPDATE command. See "Using the UPDATE command to set SETOAM, SETOSMC, and SETOPT values" on page 366 for more information.

When initializing OAM without OSMC, the system issues the same messages as when initializing with OSMC, except for messages CBR9000I and CBR9001I.

For further information on these messages, see *z/OS MVS System Messages, Vol 4 (CBD-DMO)*.

Related reading:

- For further information on the MVS START command, see the *z/OS MVS System Commands*.
- For explanations and system actions for these messages, see *z/OS MVS System Messages, Vol 4 (CBD-DMO)*.
- For information on starting DB2, see the "Starting DB2" topic in IMS in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEPH2).

Operator command applicability in a multiple OAM configuration

This section contains new information on OAM command applicability to a Tape Library or Object address space in a multiple OAM configuration.

Some OAM commands are only applicable to a specific type of OAM address space (Object or Tape Library) in a multiple OAM configuration. The following table indicates whether an OAM command is applicable to each type of OAM address space.

OAM Command	Object OAM Address Space	Tape Library OAM Address Space
F *OAM*,AUDIT,VOLUME	N[1]	Y
F *OAM*,AUDIT,VOLLIST	N[1]	Y
F *OAM*,AUDIT,LIBRARY	N[1]	Y
F *OAM*,DISPLAY,DRIVE	N[1]	N[3]
F *OAM*,DISPLAY,GROUP	Y	Y
F *OAM*,DISPLAY,LIB	N[1]	Y
F *OAM*,DISPLAY,LOSTVOL	Y	N
F *OAM*,DISPLAY,OAM	Y	Y
F *OAM*,DISPLAY,OAMXCF	Y	N
F *OAM*,DISPLAY,OSMC	Y	N
F *OAM*,DISPLAY,SETDISK	Y	N
F *OAM*,DISPLAY,SETOAM	Y	N
F *OAM*,DISPLAY,SETOPT	Y	N
F *OAM*,DISPLAY,SETOSMC	Y	N
F *OAM*,DISPLAY,SETTLIB	N	Y
F *OAM*,DISPLAY,VOL	Y	Y
F *OAM*,DUMP	Y	Y
F *OAM*,EJECT	N[1]	N[1]
F *OAM*,LABEL	N[1]	N
F *OAM*,QUERY,ACTIVE	Y	Y

OAM Command	Object OAM Address Space	Tape Library OAM Address Space
F *OAM,*QUERY,WAITING	Y	Y
F *OAM,*RELABEL	N[1]	N
F *OAM,*REMAP	N[1]	N
F *OAM,*RESTART	Y	Y
F *OAM,*START,AB	Y	N
F *OAM,*START,DASDSM	Y	N
F *OAM,*START,DIAGMSGS	Y	N
F *OAM,*START,LIBMGT	N[1]	N
F *OAM,*START,MOVEVOL	Y	N
F *OAM,*START,OBJRECV	Y	N
F *OAM,*START,OSMC	Y	N
F *OAM,*START,RECOVERY	Y	N
F *OAM,*START,RECYCLE	Y	N
F *OAM,*START,STORGRP	Y	N
F *OAM,*STOP,AB	Y	N
F *OAM,*STOP,DIAGMSGS	Y	N
F *OAM,*STOP,MOVEVOL	Y	N
F *OAM,*STOP,OAM	Y	Y
F *OAM,*STOP,OSMC	Y	N
F *OAM,*STOP,RECOVERY	Y	N
F *OAM,*STOP,RECYCLE	Y	N
F *OAM,*STOP,STORGRP	Y	N
F *OAM,*UPDATE,OAMXCF	Y[2]	N
F *OAM,*UPDATE,SETOAM	Y	N
F *OAM,*UPDATE,SETOPT	N[1]	N
F *OAM,*UPDATE,SETOSMC	Y[2]	N
F *OAM,*UPDATE,VOL	Y	N
F *OAM,*VARY,*drive-name*	N[1]	N
F *OAM,*VARY,*library-name*	N[1]	Y

[1] The Optical level of the OAM storage hierarchy is not supported in a multiple OAM configuration.

[2] Valid, but only for keyword specifications that do not involve the Optical level of the OAM storage hierarchy.

[3] To display tape drive status for drives in a tape library use the MVS DISPLAY UNIT command or the MVS LIBRARY DISPDRV command.

If a command is issued, but is not applicable to the type of OAM address space in a multiple OAM configuration, then a message is issued. See CBR1049I and CBR1130I in *z/OS MVS System Messages, Vol 4 (CBD-DMO)*.

MVS LIBRARY command applicability

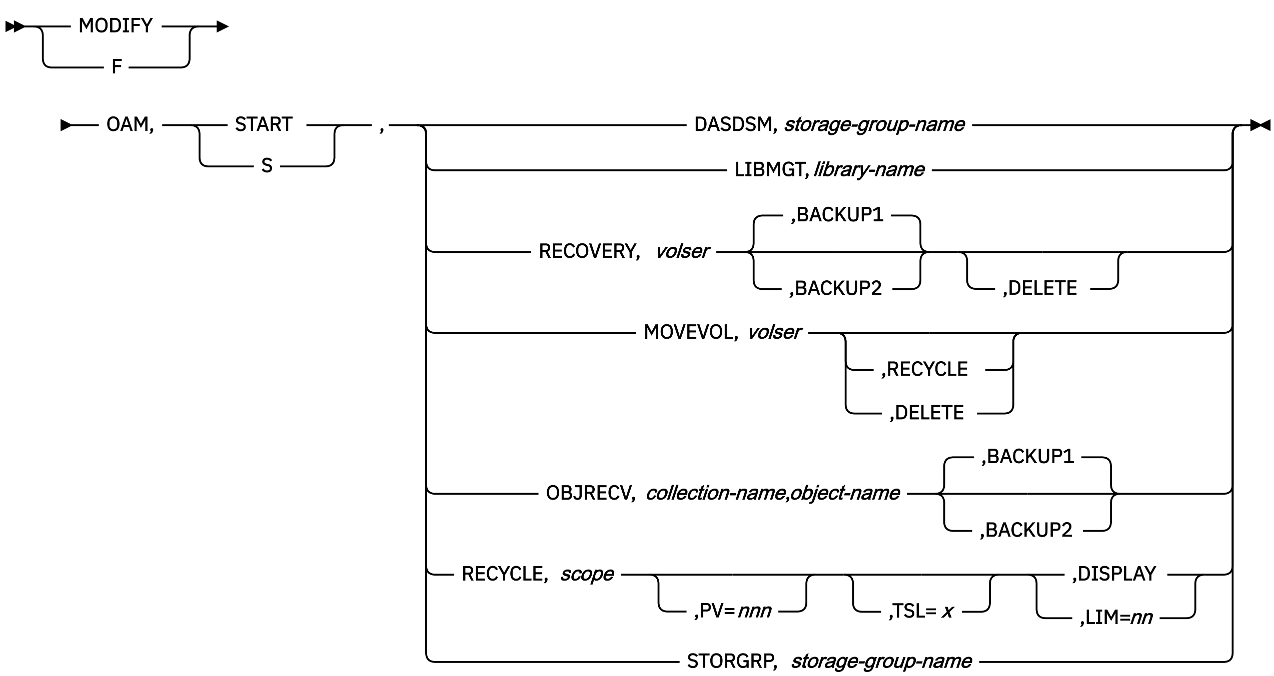
In a classic OAM configuration, the LIBRARY command works in the same way as it did in the prior release. In a multiple OAM configuration, the LIBRARY command, in most cases, is associated only with the Tape Library OAM instance. The only LIBRARY commands that also affect Object OAM instances are LIBRARY RESET,CBRUXSAE and LIBRARY RESET,EDGTVEXT.

Starting OSMC functions

Use the F OAM,START command to start the following OSMC functions:

- Storage management cycle for all or specific Object or Object Backup storage groups
- Library space management cycle for an optical library
- DASD space management cycle for an Object storage group
- Volume Recovery utility
- Single Object Recovery utility
- Move Volume utility
- Tape recycle

The following MVS command syntax starts the OSMC functions:



DASDSM,*storage-group-name*
: Starts an OSMC DASD space management cycle for the specified Object storage group that is named *storage-group-name*.

LIBMGT,*library-name*
: Starts an OSMC library management cycle for an optical library that is named *library-name*. OAM must control the library on the system where the command was entered.

MOVEVOL,*volser*[,RECYCLE | ,DELETE]

Starts the OSMC Move Volume utility for the source volume with volume serial number *volser*. Use the MOVEVOL command to migrate objects from a primary or backup source volume to one or more target volumes, or to delete a scratch volume from the OAM inventory. If you omit the RECYCLE or DELETE option, the migrated volume remains in the same Object or Object Backup storage group, and all volume record fields in the OCDB remain the same.

RECYCLE

Recycles the optical platter or rewritable tape volume after moving the objects off the cartridge. When the MOVEVOL command completes, OAM recycles the tape volume or optical platter using the parameters that are specified in the SETOAM TAPERECYCLEMODE or SETOPT OPTICALREINITMODE statement in the CBROAM*xx* member of PARMLIB.

DELETE

Deletes the optical platter or tape volume after the objects are moved off of the cartridge. After the MOVEVOL command completes and no objects remain on the cartridge, the tape volume or optical platter is deleted from the OAM configuration database. The DELETE option overrides any parameters that are specified in the SETOAM TAPERECYCLEMODE or the SETOPT OPTICALREINITMODE statement in the CBROAM*xx* member of PARMLIB.

When an optical volume is deleted, OAM displays message CBR2153I. When a tape volume is deleted, OAM displays message CBR2165I for a rewritable tape volume, or message CBR2173I for a WORM tape volume. These messages can be used to signal the tape management system that the tape volume can be returned to MVS scratch status for reuse if the tape volume is rewritable or to handle the WORM tape volume according to the installation procedures for expired WORM media. You can automate this message to notify the tape management system to return the tape volume to MVS scratch status.

OBJRECV,*collection-name*,*object-name*,BACKUP1 | BACKUP2

Starts the OSMC Single Object Recovery utility for the object that is named *object-name*, in the collection that is named *collection-name*. When you want to recover a primary copy of an object, and you have specified BACKUP1 on the START command for object recovery, OAM creates the recovery from the first backup copy of the object. If you have specified BACKUP2, OAM creates the recovery from the second backup copy of the object. BACKUP1 is the default if you have specified neither BACKUP1 nor BACKUP2.

RECOVERY,*volser*[,BACKUP1 | ,BACKUP2][,DELETE]

Starts the OSMC Volume Recovery utility for either a tape volume with volume serial number *volser* or an optical platter with volume serial number *volser* and its opposite side.

The RECOVERY command verifies that another instance of OAM in an OAMplex is not controlling the volume that is to be recovered. If another instance of OAM is controlling the volume, OAM displays message CBR1068I and cancels the RECOVERY command.

Volumes that belong to an Object Backup storage group are recovered from the primary copies of the objects. If the volume belongs to an Object Backup storage group, OAM ignores the BACKUP1 and BACKUP2 parameters.

BACKUP1

When you want to recover all objects residing on a volume that belongs to an Object storage group and you have specified BACKUP1 on the START command for Volume Recovery, OAM creates the recovery from the first backup copies of the objects. If the volume or cartridge to be recovered belongs to an Object storage group and you have specified neither BACKUP1 nor BACKUP2, BACKUP1 is the default. Backup volumes are always recovered using the primary copies of the objects.

BACKUP2

When you want to recover all objects residing on a volume that belongs to an Object storage group and you have specified BACKUP2 on the START command for volume recovery, the recovery is made from the second backup copies of the objects. Backup volumes are always recovered using the primary copies of the objects.

DELETE

After the volume recovery is complete and no objects are left on the tape or on either side of the optical platter, the tape volume or optical platter is deleted from the OAM configuration database. If the volume resides in an optical library, it is ejected. The DELETE option overrides any parameters that are specified in the SETOAM TAPERECYCLEMODE or the SETOPT OPTICALREINITMODE statement in the CBROAM*xx* member of PARMLIB.

When an optical volume is deleted, OAM displays message CBR2153I. When a tape volume is deleted, OAM displays message CBR2165I. You can automate this message to notify the tape management system to return the tape volume to MVS scratch status.

RECYCLE,*scope*{,PV=*xxx*}{,TSL=*x*}[,LIM=*yy*|,DISPLAY]

Displays a list of candidate volumes that meet user-defined criteria to the hardcopy log, and in turn, automatically selects recycle candidate volumes and initiates the MOVEVOL with RECYCLE process on those candidates until either the user-specified limit is reached or no more volumes meeting criteria are available.

scope

Indicates one of the following:

- **Specified name of an object or object backup storage group**, indicates that only tape volumes marked full, that belong to the specified object or object backup storage group, are considered candidates for this RECYCLE command.
- **(ALLGRP)** all full tape volumes that belong to all primary object storage groups defined in the ACTIVE SCDS are considered candidates for this RECYCLE command.
- **(ALLBK1)** all full tape volumes that belong to all first backup storage groups defined in the ACTIVE SCDS are considered candidates for this RECYCLE command.
- **(ALLBK2)** all full tape volumes that belong to all second backup storage groups defined in the ACTIVE SCDS are considered candidates for this RECYCLE command.

PV

=*nnn* An optional keyword indicating the valid data threshold to be used in determining whether a volume is a candidate for RECYCLE. Full tape volumes that have a percentage of valid data less than or equal to *nnn* are candidates for RECYCLE. If PV=*nnn* is not specified, the percent valid to be used to determine RECYCLE candidates is derived from the PERCENTVALID default value as defined through the SETOAM command in the CBROAMxx PARMLIB member. Valid values for *nnn* are 0 to 100.

TSL

=*x* An optional parameter that indicates the tape sublevel that a recycle candidate volume must be associated with. The TSL parameter only applies to group volumes. If a scope of (ALLBK1) or (ALLBK2) is specified with the TSL parameter, the command fails. Valid values for *x* are:

- A indicates all group volumes are recycle candidates. This is the default.
- 1 indicates only tape sublevel 1 group volumes are recycle candidates.
- 2 indicates only tape sublevel 2 group volumes are recycle candidates.

DISPLAY

An optional parameter that produces a list of volumes that meet criteria to be recycle candidates. This list is sorted by the percentage of valid data on each volume and is written to hardcopy system log through the CBR9875I message. This option does not initiate Recycle processing, and can be issued at anytime, whether a RECYCLE command is actively processing or not. The list of candidate volumes might be large as it shows all volumes that meet the user-specified criteria for RECYCLE.

If DISPLAY is not specified then LIM=*nn* must be specified.

LIM

=*nn* If the DISPLAY parameter is not specified, this keyword is required to indicate the maximum number of volumes to be selected for RECYCLE processing. Valid values for *nn* are 1 to 40.

If LIM=*nn* is not specified, then DISPLAY must be specified

STORGRP,*storage-group-name*
Starts an OSMC storage management cycle for one or more specified Object or Object Backup storage groups that are named *storage-group-name*, where *storage-group-name* can be:

- The name of a specific Object or Object Backup storage group. For example, PAY2010.
- A name with a single asterisk (‘*’) in the rightmost part of the name. This indicates that the request applies to all Object or Object Backup storage groups whose names match the characters to the left of the asterisk. For example, PAY*, which applies to all Object or Object Backup storage groups whose names start with PAY.
- A single asterisk (‘*’) . which specifies every storage group defined in the ACDS as an Object or Object Backup storage group.

Note:

In an OAMplex, if a specific storage group is processed, OAM initiates OSMC processing of the specified storage group on the system where the command was entered.

Starting the storage management cycle

You can use the OAM START command to start the storage management cycle for all Object storage groups in the active configuration or for an individual Object or Object Backup storage group.

Starting the storage management cycle for all storage groups

There are two slightly different methods for starting the OSMC storage management cycle for all storage groups:

1. The F OAM,START,OSMC command, explained in this topic, which applies only to Object storage groups, and
2. The F OAM,START,STORGRP,* command, explained in “Starting the storage management cycle for a selected set of storage groups” on page 267, which applies to all Object and Object Backup storage groups.

To start the OSMC storage management cycle for all Object (but not Object Backup) storage groups:

1. Enter the following command:

```
F OAM,START,OSMC
```

2. The system issues the following message:

```
CBR1000I OAM START command execution scheduled.
```

3. The system issues the following messages on the console that pertain to the storage management cycle:

```
CBR9018I OSMC starting Storage Management Cycle.

CBR9200I Object Processing starting for storage group xxxxx01.

CBR9370I OSMC Detail for taskname:

Explanation:
            READ     READ     READ     READ
           DISK1    DISK2      OPT     TAPE
WORK Q: aaaaaaaa bbbbbbbb cccccccc dddddddd
WAIT Q:          eeeeeeee ffffffff gggggggg
DONE:   hhhhhhhh iiiiiiii jjjjjjjj kkkkkkkk
           WRITE    WRITE    WRITE    WRITE    WRITE
           DISK1    DISK2      OPT    TAPE1    TAPE2
WORK Q: llllllll mmmmmmmm nnnnnnnn oooooooo pppppppp
WAIT Q:          qqqqqqqq rrrrrrrr ssssssss tttttttt
DONE:   uuuuuuuu vvvvvvvv wwwwwwww xxxxxxxx yyyyyyyy
           WRITE    WRITE      DIR
         BACKUP1  BACKUP2    UPDTS
WORK Q: zzzzzzzz 11111111 22222222
```

```
WAIT Q: 33333333 44444444 55555555
DONE:   66666666 77777777 88888888
End of Display Detail
CBR9201I Object Processing completed for storage group xxxxx01.

CBR9500I Shelf Manager has started optical processing for storage group xxxxx01.

CBR9501I Shelf Manager completed optical processing for storage group xxxxx01.
0 cartridges selected. Detailed messages for each volume expiration will be
written to hardcopy.

CBR9500I Shelf Manager has started tape processing for storage group xxxxx01.

CBR9501I Shelf Manager completed tape processing for storage group xxxxx01.
0 cartridges selected. Detailed messages for each volume expiration will be
written to hardcopy.

CBR9048I Storage Group xxxxx01 has successfully completed processing.

CBR9200I Object Processing starting for storage group xxxxx12.

CBR9370I OSMC Detail for taskname:

            READ     READ     READ     READ
           DISK1    DISK2      OPT     TAPE
WORK Q: aaaaaaaa bbbbbbbb cccccccc dddddddd
WAIT Q:          eeeeeeee ffffffff gggggggg
DONE:   hhhhhhhh iiiiiiii jjjjjjjj kkkkkkkk
           WRITE    WRITE    WRITE    WRITE    WRITE
           DISK1    DISK2      OPT    TAPE1    TAPE2
WORK Q: llllllll mmmmmmmm nnnnnnnn oooooooo pppppppp
WAIT Q:          qqqqqqqq rrrrrrrr ssssssss tttttttt
DONE:   uuuuuuuu vvvvvvvv wwwwwwww xxxxxxxx yyyyyyyy
           WRITE    WRITE      DIR
         BACKUP1  BACKUP2    UPDTS
WORK Q: zzzzzzzz 11111111 22222222
WAIT Q: 33333333 44444444 55555555
DONE:   66666666 77777777 88888888
End of Display Detail
```

```
CBR9201I Object Processing completed for storage group xxxxx12.

CBR9500I Shelf Manager has started optical processing for storage group xxxxx12.

CBR9501I Shelf Manager completed optical processing for storage group xxxxx12.
0 cartridges selected. Detailed messages for each volume expiration will be
written to hardcopy.

CBR9048I Storage Group xxxxx12 has successfully completed processing.

CBR9009I OSMC completed its Storage Management Cycle.  12 tasks started.
         12 tasks completed.
```

If errors occur during the storage management process, additional messages might be issued, such as message CBR9049I to indicate unsuccessful completion.

Starting the storage management cycle for a selected set of storage groups

To start the OSMC storage management cycle for a selected set of Object or Object Backup storage groups:

1. Enter the following command:

   ```
   F OAM,START,STORGRP,storage-group-name
   ```

   where *storage-group-name* can be any of:

   - The name of a particular storage group (for example, `IMAFIRST`).
   - A wild card filter for a set of storage groups whose names all begin with the same characters (for example, `IMA*`), or
   - A wild card filter consisting of a single asterisk (*), which specifies all Object or Object Backup storage groups.

2. The system issues the following set of messages for each Object or Object Backup storage group whose name matches the specified name or wild card filter:

```
CBR1000I OAM START command execution scheduled.
CBR9500I Shelf Manager has started optical processing for storage group
storage-group-name.

CBR9501I Shelf Manager completed optical processing for storage group
storage-group-name. n cartridges selected.
Detailed messages for each volume expiration will be written to hardcopy.

CBR9500I Shelf Manager has started tape processing for storage group
storage-group-name.

CBR9501I Shelf Manager completed tape processing for storage group
storage-group-name. n cartridges selected.
Detailed messages for each volume expiration will be written to hardcopy.

CBR9048I Storage Group storage-group-name has successfully completed processing.
```

3. If the shelf manager selects volumes for expiration, CBR9501I has a value for *n* cartridges. OAM issues more detailed messages for each volume directly to the hardcopy log.

If errors occur during the storage management process for a storage group, additional messages might be issued, such as message CBR9049I to indicate unsuccessful completion.

The following is a sample of the F OAM,START,STORGRP command for a single storage group named IMAFIRST:

```
CBR1000I OAM START command execution scheduled.
CBR9500I Shelf Manager has started optical processing for storage group IMAFIRST.
CBR9501I Shelf Manager completed optical processing for storage group IMAFIRST.
1 cartridges selected. Detailed messages for each volume expiration will be
written to hardcopy.

CBR9500I Shelf Manager has started tape processing for storage group IMAFIRST.
CBR9501I Shelf Manager completed tape processing for storage group IMAFIRST.
0 cartridges selected. Detailed messages for each volume expiration will be
written to hardcopy.

CBR9048I Storage Group IMAFIRST has successfully completed processing.
CBR2154I Volumes RG589A and RG589B will be reinitialized on their next mount and will
remain assigned to storage group IMAFIRST.
```

Starting the library space management cycle

Starting the library space management cycle ejects the least-recently-written or the least-recently-mounted optical disk from the optical library. This command requires that the OAM system, on which the command was entered, control the requested library. If the member name associated with the optical library specified on this command is not the member name for this instance of OAM, the command fails and message CBR1068I is issued.

If you specified the Least-Recently-Mounted (LRM) parameter on the EJECT keyword of the OAM procedure statement, the library space management cycle ejects the least-recently-mounted optical disk from the optical library.

You can start the OSMC library space management cycle:

- Manually, by entering the START command. Use the START command if the library is full or if scratch volumes are needed in the library.
- Automatically, when OAM cannot locate a library-resident optical volume on which to write an object and the following conditions are met:
  - For the storage group:
    - There are no library optical volumes or
    - Those residing in the library are not usable at this time.
  - There are no scratch volumes in the optical library.
  - The library is full.

To start the OSMC library management cycle:

1. Enter the following command:

   ```
   F OAM,START,LIBMGT,library-name
   ```

   ____

2. The system issues the following messages:

   ```
   CBR1000I OAM START command execution scheduled.
   CBR9400I Library Space Manager starting for library library-name.
   ```

   ____

3. If a shelf location has not been previously specified for the optical disk being ejected, the system issues the following message and waits for a reply:

   ```
   CBR2600A Specify shelf location for volumes volser1 and volser2.
   ```

   ____

4. Provide 1 to 32 characters of shelf information.

   If the optical volume being ejected does not already have a valid pseudo library associated with it, or the library from which the volume was ejected does not have a default pseudo library in its definition, the system issues message CBR2602A or message CBR2603A, or both:

   ```
   CBR2602A Eject pending for volser in r-library-name. Default pseudo
            library is p-library-name. Reply 'U' to use, or 'R' to respecify.
   ```

   ```
   CBR2603A Specify pseudo library name for volume volser.
   ```

   If the volume being ejected has an invalid pseudo library associated with its volume record and the library from which the volume is being ejected has a default pseudo library in its SCDS definition, message CBR2602A is issued. This message asks if the default is to be used or if another pseudo library name is requested.

   If the library from which the volume is being ejected does not have a default pseudo library in its SCDS definition, or the operator replied with "R" to message CBR2602A, the system issues message CBR2603A. This message requests a pseudo library name to assign the volume to when it becomes shelf resident.

   If a pseudo library name specified in response to message CBR2603A is not valid in the current active control data set (ACDS), or the volume record has an invalid pseudo library name associated with it, the system issues messages CBR2604I and CBR2603A:

   ```
   CBR2604I Volume volser cannot be assigned to pseudo library p-library-name, it
            is not a valid pseudo library definition in the active SMS configuration.
   CBR2603A Specify pseudo library name for volume volser.
   ```

   The system puts the optical disk cartridge in the input/output station and issues the following messages:

   ```
   CBR3122I Volumes volser1 and volser2 were ejected from library library-name
            shelf location is shelfloc.

   CBR3001A Remove cartridge from I/O station on library library-name.  Place in
            shelf location shelfloc.
   ```

   ____

5. Remove the optical disk cartridge from the input/output station of the specified optical library and return it to the shelf location indicated.

   ```
   CBR9401I Library Space Manager completed for library library-name. n optical
            disks ejected.
   ```

If any errors occur during the library space management cycle, additional messages might be issued identifying the error, and message CBR9401I is not issued.

For more information on these messages, see *MVS System Messages*.

Starting the DASD space management cycle for an individual storage group

Use DASD space management to do the following:

- Select objects that require processing.
- Determine if they have expired by examining the objects' explicit expiration date or examining the objects' management class information.
- Physically delete data for expired objects on DASD and on rewritable optical disks.
- Remove the object directory entry (indicating volume serial number and sector location and other object information) for the expired object.

To start DASD space management:

1. Enter the following command:

```
F OAM,START,DASDSM,storage-group-name
```

---

2. The system issues the following messages:

```
CBR9300I DASD Space Management starting for storage group storage-group-name.
CBR9301I DASD Space Management completed for storage group storage-group-name.
```

---

3. The following message appears if OAM is having some major problems and cannot perform DASD space management on a storage group. This message would be preceded by other CBR*xxxx*I messages:

```
CBR9043I DASD Space Manager not started for storage-group-name.
```

If any errors occur during the DASD space management cycle, additional messages might be issued identifying the error, and message CBR9043I is not issued.

Note: If OAM detects that an object storage table or an object directory is unable to extend its storage during normal processing of objects using the OSREQ application interface, DASD space management is automatically started for that storage group. If the table that runs out of extents is the object directory, all object stores to that storage group fail until more space is made available to the object directory for the storage group. DASD space management is started automatically in this instance so that subsequent stores to the storage group might have improved chances of storing successfully.

For more information on these messages, see *MVS System Messages*.

Starting the OAM Volume Recovery utility

You can use the Volume Recovery utility to perform the following tasks:

Subtask	Associated procedure (see . . .)
Recover tape volumes or optical disks.	"Starting a recovery for an optical or tape volume" on page 271
Delete a recovered tape or optical volume from the OAM inventory.	"Deleting a recovered tape or optical volume" on page 275

Use the Volume Recovery utility to recover objects that reside on an unusable optical or tape volume. It does not recover objects to the disk level. The utility only retrieves copies of the objects that are stored on disk to recover them to optical or tape volumes when recovering an Object Backup volume.

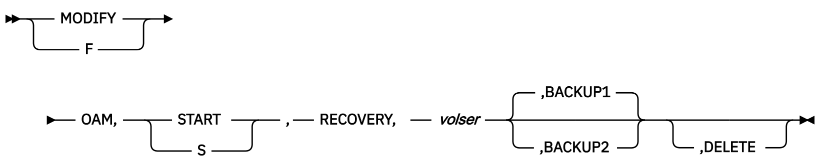
You can use the Volume Recovery utility to recover objects from tape sublevel 1 source volumes to tape sublevel 1 target volumes, or recover objects from tape sublevel 2 source volumes to tape sublevel 2 target volumes.

Typically, some of the objects are recovered to the Object or Object Backup storage group volume that is currently being written, and the rest of the objects are recovered to the next assigned Object or Object Backup storage group volume. If there are two backup copies of the object, you can indicate which copy of the object (first or second backup copy) is used in the recovery by indicating BACKUP1 or BACKUP2 on the RECOVERY command.

For more information on the Volume Recovery utility, see "Recovering an entire optical cartridge or tape volume" on page 218.

Starting a recovery for an optical or tape volume

The following fragment shows the syntax for the RECOVERY command. For the complete syntax for all F OAM,START commands, see "Starting OSMC functions" on page 263.



To recover an optical or tape volume:

1. Enter the following RECOVERY command:

   ```
   F OAM,START,RECOVERY,volser,BACKUP1
   ```

   where *volser* is the volume serial number of one of the volumes that is being recovered.

   If more than one backup copy of the objects exists, you can select whether to use the first or second backup copy for the recovery by specifying BACKUP1 or BACKUP2 on the F OAM,START,RECOVERY command when recovering a PRIMARY volume. If you omit BACKUP1 and BACKUP2, OAM recovers the first backup copy of the object by default.

   To recover a primary volume, all of the backup volumes containing backup copies of the object on the primary volume are needed whether they are optical or tape. Also, to recover a backup volume, each Object storage group must be searched for objects that have a backup copy on the backup volume to be recovered. For each of these objects, the primary copy is used to recover the backup volume. The primary copy of these objects could be on disk (DB2 or file system), optical, or tape. The Volume Recovery utility must identify the optical and tape volumes that are needed for the recovery.

   Requirement: The RECOVERY command has the following requirements:

   - The optical or tape volume that is specified on the command is not controlled by another OAM within the OAMplex.
   - Unmounted shelf-resident volumes must be available to the instance of OAM on which the command was entered.

   If the member name associated with the optical or tape volume serial number specified on this command is not the member name for this instance of OAM, the command fails. OAM issues message CBR1068I.

2. The system issues the following messages:

```
CBR1000I OAM START command execution scheduled.

CBR9800I  OAM Volume Recovery starting for volumes volser1 and volser2.

CBR9827I  OAM Volume Recovery. The following TAPE volumes are needed
          for recovery: volser1 volser2 volser3 volser4 volser5
          volser6 volser7 volser8 volser9

CBR9824I  OAM Volume Recovery. The following OPTICAL volumes are needed
          for recovery: volser1 volser2 volser3 volser4 volser5
          volser6 volser7 volser8 volser9.
```

   Message CBR9800I indicates that volume recovery has started for volumes *volser1* and *volser2*. In situations where a tape volume is being recovered, *volser2* is labeled as `N/A`.

   Message CBR9824I gives you a list of optical volumes to retrieve that are identified by *volser-n*. This message allows you to get the optical volumes that are needed for recovery processing.

   Message CBR9827I gives you a list of tape volumes to retrieve that are identified by *volser-n*. This message allows you to get the tape volumes that are needed for recovery processing.

   ____

3. For the volumes listed in messages CBR9827I and CBR9824I, the system issues the following message:

```
CBR9810D  Reply 'QUIT' to terminate or 'GO' to proceed with recovery.
```

   ____

4. If all the volumes are not available, reply QUIT to terminate recovery, and start again when the volumes have been retrieved.

   If you reply QUIT to CBR9810D, the system issues the following messages:

```
CBR9862I Volume Recovery status for volumes volser1 and volser2 is not available.
CBR9819I OAM Volume Recovery is ending for volumes volser1 and volser2
```

   ____

5. If the volumes are available, reply GO to CBR9810D.

   ____

6. The system issues the following message for each optical volume listed in message CBR9824I:

```
CBR4400A Mount volume volser on drive drive-name.
 Shelf location is shelfloc.
```

   ____

7. The system issues the following message for each tape volume that is listed in the message CBR9827I:

```
IEC501A M drive-Addr,volser,label,,,,data_set_name.
```

   Message CBR9824I might identify volumes that are either library-resident or shelf-resident optical volumes. The system automatically mounts the library-resident optical volumes; therefore, a mount message is not issued for them. The mount message CBR4400A requests only shelf-resident optical volumes for recovery.

   ____

8. Mount the optical volume that is identified by *volser* in message CBR4400A and any tape volumes identified by *volser* in message IEC501A.

When recovery is complete, the system issues the following message:

```
CBR9819I OAM Volume Recovery is ending for volumes volser1 and volser2
```

The following is a sample of the F OAM,START,RECOVERY,*volser* command:

```
F OAM,S,RECOVERY,WG360A,BACKUP2

CBR1000I OAM S command execution scheduled.
CBR9800I OAM Volume Recovery starting for volumes WG360A and WG360B.
CBR9827I OAM Volume Recovery.
The following TAPE volumes are needed for recovery:
 TNN005
CBR9824I OAM Volume Recovery.
The following OPTICAL volumes are needed for recovery:
 WG688A
CBR9810D Reply 'QUIT' to terminate or 'GO' to proceed with recovery.

If you reply GO, the system issues the following messages:

CBR9852I Volume Recovery Utility processing objects in storage group GROUP22
         for volume WG360A.
IEC501A M 1D04,TNN005,SL,COMP,OAM,OAM,OAM.BACKUP2.DATA
CBR4401I Volume WG830B mounted on drive P9D3.
CBR9863I Volume Recovery status for volumes WG360A and WG360B.
  Total: 5, Attempted: 5, Successful: 5, Unsuccessful: 0, Remaining: 0.
CBR9819I OAM Volume Recovery is ending for volumes WG360A and WG360B.

If you reply QUIT, the system issues the following messages:

CBR9862I Volume Recovery status for volumes WG360A and WG360B is not
available.
CBR9819I OAM Volume Recovery is ending for volumes WG360A and WG360B.
```

If any errors occur during the volume recovery process, additional messages might be issued identifying the errors.

Message CBR9863I provides statistics for volume recovery status. The statistics include: Total, Attempted, Successful, Unsuccessful, and Remaining. These statuses have the following meanings:

Total
The total number of objects found on *volser1* and *volser2*

Attempted
The total number of objects for which processing has begun in this utility for *volser1* and *volser2*

Successful
The total number of objects successfully recovered for *volser1* and *volser2* and written to other volumes

Unsuccessful
The total number of attempted objects for which processing has begun in this utility, but which were not successfully recovered for *volser1* and *volser2*. This number represents the number of objects that started processing but did not complete successfully.

Remaining
The total number of objects that were not recovered.

The text of the message follows:

```
CBR9863I Volume Recovery status for volumes volser1 and volser2.
          Total: total, Attempted: attempted, Successful: successful
          Unsuccessful: unsuccessful, Remaining: remaining
```

If you attempt to recover a primary volume containing one or more objects that do not have backup copies at the requested level, OAM issues message CBR9864I. Processing will continue for objects that do have a backup, but the DELETE option, if specified, is ignored since not all objects could be recovered.

```
CBR9864I A total of object-count objects on volumes volser and volser2 do not have
         a {1st|2nd} backup copy.
```

For a description of the messages, see *z/OS MVS System Messages, Vol 4 (CBD-DMO)*.

Statistics provided by the Volume Recovery utility

The Volume Recovery utility allows you to recover backup tape volumes.

Normal completion of recovery of a backup tape volume

Figure 24 on page 274 shows recovery of a backup tape volume from the primary objects that reside on both tape and optical volumes. Recovery completion is normal. All 1043 backup objects are successfully recovered from the primary objects.

```
F OAM,S,RECOVERY,CMW099

   CBR1000I OAM S command execution scheduled.
   CBR9800I OAM Volume Recovery starting for volumes CMW099 and N/A.
   CBR9827I OAM Volume Recovery.
            The following TAPE volumes are needed for recovery:
            CMW095 CMW097
   CBR9824I OAM Volume Recovery.
            The following OPTICAL volumes are needed for recovery:
            WG229A WG229B WG986A WG986B
            0009 CBR9810D Reply 'QUIT' to terminate or 'GO' to proceed with recovery.
            R 9,GO
   IEE600I  REPLY TO 0009 IS;GO
   CBR9852I Volume Recovery Utility processing objects in storage group
            GROUP10 for volume CMW099.
   CBR4401I Volume WG229A mounted on drive P15D2.
   CBR4401I Volume WG229B mounted on drive P15D2.
   IEC501A  M 1D11,CMW222,SL,NOCOMP,OAM,OAM,OAM.BACKUP.DATA.BACK1
   IEC205I  CBRRT001,OAM,OAM,FILESEQ=1, COMPLETE VOLUME LIST,
            DSN=OAM.BACKUP.DATA.BACK1,VOLS=CMW222,TOTALBLOCKS=12798
   IEF234E  K 1D11,CMW222,PVT,OAM,OAM
   CBR4401I Volume WG986A mounted on drive P15D3.
   CBR4401I Volume WG986B mounted on drive P15D3.
   IEC501A  M 1D12,CMW222,SL,NOCOMP,OAM,OAM,OAM.BACKUP.DATA.BACK1
   CBR9852I Volume Recovery Utility processing objects in storage group GROUP13 for volume
CMW099.
   IEC501A  M 1D11,CMW095,SL,NOCOMP,OAM,OAM,OAM.PRIMARY.DATA.GROUP13
   IEF234E  K 1D11,CMW095,PVT,OAM,OAM
   IEC205I  CBRRT002,OAM,OAM,FILESEQ=1, COMPLETE VOLUME LIST,
            DSN=OAM.BACKUP.DATA.BACK1,VOLS=CMW222,TOTALBLOCKS=13578
   IEF234E  K 1D12,CMW222,PVT,OAM,OAM
   IEC501A  M 1D11,CMW097,SL,NOCOMP,OAM,OAM,OAM.PRIMARY.DATA.GROUP13
   IEC501A  M 1D13,CMW222,SL,NOCOMP,OAM,OAM,OAM.BACKUP.DATA.BACK1
   CBR9863I Volume Recovery status for volumes CMW099 and N/A.
            Total: 1043, Attempted: 1043, Successful: 1043, Unsuccessful: 0, Remaining: 0.
   CBR9819I OAM Volume Recovery is ending for volumes CMW099 and N/A.
```

Figure 24. Example of a Normal Completion of Recovery of a Backup Tape Volume

Normal completion of recovery of an optical volume without all backup copies

Figure 25 on page 275 shows recovery of a primary optical volume using the first backup copies of the objects. There are a total of 1000 primary objects on the optical volume and its opposite side, but only 995 of the objects have a backup copy. Message CBR9364I shows the number of objects without a backup copy. Recovery processing is not attempted for these objects. Therefore, only 995 of the objects are successfully recovered. Message CBR9863I shows there were 5 objects remaining on the volumes following recovery completion and zero objects were unsuccessfully recovered. The "Unsuccessful" field represents only the number of objects that the Volume Recovery utility attempted and failed to recover.

```
F OAM,S,RECOVERY,WG229A,BACKUP1

   CBR1000I OAM S command execution scheduled.
   CBR9800I OAM Volume Recovery starting for volumes WG229A and WG229B.
   CBR9827I OAM Volume Recovery.
            The following TAPE volumes are needed for recovery:
            CMW222
            0010 CBR9810D Reply 'QUIT' to terminate or 'GO' to proceed with recovery.
            R 10,GO
   IEE600I  REPLY TO 0010 IS;GO
   CBR9852I Volume Recovery Utility processing objects in storage group
            GROUP10 for volume WG229A.
   CBR9864I A total of 5 objects on volumes WG229A and WG229B do not have a
            1st backup copy.
   IEC501A  M 1D15,CMW222,SL,NOCOMP,OAM,OAM,OAM.BACKUP.DATA.BACK1
   CBR2200I Scratch volumes RG877A and RG877B added to storage group
            GROUP10.
   IEF234E  K 1D15,CMW222,PVT,OAM,OAM
   CBR4401I Volume RG877A mounted on drive P15D1.
   IEC501A  M 1D12,CMW222,SL,NOCOMP,OAM,OAM,OAM.BACKUP.DATA.BACK1
   IEF234E  K 1D12,CMW222,PVT,OAM,OAM
   CBR9863I Volume Recovery status for volumes WG229A and WG229B.
            Total: 1000, Attempted: 995, Successful: 995, Unsuccessful: 0, Remaining: 5.
   CBR9819I OAM Volume Recovery is ending for volumes WG229A and WG229B.
```

Figure 25. Example of a Normal Completion of Recovery of an Optical Volume without All Backup Copies

Limited completion of recovery of a backup volume due to error condition

Figure 26 on page 275 shows the recovery of a backup volume from primary objects that reside on a tape volume. A total of 4636 objects were identified as residing on the backup volume. The Volume Recovery utility attempted to recover 1496 of these objects, but processing was stopped due to a repeating read error.

The Volume Recovery utility was unable to determine how many of the objects were successfully recovered before the failure occurred and how many remain on the volume being recovered, so the system issued message CBR9862I indicating limited recovery for the volume. The "Successful", "Unsuccessful", and "Remaining" values in message CBR9863I show statistics of **** because those values could not be accurately determined.

```
F OAM,S,RECOVERY,CMW212

   CBR1000I OAM S command execution scheduled.
   CBR9800I OAM Volume Recovery starting for volumes CMW212 and N/A.
   CBR9827I OAM Volume Recovery.
            The following TAPE volumes are needed for recovery:
            CMW222
            0013 CBR9810D Reply 'QUIT' to terminate or 'GO' to proceed with recovery.
            R 13,GO
   IEE600I  REPLY TO 0013 IS;GO
   CBR9852I Volume Recovery Utility processing objects in storage group GROUP06 for volume CMW212.
   CBR9103I A READ error occurred during Storage Management Processingfor BACKUP for Collection
            GROUP06, Object GROUP06.C1C0.HQ2020, in
            Storage Group BACK1, on volume CMW222.
            The return code is 0000000Cand the reason code is 00000910.
   CBR9915I Module CBRHROPT is stopping OSMC control task CMW212 because of repeating error condition CBR9103.
   CBR9862I Volume Recovery status for volumes CMW212 and N/A is limited.
   CBR9863I Volume Recovery status for volumes CMW212 and N/A.
            Total:  4636, Attempted: 1496, Successful: ****, Unsuccessful: ****,
            Remaining: ****.
   CBR9819I OAM Volume Recovery is ending for volumes CMW212 and N/A.
```

Figure 26. Example of a Limited Completion of Recovery of a Backup Volume Due to Error Condition

Deleting a recovered tape or optical volume

The DELETE option on the RECOVERY command removes the specified tape volume or optical platter from the OAM inventory after it has been recovered successfully.

1. Enter the following RECOVERY command:

   ```
   F OAM,START,RECOVERY,volser,DELETE
   ```

where *volser* is the volume serial number of the tape or optical volume that is to be recovered.

2. The system issues the following messages:

```
CBR1000I OAM START command execution scheduled.
CBR9800I OAM Volume Recovery Delete starting for volumes volser1 and volser2.
CBR9824I OAM Volume Recovery.
The following OPTICAL volumes are needed for recovery:
volser1 volser2 volser3 volser4 volser5
volser6 volser7 volser8 volser9.
CBR9810D Reply 'QUIT' to terminate or 'GO' to
proceed with recovery.
```

3. If the volumes are available and readable, reply GO to CBR9810D. Otherwise, reply QUIT to CBR9810D.

4. If you have specified the DELETE option on the Volume Recovery command for an optical volume, the hardcopy log displays the following additional message:

```
CBR2153I All objects on volumes volser1 and volser2 have expired,
shelf location shelfloc.
```

If you have specified the DELETE option on the Volume Recovery command for an object tape volume, the hardcopy log displays the following additional message:

```
CBR2165I Tape volume volser has had all objects expired or deleted and
can be returned to the MVS scratch pool.
```

You can use message CBR2165I to notify the tape management system that the volume can be returned to MVS scratch status.

After you delete a recovered volume, it is no longer available to any of the OAM members in the OAMplex. OAM removes the information about the tape or optical volume from the OAM inventory. The rows that are associated with these volumes are deleted from the TAPEVOL or VOLUME table in DB2. Library-resident optical volumes are ejected from the library.

The following is a sample of the F OAM,START,RECOVERY,*volser*,DELETE command:

```
F OAM,S,RECOVERY,CMW226,DELETE

   CBR1000I OAM S command execution scheduled.
   CBR9800I OAM Volume Recovery Delete starting for volumes CMW226 and N/ACBR9827I OAM Volume Recovery.
            The following TAPE volumes are needed for recovery:
            CMW222
            0010 CBR9810D Reply 'QUIT' to terminate or 'GO' to proceed with recovery.
            R 10,GO
   IEE600I  REPLY TO 0010 IS;GO
   CBR9852I Volume Recovery Utility processing objects in storage group
            GROUP04 for volume CMW226.
   IEC501A  M 1D14,CMW222,SL,NOCOMP,OAM,OAM,OAM.BACKUP.DATA.BACK1
   CBR9863I Volume Recovery status for volumes CMW226 and N/A.
            Total: 5, Attempted: 5, Successful: 5, Unsuccessful: 0, Remaining: 0.
   CBR9819I OAM Volume Recovery is ending for volumes CMW226 and N/A.
   CBR2165I Tape volume CMW226 has had all objects expired or deleted and can be returned to the MVS scratch
pool.
```

For more information about these system messages, see *z/OS MVS System Messages, Vol 4 (CBD-DMO)*.

Starting object recovery for single objects

OAM contains a Single Object Recovery utility for recovering a single object from an optical or tape volume. The system uses a backup copy of the object (either on optical or tape), if any exist. The backup copy that is used for the recovery can reside either on tape or optical media. If more than one backup copy of the object exists, you can select whether to use the first or second backup copy of the object for the recovery by specifying BACKUP1 or BACKUP2 on the F OAM,START,OBJRECV command. A new primary copy of the object is written to the same Object storage group and same media type (optical, tape, or disk) as the original object. The following are examples of how single object recovery works:

- If the primary object resides on optical disk, the backup copy that is selected for the recovery (on either optical disk or tape) creates a new optical primary copy.
- If the primary object resides on tape, the backup copy that is selected for the recovery (on either optical disk or tape) creates a new tape primary copy.
- If the primary object resides on the DB2 sublevel, the backup copy (on either tape or optical disk) creates a new primary copy on the DB2 sublevel.
- If the primary object resides on the file system sublevel, the backup copy (on either tape or optical disk) creates a new primary copy on the file system sublevel.

Note: Starting too many single object recoveries at once may result in ABEND878.

To recover a single object:

1. Enter the following command:

   ```
   F OAM,START,OBJRECV,collection-name,object-name,BACKUP1|BACKUP2
   ```

   ____

2. The system issues the following message:

   ```
   CBR1000I OAM START command execution scheduled.
   ```

   ____

3. If the backup volume is an optical volume and does not reside in an optical library, the system issues the following message:

   ```
   CBR4400A Mount volume volser on drive drive-name. Shelf location is shelfloc.
   ```

   ____

4. If the backup volume is a tape volume, the system issues the following message:

   ```
   IEC501A M drive-Addr,volser,label,,,,data_set_name.
   ```

   ____

5. Mount the optical volume or tape volume that is identified by *volser*.

   ____

6. When recovery is complete, the system issues the following message:

   ```
   CBR9830I Single Object Recovery complete for collection collection-name,
            object object-name.
   ```

   ____

The following is a sample of the F OAM,START,OBJRECV command:

```
F OAM,START,OBJRECV,GROUP22,GROUP22.IDVT.H11,BACKUP2

CBR1000I OAM START command execution scheduled.
CBR4401I Volume WG688A mounted on drive P9D1.
CBR4401I Volume WG360A mounted on drive P9D2.
CBR9830I Single Object Recovery complete for
collection GROUP22, object  GROUP22.IDVT.H11.
```

If any errors occur during the single object recovery process, additional messages might be issued identifying the errors.

For descriptions of these messages, see *MVS System Messages*.

Starting the tape recycle utility

The following example shows a `F OAM,START,RECYCLE` command with display and *percentvalid* specified, indicating that the result shows volumes that meet criteria of scope, TSL, and *percentvalid*, but these volumes are not processed.

```
F OAM,S,RECYCLE,(ALLGRP),PV=20,TSL=2,DISPLAY
CBR9880I OAM START RECYCLE command starting.
CBR9875I Recycle Candidates:
The following volumes are candidates for OAM RECYCLE command
processing using pv=20, lim=N/A, scope=(ALLGRP), maxrecycletasks=15, TSL=2.
  VOLSER  %VAL  SGNAME   STAT   VOLSER   %VAL  SGNAME   STAT
  CMW212     0  GROUP07         CMW215      0  GROUP04
  CMW250     5  GROUP17         CMW238     10  GROUP12
  CMW222    15  GROUP00         CMW237     19  GROUP22
  CMW233    20  GROUP02         CMW242     20  GROUP10
  CMW281    20  GROUP81         CMW711     20
GROUP36
OAM Recycle: End of OAM Recycle candidate volumes. Total volumes=10.
CBR9881I OAM START RECYCLE command ending successfully. Reason is
display specified.
```

The following example shows a `F OAM,START,RECYCLE` using LIM = along with *scope* and TSL, indicating that these volumes will be processed:

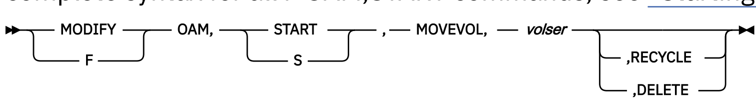
```
F OAM,S,RECYCLE,(ALLGRP),LIM=16, TSL=2
CBR9880I OAM START RECYCLE command starting.
CBR9875I Recycle Candidates:
The following volumes are candidates for OAM RECYCLE command
processing using pv=100, lim=16, scope=(ALLGRP), maxrecycletasks=15, TSL=2.
  VOLSER  %VAL  SGNAME   STAT   VOLSER  %VAL  SGNAME   STAT
  CMW210     0  GROUP07         CMW217     0  GROUP04
  CMW223     0  GROUP00         CMW235     0  GROUP12
  CMW239     0  GROUP09         CMW241     0  GROUP10
  CMW283     0  GROUP01         CMW710     0  GROUP00
  CMW712     0  GROUP02         CMW714     0  GROUP08
  CMW716     0  GROUP05         CMW718     0  GROUP04
  CMW767    20  GROUP06         CMW769    50  GROUP04
  CMW782    67  GROUP03         CMW788    90  GROUP03
OAM Recycle: End of OAM Recycle candidate volumes. Total volumes=16.
.
...normal MOVEVOL processing messages for each volume processed
.
CBR9881I OAM START RECYCLE command ending successfully. Reason is limit
 reached.
```

Note: The algorithm used by OAM to calculate the %VAL rounds down to the nearest whole percent value. A %VAL of 0 does not necessarily indicate that there is no valid data on a tape.

For descriptions of these messages, see *z/OS MVS System Messages, Vol 4 (CBD-DMO)*.

Starting the Move Volume utility

The following fragment shows the syntax for the MOVEVOL command for the Move Volume utility. For the complete syntax for all F OAM,START commands, see "Starting OSMC functions" on page 263.



Use the OAM Move Volume utility to move (migrate) objects from a primary or backup source volume (a tape volume or one side of an optical disk) to one or more target volumes. You can optionally recycle or delete the source volume after moving the objects to the target volumes.

The MOVEVOL command specified with the RECYCLE option will be rejected for both physical and logical WORM tape volumes. The MOVEVOL command without RECYCLE or with the DELETE option is allowed for WORM tape volumes. The RECYCLE option is rejected for physical WORM tape volumes because they cannot be rewritten from load point and cannot be recycled for reuse. To be consistent with the behavior of physical WORM, the RECYCLE option is also rejected for logical WORM tape volumes. However, with logical WORM, once a logical WORM volume expires and the tape management system returns the logical

WORM volume to scratch (to reuse the logical volume's VOLSER), the logical volume can be used as either WORM or R/W (rewritable).

Tip: The DELETE or RECYCLE options of the MOVEVOL command moves data on both sides of an optical volumes.

The F OAM,START,MOVEVOL command requires that the optical or tape volume specified on the command is either controlled by the instance of OAM on which the command was entered (not another OAM within the OAMplex), or if the volume is an unmounted shelf-resident volume, it must be available to the instance of OAM on which the command was entered. If the member name associated with the optical or tape volume serial number specified on this command is not blank or is not the member name for this instance of OAM, the command fails. OAM issues message CBR1068I.

You can use the MOVEVOL to move:

- Objects from tape sublevel 1 source volume to tape sublevel 1 target volume.
- Objects from tape sublevel 2 source volume to tape sublevel 2 target volume.

MOVEVOL cannot be used to move objects from tape sublevel 1 volumes to tape sublevel 2 volumes, or from tape sublevel 2 volumes to tape sublevel 1 volumes. To change an object's residency, for example, from a tape sublevel 1 volume to a tape sublevel 2 volume, change the object's storage class and start an OSMC cycle to affect the transition to another media type.

Move volume scenarios

In this example, there are four Object storage groups, two primary volumes (A and B), and two backup volumes (C and D) as shown in Figure 27 on page 279.

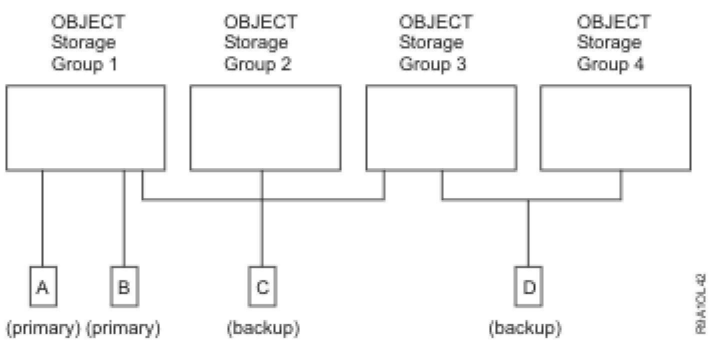


Figure 27. Object Storage Group and Volume Relationship

You *can* concurrently start the Move Volume utility in the following situations:

- A and D or B and D—A primary source volume and a backup source volume where the primary volumes containing the objects are in different Object storage groups. The Move Volume utility for primary source volume A only accesses objects in Object storage group 1, and the Move Volume utility for backup source volume D only accesses objects in Object storage groups 3 and 4.

The Move Volume utility *should not* be started concurrently for the following situations:

- A and B—Two primary source volumes in the same Object storage group. This scenario concurrently accesses objects in Object storage group 1 during the two Move Volume utilities processing primary source volumes A and B.
- A and C or B and C—A primary source volume and a backup source volume where the primary volumes containing the objects are in the same Object storage group. This scenario concurrently accesses objects in Object storage group 1 during the two Move Volume utilities processing primary source volume A and backup source volume C.
- C and D—Two backup source volumes where the primary volumes containing the objects are in the same Object storage group. This scenario concurrently accesses objects in Object storage group 3 during the two Move Volume utilities processing backup source volumes C and D.

Moving objects from a source volume

To move objects from a source volume:

1. Issue the following MOVEVOL command:

   ```
   F OAM,START,MOVEVOL,volser
   ```

   where *volser* is the volume serial of the source volume from which objects are to be moved.

   ____

2. The system issues the following messages:

   ```
   CBR1000I OAM START command execution scheduled.

   CBR9850I Move Volume Utility starting for volume volser.

   CBR9858I Move Volume Utility status for volume volser. Total: total,
            Attempted: attempted, Successful: successful,
            Unsuccessful: unsuccessful.

   CBR9859I Move Volume utility ending for volumes volser1 and volser2.
   ```

   ____

3. If any errors occur or contention exists due to concurrent processing, you must take appropriate actions to correct the errors or minimize contention. You can then re-invoke the Move Volume utility to continue moving objects from the source volume. See “Statistics provided by the Move Volume utility” on page 284 for examples of various conditions that can exist and the resulting statistics that the utility provides.

   ____

When the MOVEVOL command completes, all of the objects are moved to the target volumes and deleted from the source volume.

Attention: If you specified the Move Volume utility with the RECYCLE or DELETE option, the volume is automatically dispositioned (recycled or deleted). However, if you did not specify the RECYCLE or DELETE option, you must update the volume expiration date to ensure that OAM selects the volume for expiration processing. Use the F OAM,UPDATE,VOLUME command to update the EXPDATE field in the DB2 Volume or Tape Volume tables. For more information about this command, see “Updating fields in the DB2 Volume Table and the Tape Volume Table” on page 374.

Recommendations for using the Move Volume utility:

1. For a given optical disk (consisting of two volumes), only one Move Volume utility (which moves data from one of the two volumes on the disk) or one Volume Recovery utility (which recovers data on both volumes on the disk) can be started for a disk, but not both.
2. For a give tape volume, only one Move Volume utility or one Volume Recovery utility can be started for the volume, but not both.
3. You cannot start the Move Volume utility concurrently for both volumes on an optical disk.
4. You cannot start the Move Volume utility for a volume that is on either side of an optical disk that is being recovered by the Volume Recovery utility.
5. You can start one or more Move Volume utilities only after determining that sufficient resources are available to the utility. Consider the resources required by other OSMC functions, such as the OSMC storage management cycle, which might run concurrently with the Move Volume utility.
6. To avoid contention, it is recommended that you do not start the Move Volume utility for multiple volumes that can cause concurrent references to objects in the same Object storage group. References to objects in the same Object storage group concurrently can occur when there are two primary source volumes in the same Object storage group, two backup source volumes where the primary volumes containing the objects are in the same Object storage group, or any combination of the above. Figure 27 on page 279 illustrates these relationships.

7. If the member name associated with the optical or tape volser specified on the MOVEVOL command is not the member name for this instance of OAM or not blank, the command fails and the message CBR1068I is issued.

Moving objects and recycling the source volume

To recycle both sides of an optical platter or rewritable tape volume after moving the objects off of it:

1. Issue the following MOVEVOL command:

   ```
   F OAM,START,MOVEVOL,volser,RECYCLE
   ```

   where *volser* is the volume serial of the source volume from which objects are to be moved.

   ____

2. The system issues the following messages:

   ```
   CBR1000I OAM START command execution scheduled.

   CBR9850I Move Volume Utility starting for volume volser.
   CBR9852I Move Volume Utility processing objects in
   storage group sgroup for volume volser.
   CBR4401I Volume volser mounted on drive drive.
   CBR2200I Scratch volumes volser1 and volser2 added to
   storage group sgroup.
   CBR9858I Move Volume Utility status for volume volser. Total: total,
            Attempted: attempted, Successful: successful,
            Unsuccessful: unsuccessful.

   CBR9859I Move Volume utility ending for volumes volser1 and volser2.
   ```

   ____

When the MOVEVOL command completes, the source tape or optical cartridge is recycled using the parameters that are specified in the SETOAM TAPERECYCLEMODE statement or the SETOPT OPTICALREINITMODE statement in the CBROAM*xx* member of PARMLIB. You can reuse all of the space on rewritable optical media or tape volume. Although you cannot reclaim the previously used space on WORM optical media, you can reuse the remaining free space on the platter.

The following is a sample of the F OAM,START,MOVEVOL,*volser*,RECYCLE command:

```
F OAM,START,MOVEVOL,R8040A,RECYCLE

CBR1000I OAM START command execution scheduled.
CBR9800I OAM Move Volume Recycle starting for volumes R8040A and R8040B.
CBR9850I Move Volume Utility starting for volume R8040A.
CBR9852I Move Volume Utility processing objects in
storage group GROUP58 for volume R8040A.
CBR4401I Volume R8040A mounted on drive P8XAD3.
CBR2200I Scratch volumes R8005A and R8005B added to
storage group GROUP58.
CBR4401I Volume R8005A mounted on drive P8XAD2.
CBR9858I Move Volume Utility status for volume R8040A.
Total: 130, Attempted: 130, Successful: 130, Unsuccessful: 0.
CBR9850I Move Volume Utility starting for volume R8040B.
CBR9858I Move Volume Utility status for volume R8040B.
Total: 0, Attempted: 0, Successful: 0, Unsuccessful: 0.
CBR9859I Move Volume Utility ending for volumes R8040A and R8040B.
```

The following sections describe the completion messages.

Related reading:

- For more information about using the Move Volume utility with the RECYCLE option, see "Reusing recycled tape and optical volumes" on page 228.
- For more information about these system messages, see *z/OS MVS System Messages, Vol 4 (CBD-DMO)*.

Recycle completion messages for rewritable tape volumes

If you specify the MOVEVOL command with the RECYCLE option for a tape volume and the SETOAM TAPERECYCLEMODE statement specifies GROUP, the system issues the following message to the hardcopy log:

```
CBR2166I Tape volume volser has had all objects expired or deleted and
will remain assigned to storage group storage_group.
```

If you specify the MOVEVOL command with the RECYCLE option for a tape volume and the SETOAM TAPERECYCLEMODE statement specifies OAMSCRATCH, the system issues the following message to the hardcopy log:

```
CBR2164I Tape volume volser has had all objects expired or deleted and
has been returned to OAM scratch status.
```

If you specify the MOVEVOL command with the RECYCLE option for a tape volume and the SETOAM TAPERECYCLEMODE statement specifies MVSSCRATCH, the system issues the following message to the hardcopy log:

```
CBR2165I Tape volume volser has had all objects expired or deleted and
can be returned to the MVS scratch pool.
```

Recycle completion messages for rewritable optical volumes

If you specify the MOVEVOL command with the RECYCLE option for a rewritable optical volume and the SETOPT OPTICALREINITMODE statement specifies GROUP, the system issues the following message to the hardcopy log:

```
CBR2154I Volumes volser1 and volser2 will be reinitialized on
their next mount and will remain assigned to storage group storage_group.
```

If you specify the MOVEVOL command with the RECYCLE option for a rewritable optical volume and the SETOPT OPTICALREINITMODE statement specifies OAMSCRATCH, the system issues the following message to the hardcopy log:

```
CBR2151I Volumes volser1 and volser2 will be reinitialized on
their next mount and have been returned to OAM scratch status.
```

Recycle completion messages for WORM optical volumes

If you specify the MOVEVOL command with the RECYCLE option for a WORM optical volume and the SETOPT OPTICALREINITMODE statement specifies GROUP, the system issues the following message to the hardcopy log:

```
CBR2170I Volumes volser1 and volser2 have completed reinitialization processing
and will remain assigned to storage group storage_group.
```

If you specify the MOVEVOL command with the RECYCLE option for a WORM optical volume and the SETOPT OPTICALREINITMODE statement specifies OAMSCRATCH, the system issues the following message to the hardcopy log:

```
CBR2169I Volumes volser1 and volser2 have completed reinitialization processing
and have been returned to OAM scratch status.
```

Moving objects and deleting the source volume

To delete both sides of an optical platter or tape volume from the OAM inventory after moving the objects off of it:

1. Issue the following MOVEVOL command:

   ```
   F OAM,START,MOVEVOL,volser,DELETE
   ```

where *volser* is the volume serial of the source volume from which objects are to be moved.

____

2. The system issues several messages, some of which are listed here:

```
CBR1000I OAM START command execution scheduled.

CBR9850I Move Volume Delete starting for volume volser.

CBR9858I Move Volume Utility status for volume volser. Total: total,
         Attempted: attempted, Successful: successful,
         Unsuccessful: unsuccessful.
 CBR9852I Move Volume Utility processing objects in
storage group sgroup for volume volser.
CBR9859I Move Volume utility ending for volumes volser1 and volser2.
```

____

3. If you have specified the DELETE option on the MOVEVOL command for a rewritable tape volume, the system issues the following message to the hardcopy log. This message displays only if all of the objects have been moved from the source cartridge:

```
CBR2165I Tape volume volser has had all objects expired or deleted and
can be returned to the MVS scratch pool.
```

   If you have specified the DELETE option on the MOVEVOL command for a WORM tape volume, the system issues the following message to the hardcopy log. This message displays only if all the objects have been moved from the source cartridge:

```
CBR2173I WORM Tape volume volser has had all objects expired or deleted.
```

   If you have specified the DELETE option on the MOVEVOL command for an optical platter, the system issues the following message to the hardcopy log. This message displays only if all of the objects have been moved from the source cartridge:

```
CBR2153I All objects on volumes volser1 and volser2 have expired,
shelf location shelfloc.
```

____

4. Follow the system prompts to remove the optical cartridge from the I/O station.

____

After the MOVEVOL command completes successfully, OAM deletes the tape or optical cartridge from the OAM inventory. Library-resident optical volumes are ejected from the library. After a volume has been deleted, it is no longer available to any of the OAM members in the OAMplex.

The following is a sample of the F OAM,START,MOVEVOL,*volser*,DELETE command:

```
F OAM,START,MOVEVOL,R8002A,DELETE

CBR1000I OAM START command execution scheduled.
CBR9800I OAM Move Volume Delete starting for volumes R8002A and R8002B.
CBR9850I Move Volume Utility starting for volume R8002A.
CBR9858I Move Volume Utility status for volume R8002A.
Total: 0, Attempted: 0, Successful: 0, Unsuccessful: 0.
CBR9850I Move Volume Utility starting for volume R8002B.
CBR9852I Move Volume Utility processing objects in
storage group GROUP92 for volume R8002B.
CBR2213I No space left in storage group GROUP92.

CBR2217E Enter an optical disk cartridge that is compatible with
DEFAULT MEDIA TYPE 3995 and write compatible with optical drive device
type 3995-SW4 into library P8XB to relieve the out of space condition
in storage group GROUP92.

CBR2550I Optical disk entry into library P8XA scheduled.
CBR4420I Volume table did not contain information for volume R8000B on drive P8XAD1.
CBR4420I Volume table did not contain information for volume R8000A on drive P8XAD1.
*22 CBR4432D Enter storage group name for volumes R8000B and R8000A,
or reply 'U' to assign to scratch.
         22,u
IEE600I REPLY TO 22 IS;U
CBR4401I Volume R8000B mounted on drive P8XAD1.
CBR2100I Volumes R8000B and R8000A entered into library P8XA.
CBR2200I Scratch volumes R8000B and R8000A added to
storage group GROUP92.
CBR9859I Move Volume Utility ending for volumes R8002A and R8002B.
CBR3001A Remove cartridge from I/O station on library P8XA. Place in shelf location ??????.
CBR3122I Volumes R8002B and R8002A were ejected from library P8XA,
shelf location is ??????.
```

If any errors occur while deleting a volume after MOVEVOL completes, OAM might issue additional messages to identify the errors.

Related reading:

- For more information about using the Move Volume utility with the DELETE option, see "Deleting recycled tape and optical volumes from OAM" on page 229.
- For more information about these system messages, see *z/OS MVS System Messages, Vol 4 (CBD-DMO)*.

Statistics provided by the Move Volume utility

The following examples illustrate the messages that the Move Volume utility provides for a variety of conditions.

Normal completion

Figure 28 on page 284 shows an example of the messages when the Move Volume utility completes normally.

```
F OAM,START,MOVEVOL,WG360A

CBR1000I OAM S command execution scheduled.
CBR9850I Move Volume Utility starting for volume WG360A.
CBR9852I Move Volume Utility processing objects in storage group GROUP22
 for volume WG360A.
CBR4401I Volume WG360B mounted on drive P9D2.
CBR9858I Move Volume Utility status for volume WG360A. Total: 5,
Attempted: 5, Successful: 5, Unsuccessful: 0.
CBR9859I Move Volume Utility ending for volumes WG360A and N/A.
```

Figure 28. Example of Messages Returned after a Normal Completion of MOVEVOL

In this example, five objects were identified on the backup source volume and all five objects were successfully moved.

Note:

1. 1 to *n* CBR9852I messages are issued (one for each Object storage group processed).

2. "Successful" and "Unsuccessful" counts always add up to the "Attempted" count on a normal completion.

Normal completion with contending system activity

Figure 29 on page 285 shows an example of the messages provided when the Move Volume utility completes normally, but the utility has detected that other system activity is contending with its processing.

```
F OAM,S,MOVEVOL,BACK01
CBR1000I OAM S command execution scheduled.
CBR9850I Move Volume utility starting for volume
         BACK01.
CBR9852I Move Volume utility processing objects in
         storage group GROUP00 for volume BACK01.
CBR9131I BACK01 CBRHDUPD attempted to update collection
         GROUP00.B, object GROUP00.AAAA.A.STO5.OBJ1 in
         storage group GROUP00. The directory entry
         for the object was already changed.
CBR9131I BACK01 CBRHDUPD attempted to update collection
         GROUP00.B, object GROUP00.AAAA.A.STO5.OBJ2 in
         storage group GROUP00. The directory entry
         for the object was already changed.
CBR9858I Move Volume utility status for volume BACK01.
         Total: 10, Attempted: 10, Successful: 8,
         Unsuccessful: 2.
CBR9859I Move Volume utility ending for volumes BACK01 and N/A.
```

Figure 29. Example of Messages Returned after a Normal Completion with Contention

In this example, 10 objects are identified on the backup source volume and all 10 objects were attempted to be moved; however, only 8 of the 10 attempted objects have been successfully moved. The remaining 2 of the 10 attempted objects have not been successfully moved due to contention with other system activity for those objects.

Limited completion

Figure 30 on page 285 shows an example of the messages provided for "limited" completion of the Move Volume utility where the utility processing was not complete due to contention, errors, and so on.

```
F OAM,S,MOVEVOL,BACK01

CBR1000I OAM S command execution scheduled.
CBR9850I Move Volume utility starting for volume
         BACK01.
CBR9852I Move Volume utility processing objects in
         storage group GROUP00 for volume BACK01.
CBR9852I Move Volume utility processing objects in
         storage group GROUP01 for volume BACK01.
CBR9855I Move Volume utility processing limited for
         volume BACK01.
         Less objects than expected were found in
         collection GROUP01.A.
CBR9857I Move Volume utility status for volume BACK01
         is limited.
CBR9858I Move Volume utility status for volume BACK01.
         Total: 2, Attempted: 1, Successful: 0,
         Unsuccessful: 0.
CBR9859I Move Volume utility ending for volumes BACK01 and N/A.
```

Figure 30. Example of Messages Returned after a Limited Completion

In this example, two objects are identified on the backup source volume but only one object was moved. Because the utility cannot successfully complete due to a discrepancy in the number of objects in a collection being processed, the system cannot determine whether the one object that was attempted to be moved was successfully moved or not.

"Successful" and "Unsuccessful" counts are always zero when message CBR9857I indicates that the status is "limited".

Not available completion

Figure 31 on page 286 shows an example of the messages provided for "not available" completion of the Move Volume utility where the utility processing is not complete due to contention, errors, and so on, and has not been able to determine the number of objects to be moved.

```
  F OAM,S,MOVEVOL,BACK01

  CBR1000I OAM S command execution scheduled.
  CBR9850I Move Volume utility starting for volume
           BACK01.
  CBR9089I No storage groups defined in the active
           configuration.
  CBR9856I Move Volume utility stopping for volume
           BACK01.
  CBR9857I Move Volume utility status for volume BACK01
           is not available.
 CBR9859I Move Volume utility ending for volumes BACK01
 and N/A.
```

Figure 31. Example of Messages Returned after a Not Available Completion

In this example, the utility cannot determine how many objects needed to be moved due to an error condition.

Message CBR9858I is *not* issued when message CBR9857I indicates that the status is "not available" since there are no statistics that can be reported.

Status of the source volume

The source volume is made ineligible for writing during the execution of the Move Volume utility. Unless you specified the DELETE option on the MOVEVOL command, you can reuse the source volume after the utility completes successfully. If you specified the DELETE option on the MOVEVOL command, the source volume is removed from the OAM inventory. If you do not specify the RECYCLE or DELETE option on the MOVEVOL command, the source volume remains in its currently assigned Object or Object Backup storage group, and all volume record fields in the OCDB remain the same.

At the completion of the utility, the WRITABLE status of the source volume is restored to its previous state prior to the execution of the utility. In addition, every object that was moved is deleted from the source volume. The objects are logically deleted whereby OAM no longer maintains directory information on the former location of the object, and physically deleted on rewritable media to reclaim the space that was once occupied by the object.

Status of the objects following the completion of the utility

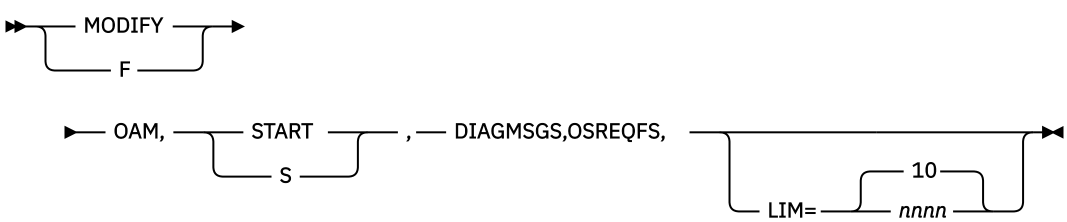
When the Move Volume utility completes, the OAM DB2 object directory table row for each object that was moved is updated to reflect the new object location. Additionally, each object is scheduled for processing during the next OSMC storage management cycle.

The next OSMC storage management cycle selects each object for processing, calculates the expiration date for the object, and updates the expiration date for the volumes on which the objects are written. As a result, you should expect an increase in the number of objects processed during the OSMC storage management cycle following the movement of objects.

Starting temporary diagnostic messages for file system errors originating from an OSREQ request

The OAM processing for the file system sublevel uses other z/OS components including Language Environment (LE) and z/OS UNIX and, when errors occur, these other components may return additional error information to OAM. When the file system sublevel processing originates from OSMC functions and utilities within the OAM address space, this additional error information is provided in error messages and repeated error conditions will cause the OSMC functions and utilities to stop processing. When the file system sublevel processing originates from an application program, the OSREQ return and reason code size limit the amount of error information that can be communicated back to the application program. Because application programs may not stop processing as a result of these error indications, OAM could issue an excessive number of messages containing the additional error information. To prevent this, OAM will, by default, issue no more than ten messages containing the additional error information, regardless of the number of errors. There are some cases, however, when additional error information is available and you may want to temporarily allow additional messages to be displayed to assist in diagnosing errors related to the file system sublevel. The OSREQ return and reason codes may identify file system related conditions or LCS reason codes.

The following MVS command syntax starts the temporary diagnostic messages:



DIAGMSGS,OSREQFS
: Starts (if not currently active) the temporary display of additional diagnostic messages for file system related errors originating from an OSREQ request.

LIM=*nnnn*
: LIM=*nnnn* specifies the approximate number of messages that OAM will issue when the display of these diagnostic messages is active. Valid values for *nnnn* are 1 through 9999. The default value is 10. If the display of additional diagnostic messages is currently active, the specified LIM value overrides the current value. The changed message limit will stay in effect until reset with another `START,DIAGMSGS,OSREQFS` command, or the messages are stopped by the `STOP,DIAGMSGS,OSREQFS` command, or OAM is restarted. When the specified LIM value has been reached, OAM automatically stops the display of the additional diagnostic messages.

It is recommended that the number of diagnostic messages displayed be limited to the minimum required to obtain the needed additional error information.

See *z/OS DFSMSdfp Diagnosis* for a description of the additional error information.

Starting automatic access to backup copies of objects

OAM can obtain a backup copy of an object if the primary copy of the object is resident on a removable media volume that is unavailable for any of the following reasons:

ALL
: Is unavailable for any of the following reasons (nonspecific).

DB2ERROR
: Encounters a DB2 error while attempting to retrieve the object data from the 4 KB, 32 KB, or LOB object storage table.

FSERROR
: Encounters an error while attempting to retrieve the object data from a file system.

LOST
: Is marked lost or is not-defined.

NOTOPER
: Resides in a library that is marked nonoperational.

OFFLINE
Resides in a library that is offline, or pending offline.

UNREAD
Is marked unreadable (possibly due to damage or destruction).

With the use of the access backup function, it is unnecessary for the application to specify the VIEW=BACKUP1 or VIEW=BACKUP2 parameter to obtain the backup copy of the object.

Automatic access to backup copies of objects must be active for one or all of the specific reasons before OAM attempts to obtain the backup copy.

Restriction: This access to backup is limited to retrieval requests that are issued by the OSREQ macro. OSMC does not support access to backup for moving objects from removable media to the DB2 tables.

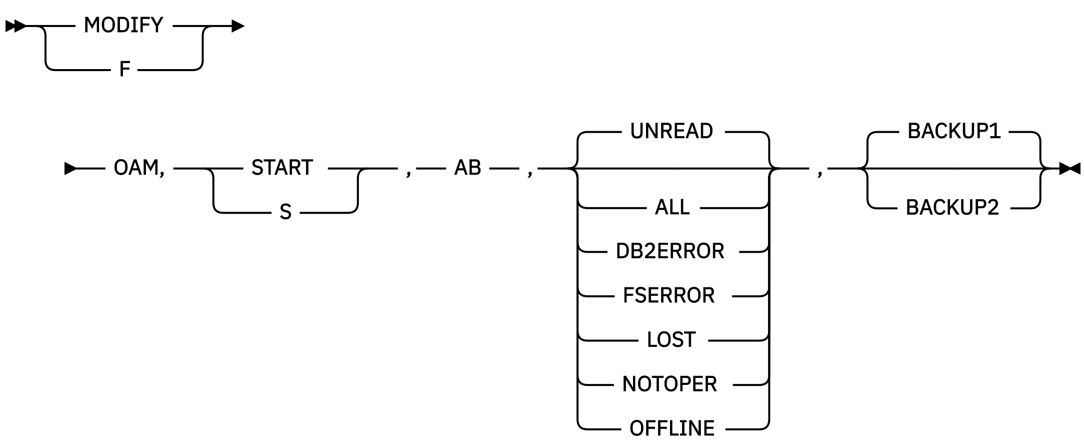
Take the following actions to ensure the retrieval of backup copies of objects. These actions can be performed any time, even when access backup is already active:

- Make a physical inventory of the damaged removable media, noting the volumes residing on the unusable media.
- Repair or replace the hardware as required.
- Obtain all backup removable media volumes containing objects that are related to the objects on the damaged removable media.
- Restore all the OAM DB2 databases from the latest image copy and complete forward recovery through the most recent updates that are available.
- Use ISMF to make any necessary updates to the OAM configuration database for any alterations or updates that are needed for your hardware configuration.
- Use the F OAM,UPDATE,VOLUME,*volser* command to alter the VOLUME table or Tape Volume table in the OCDB, changing the READABLE column value to "N" (NO) for each pair of volumes that is identified as residing on damaged (unreadable) media.

Performing one or more of these steps allows normal application processing to occur until object recovery operations restore the primary copies of objects.

For more information on this function, see "Accessing backup objects automatically" on page 219.

The following MVS command syntax starts the automatic access to backup copies function:



To start processing for the automatic access to backup copies function:

1. Enter the following command:

   ```
   F OAM,START,AB,reason,BACKUP1 | BACKUP2
   ```

   The following are valid values and descriptions for the *reason* keyword:

ALL
When a retrieve for an object is attempted and the optical or tape volume on which the object resides is not available for any of the following reasons, the backup copy of the object is retrieved.

DB2ERROR
If a DB2 error occurs while OAM is retrieving object data from the 4 KB, 32 KB, or LOB object storage table and the first or second backup copy exists, OAM retrieves the object data from the backup copy. This function allows access to backup copies of objects that reside on removable media (optical or tape) when the DB2 resident data is unavailable, such as during the recovery of DB2 tables.

Restriction: The object directory entry is necessary for OAM to proceed with any object request. If a DB2 error occurs while OAM attempts to retrieve the object directory entry, OAM does not retrieve the backup copy of the object. Without the object directory information, OAM cannot determine the primary or backup location of the object.

FSERROR
If a file system error occurs while OAM is retrieving object data from the file system and the first or second backup copy exists, OAM retrieves the object from the backup copy. This function allows access to backup copies of objects that reside on removable media (optical or tape) when the file system resident data is unavailable (because, for example, the file system is unmounted or undergoing maintenance procedures such as backup of the file system data).

LOST
When a retrieve for an object is attempted, and the optical or tape volume on which the object resides is marked lost or is not-defined, the backup copy of the object is retrieved.

NOTOPER
When a retrieve for an object is attempted and the optical or tape volume on which the object resides is in a library that is marked nonoperational, the backup copy of the object is retrieved.

OFFLINE
When a retrieve for an object is attempted and the optical or tape volume on which the object resides is in a library that is offline or pending offline, the backup copy of the object is retrieved.

UNREAD
UNREAD is the default. When a retrieve for an object is attempted and the optical or tape volume on which the object resides is marked not readable, the backup copy of the object is retrieved.

The following are the valid, optional parameters for specifying which backup copy should be accessed:

BACKUP1 | BACKUP2
When a primary copy of an object that is read by an application is not available for the specified reason, and BACKUP1 is specified on the START command for automatic access to backup, an attempt is made to retrieve the object from the first backup copy of the object. If BACKUP2 is specified, an attempt is made to retrieve the object from the second backup copy of the object. BACKUP1 is the default if neither BACKUP1 nor BACKUP2 is specified.

___

2. The system issues the following message:

```
CBR1000I OAM START command execution scheduled.
```

___

3. Once the access backup processing starts, the system issues the following message:

```
CBR1090I OAM Access Backup processing started for reason
using the {1st | 2nd} backup copy.
```

Where *reason* can be one of the following values:

- DB2 OBJECT TABLE ERRORS
- FILE SYSTEM ERRORS
- LOST VOLUMES

- NOT OPERATIONAL LIBRARIES
- OFFLINE LIBRARIES
- UNREADABLE VOLUMES

____

4. Access backup processing remains active until a STOP,AB,*reason* command is issued. The system issues the following message:

```
CBR1091I OAM Access Backup processing stopped for reason.
```

____

5. If access backup processing is active and the operator tries to issue another START,AB,*reason* command, the system issues the following message:

```
CBR1092I OAM Access Backup processing already started for reason
using the [1st | 2nd] backup copy.
```

____

The following is a sample of enabling Access Backup for UNREADABLE VOLUMES with the optional parameter BACKUP2 specified:

```
F OAM,START,AB,UNREAD,BACKUP2

CBR1000I OAM START command execution scheduled.
CBR1090I OAM Access Backup processing started for UNREADABLE VOLUMES using the
             2nd backup copy.
```

Note: If no automatic access to backup keywords have been specified on the SETOPT statement in the CBROAMxx parmlib member, then stopping OAM and starting OAM does not affect the status of automatic access to backup. If access to backup has been started and OAM is stopped, access to backup is active when OAM is started again.

Note: You can also configure automatic access to backup in the CBROAMxx PARMLIB SETOPT statement using keywords which follow very closely the convention that the MODIFY OAM,START,AB command uses.

Varying optical drives and libraries

You can vary optical drives and optical libraries online or offline, which means that you can control whether the system can access the optical drive or optical library.

If a library or drive is varied online to an instance of OAM, no associated libraries or drives in the same 3995 subsystem can be online to any other instance of OAM in the OAMplex.

The vary offline command requires that the library or drive being taken offline be controlled by the OAM targeted for the vary request.

Tip: Changing the offline status of the optical library does not affect the online/offline status of the library-resident optical drives that are contained within the library. Use the VARY SMS command to vary an optical drive or optical library online or offline.

Varying an optical drive online or offline

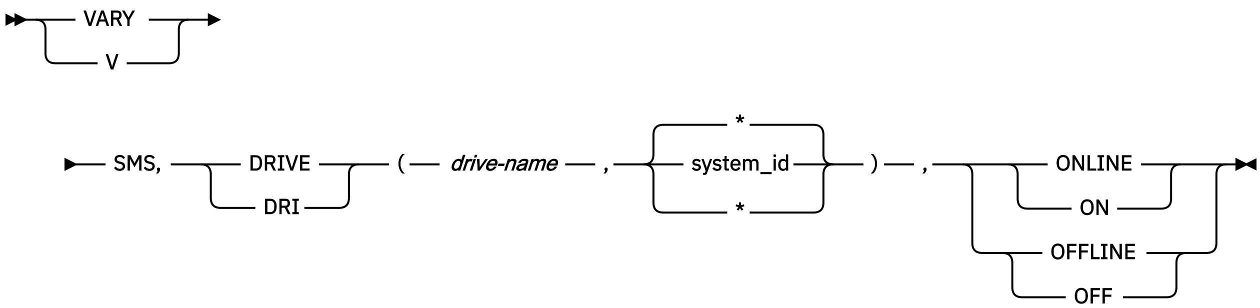
Before OAM can allow a drive to be brought online to an instance of OAM, 3995 drive processing must verify that the drive is not online to another instance of OAM in the OAMplex, or that the OAM that controls the drive has terminated or does not have a valid XCF member defined.

Drive vary processing must check the status of the optical library to which this drive is physically attached. If this drive is brought online, the library where it resides must not be online to another OAM in the OAMplex. It also checks the status of all the other libraries associated with the drive and the drives associated with those libraries.

Drive vary processing must also check the status of other drives in the optical library to which this drive is attached. If this drive is brought online, no other drives in the library where the drive resides can be online to another instance of OAM in the OAMplex.

SMS validates the specified system ID that is targeted for the vary request; it also verifies that the specified drive is defined as connected to the target system ID in the ACDS.

The following SMS command syntax varies optical drives online and offline:



DRIVE(*drive-name*)
Specifies the name of the optical drive to be varied online or offline. If the name is not specified or the specified drive is not defined in the SMS configuration, an error message is displayed.

*****
Indicates that the target of the drive vary request is the current OAM.

system-id
Specifies the MVS system that is the target of the vary request.

For optical drives, only one system ID can be specified.

ONLINE
Specifies that the optical drive is varied online.

OFFLINE
Specifies that the optical drive is varied offline.

Here is an example of the command to vary an optical drive online:

```
VARY SMS,DRIVE(drive-name,system-id),ONLINE
```

This results in MODIFY OAM,VARY,drivename,ONLINE

Here is an example of the command to vary an optical drive offline:

```
VARY SMS,DRIVE(drive-name,system-id),OFFLINE
```

This results in MODIFY OAM,VARY,drivename,OFFLINE

Tip: You can demount an optical disk cartridge on an operator-accessible drive by varying the drive offline.

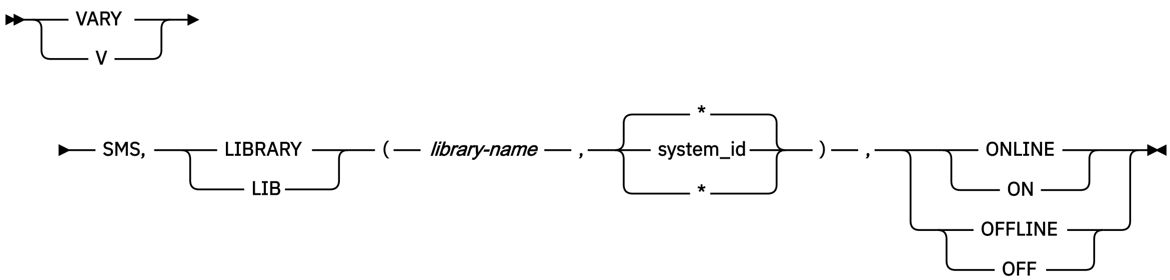
Varying a real optical library online or offline

Before OAM can allow a library to be brought online to an instance of OAM, 3995 library processing verifies that the 3995 optical library is not online to another instance of OAM in the OAMplex, or that the OAM that owns the library is failed (the XCF member name is not defined or is no longer valid for the OAMplex).

Library vary processing must also check drive status of the optical drives that are physically attached to this library and make certain that the drives are not online to another instance of OAM, or that the OAM that controls the drive has terminated or does not have a valid XCF member defined. It also checks the status of associated libraries and controller and the status of the drives in those libraries.

SMS validates the specified system ID targeted for the vary request, and verifies that the specified drive is defined as connected to the target system ID in the ACDS.

The following SMS command syntax varies optical libraries online and offline:



LIBRARY(*library-name*)
: Specifies the name of the real optical library that is varied online or offline. If the name is not specified or the specified library is not defined in the SMS configuration, an error message is displayed.

  Restriction: Pseudo libraries cannot be varied online or offline. If a pseudo library name is used to attempt to vary a library online or offline, an error message is displayed.

system-id
: Specifies the MVS system that is the target of the vary request.

  For optical libraries, only one system ID can be specified.

*****
: Indicates that the target for the library vary request is the current OAM system.

ONLINE
: Specifies that the optical library is varied online.

OFFLINE
: Specifies that the optical library is varied offline.

Recommendation: To change the system where an optical library is online, the library and drives must be first varied offline to the system where they are currently online. After you successfully bring the 3995 device addresses online to z/OS, you can vary online the optical library and drives to OAM.

Here is an example of the command to vary an optical library online:

```
VARY SMS,LIBRARY(library-name,system-id),ONLINE
```

This results in MODIFY OAM,VARY,libname,ONLINE.

Here is an example of the command to vary an optical library offline:

```
VARY SMS,LIBRARY(library-name,system-id),OFFLINE
```

This results in MODIFY OAM,VARY,libname,OFFLINE.

Varying an optical library offline does not occur immediately if the drives are active or on the queue to be processed. The vary command activates immediately but does not complete until after the activity on the drives has completed. Once the library is pending offline, all library specific activity to that library fails. Write activity to a storage group that spans multiple libraries including the offline library is scheduled to one of the other libraries.

Entering an optical disk into an optical library

When OAM needs an additional optical disk volume to satisfy an out-of-space condition for a particular storage group which contains a real library, the following messages are displayed:

- For 3995 libraries:

```
CBR2213I No space left in storage group storage-group-name.

CBR2217E Enter an optical disk cartridge that is compatible with the DEFAULT
         MEDIA TYPE library-default-media-type into library library-name
         to relieve the out of space condition in storage group
         storage-group-name.

CBR2550I Optical disk entry into library library-name scheduled.

CBR4401I Volume volser1 mounted on drive drive-name.

CBR2100I Volumes volser1 and volser2 entered into library.
```

You can enter the following in response to message CBR2217E:

- An unlabeled optical disk
- A scratch optical disk
- An optical disk whose volumes are assigned to the requested storage group, writable, not full, and not write-protected

If all slots in the optical library are either occupied by an optical disk cartridge or reserved for an optical disk cartridge mounted on one of the library optical drives, the optical library is full. This condition is detected by displaying the optical library status and checking the CBR1110I message field EMP SLT (empty slot) for a value of zero. The system automatically runs library space management for this library before requesting an additional optical disk cartridge.

Recommendation: When loading an optical disk cartridge into a library with 3995 optical disk drives, the media type of the cartridge must be consistent with the default media type for the library. If the media type of the cartridge is not consistent with the default media type, the cartridge is ejected. If the media is being isolated for specific applications, assign the cartridges to a storage group. The cartridges cannot be used as scratch volumes.

If the media type entered is compatible with the library default media type but is not write compatible with the drives in the library, the out-of-space condition is not relieved. An example of this scenario might be:

- A 3995-C*xx* library containing 3995-SW3 drives
- With a library default media type of 3995
- And the volume entered is a single-density WORM cartridge

In this case, the out-of-space condition still exists because the 3995-SW3 drives cannot write to a single-density WORM cartridge.

In an OAMplex, if a shelf-resident volume is entered into a library that is not known to an OAM in the OAMplex, that OAM removes it from its configuration and issues a message that the volume is no longer known. After that time, any read for that volume on that OAM fails with a volume unknown error. If the volume is ejected and assigned back to a pseudo library that is known to OAM, the volume is added back to the configuration.

Note: Because WORM optical volumes that are full or have very little free space are not useful as scratch volumes, the operator is notified, by a message, if the kilobytes free are less than the SCRENTRYTHRESHOLD parameter. The message contains the number of kilobytes that are free and the percentage of free space that this represents on the volume. The operator can choose to fail the cartridge entry process, causing the cartridge to be ejected from the library.

For more information on checking the status of an optical library, see "Displaying library detail status" on page 332.

Entering an unlabeled optical disk into a 3995 optical library

To enter an unlabeled optical disk into a 3995 optical library:

1. Put the optical disk cartridge into the optical library input/output station. The system issues the following message:

```
CBR2550I Optical disk entry into library library-name scheduled.
```

____

2. The optical disk moves from the input/output station when an optical drive becomes available. The system issues the following message for each side of the optical disk cartridge.

```
CBR3381I Volume mounted on drive on drive-name contains an unrecognized format.
```

____

3. The system issues the following message and waits for a reply:

```
CBR4438D Volume in drive-name has unrecognized media format.
         Reply 'F' to format or 'C' to cancel.
```

   This message is unique to a 3995 library. It is issued when a volume has been entered that is either unlabeled or of a format not known to the library. A reply of 'F' causes the volume to be formatted and any data existing on the volume is destroyed. Replying 'C' at this point causes the formatting processing to stop, leaving only the one side of the optical cartridge formatted.

   Attention: Formatting a rewritable optical disk volume is a time consuming process. Do not interrupt this process by assuming that OAM is inactive during this time frame. OAM issues a completion message when this process is finished.

____

4. To continue the label process, reply with an 'F'. To cancel the label process, reply with an 'C'. If a reply of 'F' is entered, the system issues the following message and waits for a reply:

```
CBR4412D Enter VOLSER for volume on drive drive-name in library library-name.
```

____

5. Provide a 1 to 6 character unique volume serial number. The system issues the following message to verify the volume serial number. To accept the volume serial number, reply with a 'U'. A reply of 'R' allows you to enter a different volume serial number.

```
CBR4424D VOLSER for unlabeled volume in drive drive-name is volser. Reply 'U'
         to use this volser, or 'R' to retry.
```

____

6. The system issues the following message and waits for a reply:

```
CBR4439D Enter VOLSER for opposite side of volume volser in drive drive-name.
```

____

7. Provide a 1 to 6 character unique volume serial number. The system issues the following message to verify the volume serial number. To accept the volume serial number, reply with a 'U'. A reply of 'R' allows you to enter a different volume serial number.

```
CBR4424D VOLSER for unlabeled volume in drive drive-name is volser. Reply 'U'
         to use this volser, or 'R' to retry.
```

____

8. Once the system verifies the volume serial number to be used on the cartridge, the system issues the following message and waits for a reply:

```
CBR4406D Enter owner information for volume volser on drive drive-name.
```

____

9. Provide 1 to 64 characters of owner identification. The system issues the following message:

```
CBR4432D Enter storage-group-name for volumes volser1 and volser2, or reply ‘U’
         to assign to scratch.
```

10. If the volumes are to be assigned to scratch status, reply ‘U’ to this message; otherwise, reply with the name of the object storage group or object backup storage group to which the volumes are to be assigned. The system issues the following message:

```
CBR4401I Volume volser2 mounted on drive drive-name.
```

11. The system issues the following message:

```
CBR2100I Volumes volser1 and volser2 entered into library: library-name.
```

The two volumes on the optical disk are added to a storage group or are assigned to scratch status and are available for use. In the message text, *volser1* and *volser2* are replaced by the volume serial numbers you entered for the optical volumes.

For further information about commands from the 3995 dynamic console, see the *3995 Operator Guide for C-Series Models*.

Entering a labeled optical disk into an optical library

To put a labeled optical disk into the optical library without prompting from the system:

1. Put the optical disk cartridge into the optical library input/output station. The system issues the following message:

```
CBR2550I Optical disk entry into library library-name scheduled.
```

   Requirement: When loading an optical disk cartridge into a library with 3995 optical disk drives, the media type of the cartridge must be consistent with the default media type for the library and must be compatible with the library resident drives. If the media type of the cartridge is not consistent with the default media type or compatible with the library resident drives, the cartridge is ejected. If the media is being isolated for specific applications, assign the cartridges to a storage group. The cartridges cannot be used as scratch volumes.

2. The optical disk moves from the input/output station when an optical drive becomes available. The system reads and verifies the volume label on each side of the optical disk. The system issues the following messages:

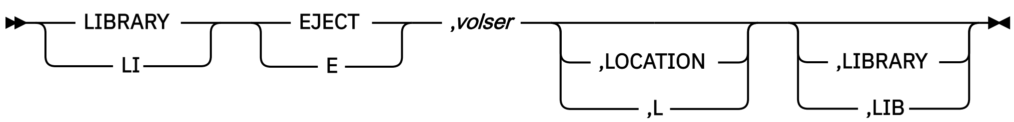
```
CBR4420I Volume table did not contain information for volser1 on drive
         drive-name.

CBR4420I Volume table did not contain information for volser2 on drive
         drive-name.

CBR4432D Enter storage group name for volumes volser1 and volser2,
         or reply ‘U’ to assign to scratch.

CBR4401I Volume volser1 mounted on drive drive-name.

CBR2100I Volumes volser1 and volser2 entered into library library-name.
```

Ejecting an optical disk

Use the MVS LIBRARY EJECT command to eject a specific optical disk from an optical library. This command requires that the optical volume specified be controlled by the instance of OAM on which the command was entered. If the member name associated with the volume serial number is not the member name for this instance of OAM, the command fails and message CBR1068I is issued.

In an OAMplex, if a shelf-resident volume is entered into a library that is not known to an OAM in the OAMplex, that OAM removes it from its configuration and issues a message that the volume is no longer known. After that time, any read for that volume on that OAM fails with a volume unknown error. If the volume is ejected and assigned back to a pseudo library that is known to OAM, the volume is added back to the configuration.

The following command syntax ejects a specific optical disk:



The following describes the values of the options for the EJECT command:

volser
: Specifies the volume serial number of one of the two optical volumes on the optical disk that is ejected.

LOCATION
: Prompts the operator to specify the new shelf location information for the ejected optical disk volumes.

LIBRARY
: Prompts the operator to specify a pseudo library name to which the ejected volume will be assigned.

To eject an optical disk from an optical library, enter the following command:

```
LI EJECT,volser,LOCATION,LIBRARY
```

An optical disk can arrive in the input/output station without an operator request in the following ways:

- The system can eject the optical disk as a result of the Library Space Manager utility running on that library.

  The system ejects an optical disk volume if there are no scratch optical volumes in the optical library, no empty slots in the optical library, and the system is attempting to write to the storage group that has an out-of-space condition.
- The system ejects an optical disk volume if the volume is expired.
- The system ejects an optical disk volume if an unknown volume results in an error during volume entry sequence.
- A storage administrator can specify an EJECT line operator next to an optical volume serial number on an ISMF Mountable Optical Volume List panel.
- The system responds to a REMAP command against the library control inventory.

Tip: You can demount an optical disk on an operator-accessible drive by varying the drive offline.

Specifying the shelf location

If a shelf location has not been specified previously for the optical disk being ejected or a new shelf location was requested on EJECT command, the system issues the following messages and waits for the appropriate responses:

```
CBR1000I OAM EJECT command execution scheduled.
```

```
CBR2600A Specify shelf location for volumes volser1 and volser2.
```

Provide 1 to 32 characters of shelf information. The system issues the following messages:

```
CBR2603A Specify pseudo library name for volume volser1.
```

Provide the name of the associated pseudo library for this volume. The system issues the following messages:

```
CBR3001A Remove cartridge from I/O station on library library-name. Place in
         shelf location shelf-location.
```

```
CBR3122I Volumes volser1 and volser2 were ejected from library library-name,
         shelf location is shelf-location.
```

Associating pseudo libraries

When an optical volume is ejected from a library, it must be assigned to a pseudo library. In previous releases, shelf resident volumes were associated with a pseudo optical library that represented volumes of that media type. Pseudo libraries do not have a device type association.

If the volume being ejected already has a pseudo library name associated with it in its volume record, that pseudo library continues to be used unless it is overridden with the LIBRARY keyword on the EJECT command. If the volume being ejected does not already have a valid pseudo library associated with it, the default pseudo library associated with the library from which the volume was ejected can be used, if one exists, unless it is overridden with the LIBRARY keyword on the EJECT command. If the optical volume being ejected does not already have a valid pseudo library associated with it, or the library from which the volume was ejected does not have a default pseudo library in its definition, or the LIBRARY keyword was specified on the EJECT command, the system issues message CBR2602A or CBR2603A, or both.

```
CBR2602A Eject pending for volser in r-library-name. Default pseudo library
         is p-library-name. Reply 'U' to use, or 'R' to respecify.
```

```
CBR2603A Specify pseudo library name for volume volser.
```

If the volume being ejected has an invalid pseudo library associated with its volume record and the library from which the volume is being ejected has a default pseudo library in its SCDS definition, message CBR2602A is issued. This message asks if the default is to be used or if another pseudo library name is requested.

If the library from which the volume is being ejected does not have a default pseudo library in its SCDS definition, or "R" was replied to message CBR2602A, message CBR2603A is issued. This message requests a pseudo library name to assign the volume to when it becomes shelf resident.

If a pseudo library name specified in response to message CBR2603A is not valid in the current ACDS, or the volume record has an invalid pseudo library name associated with it, message CBR2604I is issued, followed by message CBR2603A.

```
CBR2604I Volume volser cannot be assigned to pseudo library p-library-name, it is
         not a valid pseudo library definition in the active SMS configuration.
```

```
CBR2603A Specify pseudo library name for volume volser.
```

For more information regarding pseudo libraries, see "Pseudo optical library concept" on page 42.

Removing the optical disk cartridge

When you receive the following message, remove the optical disk cartridge from the input/output station of the specified optical library and return it to the shelf location indicated:

```
CBR3001A Remove cartridge from I/O station on library library-name. Place in
         shelf location shelfloc.
```

Attention: Upon ejection of a cartridge, immediately remove it from the input/output station; otherwise, performance is degraded.

Mounting an optical disk on an operator-accessible drive

When the system requests a shelf-resident volume, the following message is displayed requesting you to mount an optical disk cartridge on an operator-accessible optical drive:

```
CBR4400A Mount volume volser on drive drive-name. Shelf location is shelfloc.
```

If the *volser* is ??????, the request is for an unlabeled optical volume. Locate an unlabeled optical disk cartridge and load it on drive *drive-name*.

If a volume serial number is specified in *volser*, the request is for an already labeled optical volume. Locate the optical disk cartridge and load it into the drive *drive-name*. The system issues the following message:

```
CBR4401I Volume volser mounted on drive drive-name.
```

The volume is now ready for the system to use.

Recommendations:

1. You can also mount an optical disk on an operator-accessible optical drive without waiting for a request from the system. To mount the optical disk, vary the operator-accessible optical drive offline, load the optical disk cartridge into the drive, and vary the drive online.
2. The system places a response time limitation of five minutes from the time the operator mount message is received on the console to the time the mount is completed. If the mount is not completed within the allotted time, the operator has the option of canceling or retrying the optical disk mount.

To label a disk on an operator-accessible drive, see "Labeling an optical disk on a 3995 operator-accessible drive" on page 299.

Demounting and removing an optical disk cartridge from an operator-accessible drive

To demount an optical disk cartridge from an operator-accessible drive, you must vary the drive offline using the OAM VARY command. The cartridge is demounted so that the operator can remove the cartridge.

Labeling an optical disk on a 3995 operator-accessible drive

When there are no scratch optical volumes on the shelf, and a storage group which contains a pseudo library is out of space, the following messages are displayed requesting you to label an unlabeled optical disk on the operator-accessible optical drive:

```
CBR2213I No space left in storage group storage-group-name.

CBR2212E Use the OAM LABEL command to label optical disks for shelf use
         to relieve the out of space condition in the storage group
         storage-group-name.


CBR1000I OAM L command execution scheduled.

CBR4400A Mount volume ?????? on drive drive-name. Shelf location is ??????

CBR4419I Previously labeled volume volser1 was mounted on drive drive-name.

CBR4423D Enter shelf information for volume volser1 on drive drive-name.


CBR4430A Remove and flip cartridge on drive drive-name.

CBR4419I Previously labeled volume volser2 was mounted on drive drive-name.

CBR4432D Enter storage group name for volumes volser1 and volser2,
         or reply 'U' to assign to scratch.

CBR4401I Volume volser2 mounted on drive drive-name.

CBR2102I LABEL function complete for volumes volser1 and volser2.
```

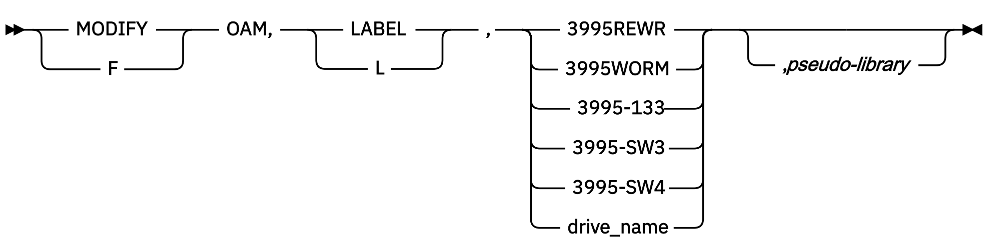
Tip: You can also use a library optical drive to label an optical disk. See "Entering an optical disk into an optical library" on page 292 to label an optical disk in an optical library.

Use the OAM LABEL command to label optical volumes on operator-accessible optical drives. OAM rejects the LABEL command if there is no operator-accessible optical drive online and operational.

Note: Because WORM optical volumes that are full or have very little free space are not useful as scratch volumes, the operator is notified, by a message, if the kilobytes free are less than the SCRENTRYTHRESHOLD parameter. The message contains the number of kilobytes that are free and the percentage of free space that this represents on the volume. This message gives the operator the opportunity to fail the cartridge entry process, causing the cartridge to be ejected from the library.

In an OAMplex, if a shelf-resident volume is entered into a library that is not known to an OAM in the OAMplex, that OAM removes it from its configuration and issues a message that the volume is no longer known. After that time, any read for that volume on that OAM fails with a volume unknown error. If the volume is ejected and assigned back to a pseudo library that is known to OAM, the volume is added back to the configuration.

The following command syntax labels an optical disk on an operator-accessible optical drive:



The following describes the values of the options for the LABEL command:

3995REWR
: Refers to a request for a single-density, rewritable optical disk media to be labeled on an IBM 3995-131 operator-accessible optical disk drive.

3995WORM
Refers to a request for a single-density, WORM optical disk media to be labeled on an IBM 3995-132 operator-accessible optical disk drive

3995-133
Refers to a request for a single- or double-density, WORM or rewritable, optical disk media to be labeled on an IBM 3995-133 operator-accessible optical disk drive.

3995-SW3
Refers to a request for a double- or quad-density capacity, WORM or rewritable optical disk media to be labeled on an IBM 3995-SW3 operator-accessible optical disk drive.

3995-SW4
Refers to a request for a quad- or 8*x*-density capacity, WORM or rewritable optical disk media to be labeled on an IBM 3995-SW4 operator-accessible optical disk drive.

drive_name
Refers to a IBM 3995 operator-accessible drive on which the label command is processed. The optical media being labeled must be compatible with the drive requested for reading and writing purposes. The name of the optical drive must be defined in the active SCDS configuration.

pseudo-library
Refers to the name of the pseudo library whose drives are to be considered for the label request, and the pseudo library with which the newly labeled volume will be associated. If this keyword is not specified, the default is the pseudo library associated with the operator-accessible drive that is performing the label function. If a pseudo library name specified on the LABEL command is invalid for the current ACDS, message CBR1305I is issued and the command fails.

Attention: Due to a hardware restriction, inserting a double-, quad-, or 8*x*-density, rewritable cartridge into a single-density, WORM (3995-132) or rewritable (3995-131) operator-accessible drive can result in the cartridge being demounted with no error message posted on the library service panel or to the host.

If the label request is directed to a specific drive, OAM verifies that the drive requested is controlled by the OAM on which the command was entered. If this criteria is not satisfied, the request fails and message CBR1068I is issued.

If a pseudo-library name is specified on the LABEL command, a media type must be specified (3995REWR, 3995WORM, 3995-133, 3995-SW3, or 3995-SW4) because pseudo-libraries no longer require a device type affinity, so mixed devices and media types might be associated with a pseudo-library. If the drive name is specified, specification of a pseudo-library is ignored.

If only a media type is specified on the LABEL command, the command is processed on any operator-accessible drive that can use the specified media type and that is connected to the OAM where the command was entered. Labels are processed on the system where the command is entered. The command must be issued on a system with an online and operational operator-accessible drive that can use the media type.

To write the label on two volumes of an optical disk on an operator-accessible drive:

1. Enter the following command:

   ```
   F OAM,LABEL,{3995REWR|3995WORM|3995-133|3995-SW3|3995-SW4,drive_name},pseudo-lib
   ```

   ____

2. The system issues the following message:

   ```
   CBR1000I OAM LABEL command execution scheduled.
   ```

   ____

3. When an operator-accessible optical drive becomes available, the system issues the following messages and waits for you to mount an optical disk cartridge:

   ```
   CBR4400A Mount volume ?????? on drive drive-name. Shelf location is ??????.
   ```

____

4. Put an unlabeled optical disk cartridge into the operator-accessible optical drive. The system issues the following message:

```
CBR3381I Volume mounted on drive drive-name contains an unrecognized format.
```

____

5. The system issues the following message and waits for a reply:

```
CBR4438D Volume in drive drive-name has unrecognized media
         format.  Reply ‘F’ to format or ‘C’ to cancel.
```

This message is unique to a 3995 library. It is issued when a volume has been entered that is either unlabeled or of a format not known to the library. A reply of ‘F’ causes the volume to be formatted and any data existing on the volume is destroyed.

____

6. To continue the label process, reply with an ‘F’. To cancel the label process, reply with an ‘C’. If a reply of ‘F’ is entered, the system issues the following message and waits for a reply:

```
CBR4405D Enter VOLSER for volume on drive drive-name.
```

____

7. Provide a 1 to 6 character unique volume serial number. The system issues the following message to verify the volume serial number. To accept the volume serial number, reply with a ‘U’. A reply of ‘R’ allows you to enter a different volume serial number.

```
CBR4424D Volser for unlabeled volume in drive drive-name is volser.
         Reply ‘U’ to use this volser, or ‘R’ to retry.
```

____

8. Once the system verifies the volume serial number to be used on the cartridge, the system issues the following message and waits for a reply:

```
CBR4406D Enter owner information for volume volser on drive drive-name.
```

____

9. Provide 1 to 64 characters of owner identification. The system issues the following message and waits for a reply:

```
CBR4423D Enter shelf information for volume volser on drive drive-name.
```

____

10. Provide 1 to 32 characters of shelf information. The system issues the following message:

```
CBR4430A Remove and flip cartridge on drive drive-name.
```

11. Remove the optical disk cartridge, flip it over, and reinsert it into the operator-accessible optical drive. The system issues the following message:

```
CBR3381I Volume mounted on drive drive-name contains an unrecognized format.
```

____

12. The system issues the following message and waits for a reply:

```
CBR4438D Volume in drive drive-name has unrecognized media format. Reply ‘F’
         to format or ‘C’ to cancel.
```

This message is unique to a 3995 library. It is issued when a volume has been entered that is either unlabeled or of a format not known to the library. A reply of ‘F’ causes the volume to be formatted and any data existing on the volume is destroyed.

13. To continue the label process, reply with an 'F'. To cancel the label process, reply with an 'C'. If a reply of 'F' is entered, the system issues the following message and waits for a reply:

```
CBR4439D Enter volser for opposite side of volume volser in drive
         drive-name.
```

14. Provide a 1 to 6 character unique volume serial number. The system issues the following message to verify the volume serial number. To accept the volume serial number, reply with a 'U'. A reply of 'R' allows you to enter a different volume serial number.

```
CBR4424D Volser entered for unlabeled volume in drive drive-name is volser.
         Reply 'U' to use this volser, or 'R' to retry.
```

15. Once the system verifies the volume serial number to be used on the cartridge, the system issues the following message and waits for a reply:

```
CBR4432D Enter storage-group-name for volumes volser1 and volser2, or
         reply 'U' to assign to scratch.
```

16. If the volumes are to be assigned to scratch status, reply 'U' to this message; otherwise, reply with the name of the object storage group or object backup storage group to which the volumes are to be assigned. The system issues the following messages:

```
CBR4401I Volume volser2 mounted on drive drive-name.

CBR2102I LABEL function complete for volumes volser1 and volser2.
```

The two volumes on the optical disk are added to a storage group or are assigned to scratch status and are available for use. In the message text, *volser1* and *volser2* are replaced by the volume serial numbers that you entered for the optical volumes.

For further information about commands from the 3995 dynamic console, see the *3995 Operator Guide for C-Series Models*.

Relabeling a 3995 optical disk volume

The relabel command is used to allow the operator to rename a volume serial number on a previously defined 3995 optical disk volume. As a preventive measure to keep from losing all active primary or backup copies of objects on the 3995 disk volume that is being relabeled, the following conditions apply:

- The optical disk volume being relabeled cannot be an Object Backup volume.
- There can be no active primary copy of an object on the volume.
- There can be no write requests scheduled for the volume.
- In an OAMplex, the requested volume must be controlled by the OAM on which the command was entered.

To erase all copies of objects on the 3995 optical disk volume, submit the OAM utility job to reformat the volume. See "Reformatting a 3995 optical disk" on page 303 for more information on this utility.

The following command syntax relabels an optical disk volume:

```
►►─┬─ MODIFY ─┬─►
   └─ F ──────┘

►─ OAM, ─┬─ RELABEL ─┬─ , ─ old_volser ─ , ─ new_volser ─┬──────────────────┬─►◄
         └─ RL ──────┘                                   └─ , ─ drive_name ─┘
```

The following describes the values of the options for the RELABEL command:

old_volser
: This a required parameter that specifies the current volume serial number of the OAM volume.

new_volser
: This is a required parameter that specifies the new volume serial number to be assigned to the optical volume.

drive_name
: This is an optional parameter that specifies a write compatible operator-accessible drive for processing this relabel request if the volume resides outside a 3995 optical disk library. If *drive_name* is not supplied or if the volume is library-resident, OAM selects an optical drive in the ACDS that is capable of processing this request.

To relabel a requested optical disk volume and direct the request to a specific operator-accessible drive, enter the following command:

```
F OAM,RELABEL,old_volser,new_volser,drive_name
```

This system issues the following messages:

```
CBR1000I OAM RELABEL command execution scheduled.

CBR4460I Volume old_volser on drive-name has been relabeled to new_volser.

CBR2822I RELABEL function completed for volume old_volser to new_volser.
```

When the command is accepted, if the requested volume is shelf-resident and not mounted on the selected drive, OAM asks the operator to mount the requested volume on the selected optical drive. If the requested volume is library-resident, OAM mounts the volume on a library drive. Following successful completion of this processing, OAM performs the following functions:

- deletes the row in the DB2 Volume Table row for the old volume serial number
- inserts a new row in the DB2 Volume Table for the new volume serial number
- updates the row of the opposite volume in the DB2 Volume Table with the new volume serial number
- issues message CBR4460 to inform the operator that the relabeling of the 3995 optical disk volume has completed

Reformatting a 3995 optical disk

Note: The OAMUTIL REFORMAT utility applies only in a classic OAM configuration. It cannot be used in a multiple OAM configuration because the Optical level of the OAM storage hierarchy is not supported in a multiple OAM configuration.

Use the OAM reformat utility (the TSO/E command OAMUTIL) to perform various tasks against a 3995 optical disk cartridge to reclaim usable space on the cartridge. To invoke this utility, run the CBRSAMUT SAMPLIB job or issue a TSO/E command to start the utility. (See "CBRSAMUT" on page 466 for a sample job that can be used within your installation.) Use this utility to perform the following tasks:

- Reformat one or both sides of a 3995 optical disk cartridge.
- Reformat and rename the volume serial number of one or both sides of an optical disk cartridge.

- Return the volumes back to the SCRATCH storage group (only when there is a request for both sides of the optical disk to be reformatted) to be used for subsequent write requests.

The reformat utility can be run regardless of whether the volume is inside or outside a 3995 optical library, the volume belongs to an Object or an Object Backup storage group, or the media is rewritable or WORM. Reformatting rewritable media reclaims the used space; however, reformatting WORM media cannot reclaim the used space.

For reformatting volumes belonging to an Object storage group, this utility can be run conditionally (using the NOFORCE parameter) or unconditionally (using the FORCE parameter). For reformatting volumes belonging to an Object Backup storage group, this utility can only be run unconditionally (using the FORCE parameter).

When a *conditional* request is submitted for an OAM PRIMARY optical volume, OAM checks to see if any active primary copies of objects on the volume exist before performing the reformat. If any active object is found, the request fails.

When a *conditional* request is submitted for an OAM BACKUP optical volume, the request fails for the reformat.

This utility can only be run *unconditionally* (using the FORCE parameter) for optical volumes that belong to an OAM BACKUP storage group.

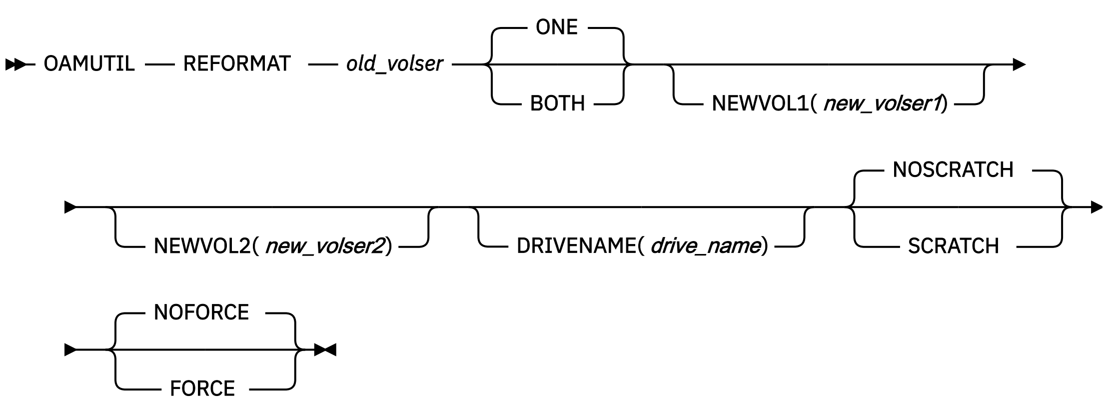
When an *unconditional* request is submitted, there is no verification of whether there are any non expired objects still on the volume before the operation being performed. The volume is reinitialized if the OAM environment allows it at the time of the request. Therefore, it is recommended, that before submitting an unconditional reformat request, you should select all Object Directory entries for objects that exist on the requested volume (using SPUFI) to verify that the object can be deleted. Or, you can use the Move Volume utility to move all the objects off the requested volume before performing the reformat. When an unconditional reformat request is ran successfully, all objects on the requested volume, regardless if the objects are owned by OAM, are all erased or discarded.

Attention: If you use the FORCE parameter or the unconditional form of the format command, be aware that when you request a reformat of the optical disk, there is no reversal. If OAM issues any error messages indicating DB2 or other problems, you are responsible for ensuring that all objects that resided on this volume are no longer referenced in the OAM object directory table.

For a reformat command to run successfully, the following OAM environment must exist:

- No write, relabel, or reformat requests are scheduled for the requested volume.
- No eject request is scheduled (for the requested volume or its opposite side).
- The volume is not write protected and is not marked not writable.
- In an OAMplex environment, the requested volume is controlled by the instance of OAM on which the reformat request was entered.

The following is the syntax for the OAMUTIL REFORMAT command:



The following is a description of each of the keywords for this command:

old_volser

Indicates the volume serial number of the volume to be reformatted. The FORCE keyword is required if *old_volser* belongs to an Object Backup storage group.

ONE | BOTH

An optional parameter. ONE indicates that only one side of optical cartridge should be reformatted. This is the default.

BOTH indicates that both sides of the optical disk cartridge should be reformatted.

NEWVOL1(*new_volser1*)

An optional parameter that indicates the new volume serial number for side one of the optical disk cartridge. If not specified, there is no change to the volume serial number of side one of the optical disk cartridge.

NEWVOL2(*new_volser2*)

An optional parameter that indicates the new volume serial number for side two of the optical disk cartridge. If not specified, there is no change to the volume serial number of side two of the optical disk cartridge.

DRIVENAME(*drive_name*)

An optional parameter that indicates a specified operator-accessible drive that is capable of processing this request. This parameter is valid only if the requested optical disk volume is shelf-resident. If this parameter is not specified, or if the requested optical disk volume is library-resident, OAM selects an optical drive in the SMS ACDS capable of processing this request.

NOSCRATCH | SCRATCH

An optional parameter that indicates which storage group the volume should be placed in after reformatting. NOSCRATCH indicates that the volume should remain in the same storage group to which it was assigned before the successful completion of the reformat execution. *This is the default.*

SCRATCH indicates that both volumes on an optical disk cartridge should be placed in the SCRATCH storage group on successful execution of the reformat request.

NOFORCE | FORCE

An optional parameter that indicates what type of reformat request is being run. NOFORCE indicates that the reformat request is conditional. The utility must verify the existence of any backup copies or active primary copies of objects on the volume before performing the reformat. If these objects exist on the volume, the reformat fails. *This is the default.*

FORCE indicates that the reformat request is unconditional. This parameter allows you to physically erase all copies of objects on a rewritable optical disk cartridge or discard all copies of objects on a WORM optical disk cartridge without first verifying if there are any backup copies or active primary copies of objects on the volume. The volume is reinitialized even if there are non expired objects on the volume. The FORCE option is required when reformatting a volume belonging to an Object Backup storage group.

To reformat a requested optical disk volume, enter the following TSO/E command:

```
OAMUTIL REFORMAT old_volser BOTH
```

This system issues the following messages:

```
CBR4401I Volume old_volser1 mounted on drive drive-name.

CBR4465I Volumes old_volser1 and old_volser2 are being reformatted on drive
         drive-name.

CBR4401I Volume old_volser2 mounted on drive drive-nam.

CBR4462I Volume old_volser1 has been reformatted to new_volser1.

CBR4462I Volume old_volser2 has been reformatted to new_volser2.
```

The following conditions exist upon successful completion of this command:

- All object directory entries for the requested volumes are deleted from the 3995 controller, the Object Directory Table, and the Deleted Objects Table.
- All space used by the erased objects is reclaimed for rewritable optical disk volumes.
- If SCRATCH is specified, both volumes on the optical disk cartridge are returned to the SCRATCH storage group.
- If NEWVOL1 is specified, a new row for this new volume serial number for the first side of the optical disk cartridge is added to the Volume Table. The row for the old volume serial number of the requested volume is deleted from the Volume Table. If NEWVOL1 is not specified, the row for the requested volume in the OAM Volume Table is updated.
- If NEWVOL2 is specified and it is a both side reformat, a new row for the new volume serial number for the second side of the optical disk cartridge is added to the Volume Table. The row in the Volume Table for the old volume serial number is deleted. If NEWVOL2 is not specified or the request is for a one side reformat, the row in the Volume Table for the opposite side of the requested volume is updated.

For information concerning messages generated from this command, see *z/OS MVS System Messages, Vol 4 (CBD-DMO)*.

Displaying status

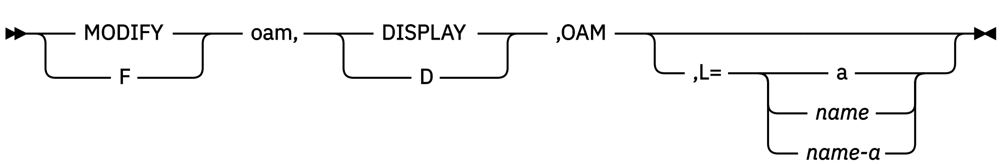
You can display the status of various items of the OAM system:

- OAM status
- OAM XCF status
- OSMC summary status
- OSMC task status
- Drive online/offline connectivity
- Drive detail status
- Library online/offline connectivity
- Library detail status
- Storage group and volume status
- SETDISK parameters
- SETOAM parameters
- SETOPT parameters
- SETOSMC parameters

Requirement: To display information in the Tape Volume table concerning objects stored on tape volumes, you must use the SPUFI SELECT command. Using the DISPLAY command against these tape volumes only provides information from the tape configuration database, not the Tape Volume table.

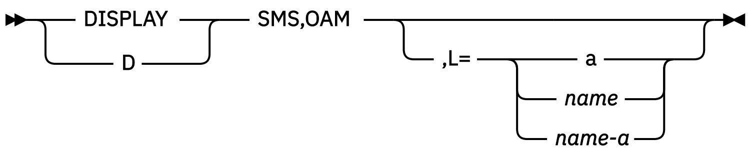
Displaying OAM Status

The following command displays OAM status:



Where *oam* is the taskname of the OAM address space for the desired OAM instance and L={a | *name* | *name-a*} specifies the location where the results of the inquiry are to be displayed where *name* is the console name and *a* is the display area on the console screen.

In a classic OAM configuration, or for the tape library address space in a multiple OAM configuration, the following command can also be used and will be transformed into the above version of the command:



OAM

Displays OAM status.

L={a | *name* | *name-a*}

Specifies the location where the results of the inquiry are to be displayed, where *name* is the console name, and *a* is the display area on the console screen.

To display OAM status, enter the following command:

```
DISPLAY SMS,OAM
```

This results in F OAM,D,OAM command being issued.

Many commands use the MVS operator MODIFY command. Throughout this topic, the command syntax is:

```
F OAM,...
```

The command F is the abbreviation of the MVS MODIFY command.

The “OAM” portion identifies an OAM address space where this identification for the OAM address space can be defined in one of the following ways:

- As the name of the cataloged procedure in SYS1.PROCLIB that you use to start the OAM address space:

```
START OAM,...
START OAMA,...
START OAMB,...
```

- As the task ID assigned in the address space START command:

  START *procname.OAM*,...START *OAM.OAMA*,...START *OAM.OAMB*,...

In a multiple OAM configuration, it is recommended that a naming convention be established to distinguish one OAM Object address space from another using a prefix of “OAM” and a lettered suffix (OAMA, OAMB). This helps to differentiate OAM subsystems (recommended to be named using a prefix of “OAM” and a numbered suffix) from OAM address spaces. This naming convention also can ease operations:

- Commands can be directed to a specific OAM address space:

```
F OAMA,...
F OAMB,...
```

- Commands can also be directed to all OAM address spaces using wildcarding:

```
F OAM*,...
```

In some cases the DISPLAY SMS command can be used to obtain status about various aspects of OAM as shown in the table below.

Note that in a multiple OAM configuration:

- a DISPLAY SMS command will be sent only to the Tape Library OAM address space (if it exists) and only for those types of displays that are applicable to a Tape Library OAM address space
- a MODIFY command (F *oam*) must be used to direct a command to an Object OAM address space

Command	Classic OAM Configuration	Multiple OAM Configuration with Tape Library Address Space	Multiple OAM Configuration with no Tape Library Address Space	Alternate Syntax *Required* for Object Address Space in multiple OAM configuration *Optional* in classic OAM configuration or for Tape Library Address Space in multiple OAM configuration
D SMS,OAM	valid	valid	error[4]	F *oam*,D,OAM
D SMS,LIBRARY	valid	valid	error[4]	F *oam*,D,LIB[1]
D SMS,STORGRP	valid	valid	error[4]	F *oam*,D,GROUP
D SMS,VOLUME	valid	valid	error[4]	F *oam*,D,VOL
D SMS,OAMXCF	valid	error[2]	error[4]	F *oam*,D,OAMXCF
D SMS,OSMC	valid	error[2]	error[4]	F *oam*,D,OSMC
D SMS,DRIVE	valid	error[3]	error[3,4]	F *oam*,D,DRIVE[1]

The output of some of the displays will be different in a multiple OAM configuration depending upon whether the OAM address space processing the display command is an Object OAM address space or a Tape Library OAM address space.

- When displaying the status for "OAM", an Object OAM address space will only display values applicable to object processing and a Tape Library OAM address space will only display values applicable to tape library processing. For more information see CBR1100I in *z/OS MVS System Messages, Vol 4 (CBD-DMO)*.
- When displaying the status for "STORGRP", an Object OAM address space will only display values applicable to object processing and a Tape Library OAM address space will only display values applicable to tape library processing. For more information see CBR1130I in *z/OS MVS System Messages, Vol 4 (CBD-DMO)*.
- When displaying the status for "VOLUME", an Object OAM address space will only display values applicable to object processing (message CBR1240I in *z/OS MVS System Messages, Vol 4 (CBD-DMO)* for object related tape volumes; message CBR1140I in *z/OS MVS System Messages, Vol 4 (CBD-DMO)* for optical volumes will not be displayed because the Optical level of the OAM storage hierarchy is not supported in a multiple OAM configuration) and a Tape Library OAM address space will only display values applicable to tape library processing (message CBR1180I in *z/OS MVS System Messages, Vol 4 (CBD-DMO)*).

[1] Note that F *oam*,D,DRIVE and F *oam*,D,LIB are not valid in a multiple OAM configuration in an Object OAM address space since the Optical level of the OAM storage hierarchy is not supported in a multiple OAM configuration.

[2] This command is only applicable to an Object OAM address space in a multiple OAM configuration and therefore requires the F *oam* form of the command rather than the D SMS form.

[3] The Optical level of the OAM storage hierarchy is not supported in a multiple OAM configuration.

[4] In a multiple OAM configuration, D SMS commands applicable to OAM are processed by the Tape Library OAM address space and thus not valid if no Tape Library Address Space is active.

The OAM display status shows which backup copy is being used by automatic access to backup processing for each of the possible reasons. If automatic access to backup is not active for a specific reason, the status display shows that no backup copy is being used. The following information is displayed for an optical library.

```
CBR1100I OAM status:
TOT   USE   TOT  USE  AVL  TOT  USE   AVL  TOT  USE  AVL  SCR   REQ
LIB   LIB   DRV  DRV  DRV  LDR  LDR   LDR  SDR  SDR  SDR  VOL    CT
aaa   bbb   ccc  ddd  eee  fff  ggg   hhh  iii  jjj  kkk  lll   mmm
Category count scratch transition {ENABLED|OPERATOR DISABLED}.
exitname processing {ENABLED|DISABLED|BYPASSED|BYPASSED_RS|OPERATOR-DISABLED}.
Access Backup status for xxx reasons, using yyy backup copy.
Diagnostic messages {ACTIVE|INACTIVE} for OSREQFS. Limit=nnnn.
DB2 SSID: ssid
XCF GROUP NAME: group-name
XCF MEMBER NAME: member-name
CBROAM: parmlib-suffix
OAMn Parms: TIME=xxx MSG=xx OTIS=x
            UPD=x QB=x MOS=xxxx LOB=x DP=x
```

The fields in the data line specify the number of each resource, as follows:

aaa
: Total number of optical libraries in the configuration.

bbb
: Number of usable optical libraries (online and operational).

ccc
: Total number of optical drives in the configuration.

ddd
: Number of usable optical drives.

eee
: Number of available optical drives (online, operational, and not in use).

fff
: Total number of library optical drives in the configuration.

ggg
: Number of usable library optical drives.

hhh
: Number of available library optical drives.

iii
: Total number of operator-accessible optical drives in the configuration.

jjj
: Number of usable stand-alone or operator-accessible optical drives.

kkk
: Number of available operator-accessible optical drives.

lll
: Number of scratch optical volumes in the OAM configuration database.

mmm
: Total number of optical read requests waiting to be scheduled.

nnnn
: When the status for diagnostic messages issued for file system related errors originating from OSREQ requests is 'ACTIVE', indicates the approximate number of messages yet to be issued.

For the OAM category count scratch transition setting, the following is displayed in the status message in a classic OAM configuration or for the Tape Library address space in a multiple OAM configuration:

`Category count scratch transition` {ENABLED|OPERATOR DISABLED}

ENABLED
: Category count I/O calls to the library for transitions from private to scratch are enabled.

OPERATOR DISABLED
Category count I/O calls to the library for transitions from private to scratch have been disabled by the operator. Other transitions from scratch to private (during job processing) continue to issue the category count call to the library.

A monitoring task continues to update the scratch count every 10 minutes.

exitname
The name of the exit for which status is being displayed. This line is repeated for each installation exit. You can use the CBRUXSAE and EDGTVEXT exits for object support. However, you would use the other exits (CBRUXENT, CBRUXEJC, CBRUXCUA, and CBRUXVNL) with tape libraries. The status codes are as follows:

ENABLED
The exit is enabled and executed when the requested function is required.

DISABLED
The exit has been disabled due to an error or an abend in the installation exit. For EDGTVEXT, OAM continues releasing object tape volumes from the OAM inventory.

BYPASSED
For CBRUXVNL, either the exit returned code 16, indicating that it was not to be called again, or an error (or abend) occurred in the exit and the exit will not be invoked. For CBRUXSAE, either the exit returned a return code 16, or it returned return codes for each of the five OSREQ functions (STORE, RETRIEVE, QUERY, CHANGE and DELETE), effectively putting all OSREQ functions in bypass mode. For all other exits, the exit returned a return code 16, indicating that the requested function is to continue without calling the exit.

BYPASSED_RS
Bypass in restricted-store mode: This is used exclusively for the CBRUXSAE PROCESSING for STORE. The exit returned a return code 254 for an OSREQ STORE (or STOREBEG) request indicating that subsequent stores are allowed to existing collections but are not allowed to collections that do not exist.

OPERATOR-DISABLED
Only the exits for the tape library uses this status code. For information on displaying OAM status for tape libraries, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

If the CBRUXSAE user exit is in ENABLED mode then the following lines are displayed:

```
CBRUXSAE processing {ENABLED|BYPASSED|BYPASSED_RS} for STORE.
CBRUXSAE processing {ENABLED|BYPASSED} for RETRIEVE.
CBRUXSAE processing {ENABLED|BYPASSED} for QUERY.
CBRUXSAE processing {ENABLED|BYPASSED} for CHANGE.
CBRUXSAE processing {ENABLED|BYPASSED} for DELETE.
```

The following fields are displayed in the status message for the OAM access backup processing:

access backup status
The status of Access Backup processing for this reason. The following are valid *status* values:

- `ACTIVE`—Access Backup processing is active for this reason.
- `INACTIVE`—Access Backup processing is inactive for this reason.

xxx
The reason for which Access Backup processing can be activated. The following are valid *xxx* values:

- `DB2 OBJECT TABLE ERRORS`—The backup copy of an object is retrieved when a DB2 object table error occurs while attempting to retrieve the primary copy of the object.
- `FILE SYSTEM ERRORS`—The backup copy of an object is retrieved when a file system error occurs while attempting to retrieve the primary copy of the object.
- `LOST VOLUMES`—The backup copy of an object is retrieved when the optical or tape volume on which the object resides is marked lost or is not-defined.

- NOT OPERATIONAL LIBRARIES—The backup copy of an object is retrieved when the primary copy resides on a volume that is in a library that is not operational.
- OFFLINE LIBRARIES—The backup copy of an object is retrieved when the primary copy resides on a volume that is in an offline library.
- UNREADABLE VOLUMES—The backup copy of an object is retrieved when the primary copy resides on a volume with READABLE=N.

yyy
The indicator of which backup copy, if any, is being used for automatic access to backup processing. The following are valid values for *yyy*:

- 1st—Access Backup processing accesses the first backup copy of the object when the primary copy is unavailable for the reason shown in *xxx*.
- 2nd—Access Backup processing accesses the second backup copy of the object when the primary copy is unavailable for the reason shown in *xxx*.
- NO—Access Backup processing is inactive for the reason shown in *status*; therefore, no backup copy is being used.

ssid
Specifies the name of the DB2 subsystem that OAM uses for object support. The subsystem name is from 1 to 4 characters.

group-name
The XCF group name for this OAMplex.

member-name
The XCF member name for this instance of OAM in an OAMplex.

cbroam-parmlib-suffix
This field displays the suffix of the CBROAMxx PARMLIB member that was in effect during OAM initialization.

OAMn Parms
Displays settings that resulted from the parameters specified for the OAMn subsystem definition (in the IEFSSNxx PARMLIB member or on the SETSSI ADD command). Refer to Changing system libraries for detailed information on the meaning of these settings.

Note: Refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* for more information on these parameters.

TIME=*xxx*
Specifies whether or not the timestamp value in the object directory in DB2 is to be based on GMT or local times.
Specifies upon which time zone the timestamp value in the object directory in DB2 is to be based and also affects the time zone for CYCLE START and CYCLE END times defined within the storage group definition for each object and object backup storage groups:

- TIME=GMT Indicates that the timestamp value is based on GMT.
- TIME=LOC Indicates that the timestamp value is based on local time. This is the default value.

MSG=*xx*
Specifies the format for how the OAM message text appears:

- MSG=EM Indicates that the message text is in mixed-case English. This is the default.
- MSG=EU Indicates that the message text is in uppercase English.

UPD=*x*
Specifies whether DB2 updates for the pending action date (ODPENDDT) and the last reference date (ODLREFDT) fields should be performed:

- UPD=Y Indicates that the ODPENDDT and ODLREFDT fields should be updated on all OSREQ RETRIEVE requests.

  Note: The ODLREFDT field is not updated for OSREQ CHANGE requests. This is the default.

- UPD=N Indicates that the ODPENDDT and ODLREFDT fields should not be updated for any OSEQ RETRIEVE requests.

  Note: The ODLREFDT field is not updated for OSREQ CHANGE requests.

  Restriction: If you use UPD=N, you cannot base transition criteria on the time since last use parameter in the ISMF management class definition.

- UPD=C Indicates that the ODPENDDT and ODLREFDT fields should be updated on all OSREQ RETRIEVE and on all OSREQ CHANGE requests.

  Note: Regardless of the setting for UPD=, the ODLREFDT is not updated when:

  – OSREQ RETRIEVE results in RECALL being scheduled, or 2.

  – OSREQ RETRIEVE of object currently in RECALL mode.

QB=*x*

Specifies whether or not an OSREQ QUERY request results in a call into the OAM address space to retrieve the backup retrieval order keys. This specification is at the global level and pertains to all OSREQ QUERY processing.

- QB=Y Indicates that OSREQ QUERY requests result in a call into the OAM address space for each backup copy. The OSREQ QUERY will return a complete backup retrieval order key for each backup copy. If a backup copy does not exist, then the OAM address space will not be called and the backup retrieval order key will contain binary zeros. This is the default.
- QB=N Indicates that OSREQ QUERY requests will not result in a call into the OAM address space for each backup copy. The backup retrieval order key will contain binary zeros for each backup copy regardless if the backup copy exists or not.

MOS=*xxxx*

Specifies the maximum object size limit in MB. Valid values are 50–2000. The maximum object size is checked when objects are initially stored through the OSREQ programming interface and is not checked on subsequent retrievals.

OTIS=x

Specifies whether OTIS should wait for JES to completely initialize before OTIS is started:

- OTIS=Y Indicates that OTIS will not start until JES is completely initialized.
- OTIS=N Indicates that OTIS will start independently from JES. This is the default.

LOB=*x*

Specifies whether or not OAM exploits DB2 LOB support for large objects that exceed 32 KB (32640 bytes). LOB has the following options:

- LOB=A Indicates that, for all storage groups, objects that exceed 32 KB are to be stored in a LOB storage structure when stored to DB2. LOB=A indicates to OAM that the installation has created LOB storage structures and associated V_OSM_LOB_BASE_TBL views for ALL object storage groups defined in the ACDS. This results in optimal performance when you want to store large objects (greater than 32 KB) to DB2, because OAM does not query DB2 to see if the LOB base table view exists. If the LOB base table view does not exist, the large object store fails.
- LOB=P Indicates to OAM that the installation has created LOB storage structures and associated V_OSM_LOB_BASE_TBL views for a PARTIAL list of object storage groups defined in the ACDS. However, for object stores to DB2 greater than 256M, the LOB=P designation is treated the same as LOB=A; an attempt to store the object to the LOB base table is made without first checking if the V_OSM_LOB_BASE_TBL view exists.

  If the object being stored is less than or equal to 256M and greater than or equal to 32KB, then LOB=P requires OAM to query DB2 to see if the LOB base table view exists for a given object storage group. If the LOB base table view does exist for a given object storage group, large objects are stored in the associated LOB storage structure. If the LOB base table view does not exist, large objects are stored in the 32KB data table.

- LOB=N Indicates that objects that exceed 32 KB and less than or equal to 256 MB are to be stored in a 32 KB data table when stored to DB2. Stores will fail for objects that exceed 256 MB. This is the default.

DP=*x*
Specifies that scope at which deletion-protection is enabled or disabled. If a given object storage group has deletion-protection enabled, then no objects can be deleted from that object storage group prior to the object's expiration date.

- DP=A Indicates that deletion-protection is enabled for all object storage groups.
- DP=P Indicates that deletion-protection is partially enabled. Specifically, deletion-protection is enabled only for object storage groups that have the OAM Deletion Protection setting defined as ENABLED in ISMF for the object storage group SMS construct.
- DP=N Indicates that deletion-protection is not enabled for any object storage group. This is the default.

The following is a sample of DISPLAY SMS,OAM status:

```
CBR1100I OAM status: 418
OPT. TOT  USE  TOT  USE  AVL  TOT  USE  AVL  TOT  USE  AVL  SCR  REQ
     LIB  LIB  DRV  DRV  DRV  LDR  LDR  LDR  SDR  SDR  SDR  VOL  CT
       4    0   10    0    0    8    0    0    2    0    0    4    0
TAPE TOT  ONL  TOT  TOT  TOT  TOT   TOT   ONL   AVL   TOTAL
     LIB  LIB  AL   VL   VCL  ML    DRV   DRV   DRV  SCRTCH
       6    0    0    0    0    1     0     0     0     107
There are also 0 VTS distributed libraries defined.
Category count scratch transition ENABLED.
CBRUXCUA processing ENABLED.
CBRUXEJC processing ENABLED.
CBRUXENT processing ENABLED.
CBRUXVNL processing ENABLED.
CBRUXSAE processing ENABLED.
CBRUXSAE processing ENABLED for STORE.
CBRUXSAE processing ENABLED for RETRIEVE.
CBRUXSAE processing ENABLED for QUERY.
CBRUXSAE processing ENABLED for CHANGE.
CBRUXSAE processing ENABLED for DELETE.
EDGTVEXT processing ENABLED.
Access Backup processing ACTIVE for UNREADABLE VOLUMES, using 1st
backup copy.
Access Backup processing ACTIVE for OFFLINE LIBRARIES, using 1st
backup copy.
Access Backup processing ACTIVE for NOT OPERATIONAL LIBRARIES, using
1st backup copy.
Access Backup processing ACTIVE for DB2 OBJECT TABLE ERRORS, using
1st backup copy.
Access Backup processing ACTIVE for LOST VOLUMES, using 1st backup
copy.
Access Backup processing ACTIVE for FILE SYSTEM ERRORS, using 1st
backup copy.
Diagnostic messages ACTIVE for OSREQFS. Limit=  10.
DB2 SSID:  DB2
XCF GROUP NAME: -N/A-
XCF MEMBER NAME: -N/A-
CBROAM: 19
OAMn Parms: TIME=LOC  MSG=EM  UPD=C  QB=Y
            MOS=2000  OTIS=N  LOB=P  DP=P
```

If both optical and tape libraries are defined in the active SMS configuration, the optical library information is displayed first, followed by the tape library information. For an example of this display for a tape library, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

The CBRUXSAE installation exit performs security authorization checking for users performing OSREQ transactions on object data. For information on the CBRUXSAE exit, see *z/OS DFSMS OAM Application Programmer's Reference*.

Displaying OAM configuration information

Use the DISPLAY OAM,CONFIG command to display the configuration for the OAM component of DFSMS. OAM subsystem and address space configuration will be displayed as well as OAMplex membership.

DISPLAY OAM,CONFIG
DISPLAY OAM,CONFIG [,L={*a*|*name*|*name-a*|}]

L=*a, name, or name-a*
: Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

The following is example output for a classic OAM configuration.

```
CBR1960I OAM configuration data:
OAM  OAM     OAM     OAM      OAM     OAMPLEX  DB2  DB2
SUB  PROC    TASKID  STC#     TYPE    GROUP    ID   GATT
OAM1 OAM     OAM     STC00317 CLAS             DB2A
```

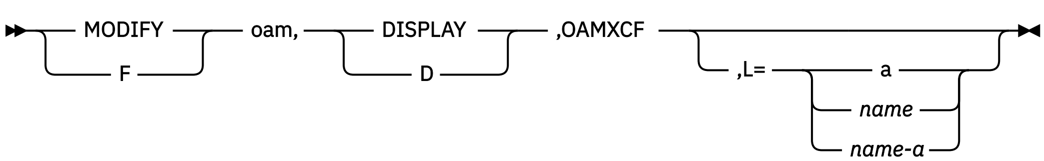
The following is example output for a multiple OAM configuration.

```
CBR1960I OAM configuration data:
OAM  OAM     OAM     OAM      OAM     OAMPLEX  DB2  DB2  DB2
SUB  PROC    TASKID  STC#     TYPE    GROUP    ID   SSID GATT
OAM0 OAMT    OAMT    STC07053 TLIB             NONE
OAM1 OAMA    OAMA    STC07051 OBJ     PRODPLX2 DB2A DB2A DBG1
OAM2 OAMB    OAMB    STC07052 OBJ     PRODPLX2 DBG2 DB2B DBG2
```

Note: For a detailed description of the output of the DISPLAY OAM,CONFIG command, see the description of message CBR1960I in *z/OS MVS System Messages, Vol 4 (CBD-DMO)*.

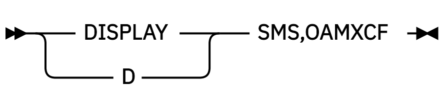
Displaying OAM XCF status

This command displays system status for this instance of OAM in relation to the sysplex and XCF. The following command displays OAM XCF status:



Where *oam* is the taskname of the OAM address space for the desired OAM instance and L={a | *name* | *name-a*} specifies the location where the results of the inquiry are to be displayed where *name* is the console name and *a* is the display area on the console screen.

In a classic OAM configuration, or for the tape library address space in a multiple OAM configuration, the following command can also be used and will be transformed into the above version of the command:



The following is a description of the keyword for this command:

OAMXCF
: Displays OAM XCF status.

To display OAM XCF status, enter the following command:

```
DISPLAY SMS,OAMXCF
```

This results in F OAM,D,OAMXCF command being issued.

The following information is displayed:

```
CBR1250I OAM XCF STATUS:
XCF MEMBER NAME    USER              SYSTEM    OPT     OPT    TAPE
                   STATE             NAME      READ    WRITE  READ
xcf-member-name    aaaaaaaaaaaaaaaa  bbbbbbbb  cccc    ddddd  eeeee
-----------------------------------------------------------------------------

this-xcf-member   ffffffffffffffff   gggggggg  hhhh    iiiii  jjjjj
XCF GROUP NAME:  xcf-group-name
OAM XCF timeout value for XCFOPTREADA is seconds.
OAM XCF timeout value for XCFOPTREADM is seconds.
OAM XCF timeout value for XCFOPTWRITEA is seconds.
OAM XCF timeout value for XCFOPTWRITEM is seconds.
OAM XCF timeout value for XCFTAPEREADA is seconds.
OAM XCF timeout value for XCFTAPEREADM is seconds.
```

For instances of OAM other than the OAM where the display command was issued, the fields displayed in each data line are as follows:

xcf-member-name
: The member name associated with an instance of OAM in the OAMplex

aaaaaaaaaaaaaaaa
: The user state of *xcf-member-name* on this data line. OAM defined user states are INITIALIZING, TERMINATING, RESTARTING, or ACTIVE.

bbbbbbbb
: The system name associated with *xcf-member-name* on this data line.

cccc
: The number of optical reads sent from the OAM where the command was entered to the OAM on the display line to be processed.

ddddd
: The number of optical writes sent from the OAM where the command was entered to the OAM on the display line to be processed.

eeeee
: The number of tape reads sent from the OAM where the command was entered to the OAM on the display line to be processed.

For instances of OAM on the system where the display command was issued, the following fields are displayed in the last data line of the multiline message:

this-xcf-member
: The member name associated with this instance of OAM in the OAMplex where the display command was issued.

ffffffffffffffff
: User state of *this-xcf-member* where the command was issued. OAM defined user states are INITIALIZING, TERMINATING, RESTARTING, or ACTIVE.

gggggggg
: System name associated with *xcf-member-name* on this data line.

hhhhh
: The total number of optical reads sent from the OAM where the command was entered to other OAMs in the OAMplex.

iiiii
: The total number of optical writes sent from the OAM where the command was entered to other OAMs in the OAMplex.

jjjjj
: The total number of tape reads sent from the OAM where the command was entered to other OAMs in the OAMplex.

xcf-group-name
: The XCF group associated with the OAMplex.

seconds

The OAM XCF timeout values for each XCFTIMEOUT subparameter (specified in the CBROAM*xx* member of PARMLIB when OAM was initialized or set by operator command) in effect for the OAM where the command was entered.

The following is a sample of DISPLAY SMS,OAMXCF status:

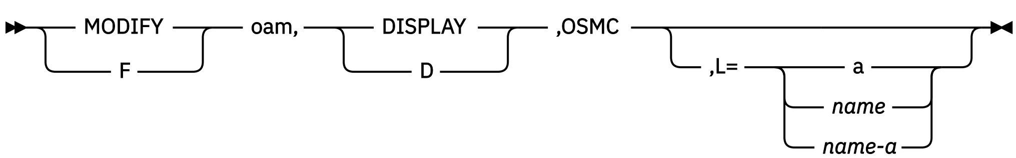
```
CBR1250I OAM XCF status:
XCF MEMBER NAME   USER          SYSTEM    OPT    OPT    TAPE
                  STATE          NAME     READ   WRITE  READ
OAMSYS2           ACTIVE        SYSTEM2     27      65     0
OAMSYS3           ACTIVE        SYSTEM3     36       0    22
------------------------------------------------------------
OAMSYS1           ACTIVE        SYSTEM1     63      65    22
XCF GROUP NAME:  OAMGRP1
OAM XCF timeout value for XCFOPTREADA is 20.
OAM XCF timeout value for XCFOPTREADM is 50.
OAM XCF timeout value for XCFOPTWRITEA is 150.
OAM XCF timeout value for XCFOPTWRITEM is 150.
OAM XCF timeout value for XCFTAPEREADA is 40.
OAM XCF timeout value for XCFTAPEREADM is 50.
```

If the instance of OAM is not part of an OAMplex, the following message will display:

```
CBR1069I Command rejected. OAM is not a member of an XCF group in a
         sysplex environment.
```

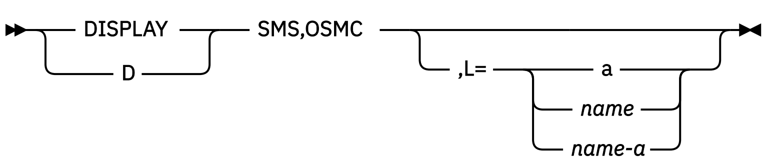
Displaying OSMC summary status

The following command displays OSMC status:



Where *oam* is the taskname of the OAM address space for the desired OAM instance and L={a | *name* | *name-a*} specifies the location where the results of the inquiry are to be displayed where *name* is the console name and *a* is the display area on the console screen.

In a classic OAM configuration, or for the tape library address space in a multiple OAM configuration, the following command can also be used and will be transformed into the above version of the command:



OSMC

Displays OSMC status.

L={a | *name* | *name-a*}

Specifies the location where the results of the inquiry are to be displayed, where *name* is the console name, and *a* is the display area on the console screen.

To display OSMC summary status, enter the following command:

```
DISPLAY SMS,OSMC
```

This results in F OAM,D,OSMC command being issued.

The following information is displayed:

```
D SMS, OSMC
CBR9350I OSMC Summary Status:
TASK      TASK      TASK       START       OBJECTS       OBJECTS
NAME      TYPE      STAT       TIME        COMPLETED     ACTIVE
CBR9364I OSMC Summary Status 2:
          TASK      TASKS      TASKS
ACTIVITY  TYPE      ACTIVE     QUEUED
IMBKUP    I              0          0
RCLDISK    B              0          0
End of Display Summary
CBR9355I No OSMC processes are active at this time.
```

```
CBR9350I OSMC Summary Status:

Explanation:

TASK     TASK     TASK    START      OBJECTS      OBJECTS
NAME     TYPE     STAT    TIME       COMPLETED    ACTIVE
tskname  tsktype  tskstat starttime  objcomplete  objactive
```

Message CBR9364I will display the number of Immediate Backup tasks and the number of Recall to Disk tasks that are currently processing and queued to process.

```
CBR9364I OSMC Summary Status 2:

Explanation:

          TASK   TASKS    TASKS
ACTIVITY  TYPE   ACTIVE   QUEUED
IMBKUP    I      active   queued
RCLDISK    B      active   queued
```

The fields displayed in each data line are as follows:

tskname
: Name of control task

tsktype
: Type of control task:

  B
  : Recall to Disk

  C
  : Storage Group running as part of an OSMC Cycle as part of MODIFY OAM,START,OSMC only

  D
  : DASD space management

  G
  : Single storage group processing from an Operator Start command or Automatic Start based on Start time in ISMF

  I
  : Immediate Backup

  L
  : Library space management

  M
  : Move Volume

  R
  : Volume recovery

  Y
  : Operator initiated Recycle command process

tskstat
: Task current status:

ƀ
(Blank) OSMC is running

A
Task is ACTIVE

P
Task is stopping

T
Task is terminating

starttime
Task start time (hh.mm.ss)

objcomplete
Number of objects that have completed processing

objactive
Number of objects being processed

The system displays the following information before it issues the CBR9350I message if a RECYCLE task type is currently active.

```
CBR9356I Recycle Summary Status:

 Explanation:

TASK    TASK TASK  START     START           VOLS      VOLS
NAME    TYPE STAT  DATE      TIME      LIMIT COMPLETE  ACTIVE
RECYCLE Y    tstat startdate starttime limit volcomp   volact
```

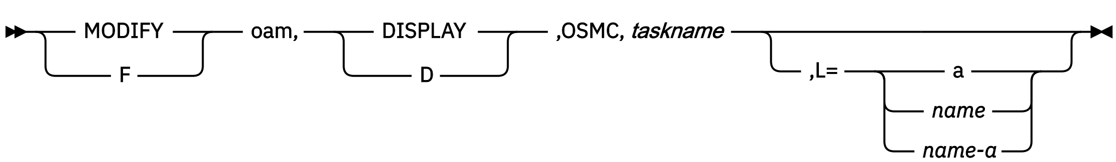
Note:

The system issues this message in response to a D SMS,OSMC operator command if there is an active RECYCLE command processing. The system provides the summary status information for the recycle process associated with the MODIFY OAM,START,RECYCLE command.

The summary information includes the name of the task, type of task, a task status (ending or stopped), or blank, the date and time the task was started, the limit of volumes to be recycled as indicated on the MODIFY OAM,START,RECYCLE operator command, the number of volumes completed processing, and the number of volumes still being actively processed.

Displaying OSMC task status

The following command displays OSMC task status:



Where *oam* is the taskname of the OAM address space for the desired OAM instance, *taskname* specifies the name of the task and L={a | *name* | *name-a*} specifies the location where the results of the inquiry are to be displayed where *name* is the console name and *a* is the display area on the console screen.

To display the status of an OSMC task, enter the following command:

```
DISPLAY SMS,OSMC,TASK(taskname)
```

This results in F OAM,D,OSMC,taskname command being issued.

OSMC
Displays OSMC task status

TASK(*taskname*)
Specifies the name of the task for which a status display is requested

The value of the task name depends on the type of OSMC process which the task represents. The following list shows the type of OSMC process, and the value used for its name.

Library space management
The library name

Volume recovery
The volume serial of one of the volumes on the disk being recovered

DASD space management
The name of the storage group being processed by DASD space management

OSMC processing of one storage group
The name of the storage group being processed

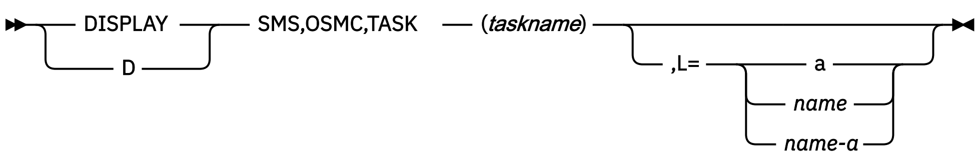
Storage management cycle processing of a storage group
The name of the storage group being processed

Move volume
The volume serial of the source volume from which objects are being moved

Recycle
The name of the volser being processed as a result of the RECYCLE command. RECYCLE is an accepted value.

L={a | *name* | *name-a*}
Specifies the location where the results of the inquiry are to be displayed, where *name* is the console name, and *a* is the display area on the console screen.

In a classic OAM configuration, or for the tape library address space in a multiple OAM configuration, the following command can also be used and will be transformed into the above version of the command:



[NOT Programming Interface Information] Detail status information is provided for the OAM storage management component task specified in the DISPLAY command. The number of internal work items queued on the work and wait queues and the number of internal work items completed for each of the OSMC services is displayed. The number of internal work items does not exactly equate to the number of objects processed; there might be multiple internal work items per object or there might be internal work items not associated with any object. This information is better used for problem determination and monitoring the progress of the storage management component than for tracking the number of objects processed. [End NOT Programming Interface Information]

The fields displayed in each data line represent the services that OSMC performs during its processing. The following information is displayed:

```
CBR9370I OSMC Detail for taskname:
           READ      READ     READ     READ
           DISK1    DISK2      OPT     TAPE
WORK Q: aaaaaaaa bbbbbbbb cccccccc dddddddd
WAIT Q:          eeeeeeee ffffffff gggggggg
DONE:   hhhhhhhh iiiiiiii jjjjjjjj kkkkkkkk
           WRITE    WRITE    WRITE    WRITE    WRITE
           DISK1    DISK2      OPT    TAPE1    TAPE2
WORK Q: llllllll mmmmmmmm nnnnnnnn oooooooo pppppppp
WAIT Q:          qqqqqqqq rrrrrrrr ssssssss tttttttt
DONE:   uuuuuuuu vvvvvvvv wwwwwwww xxxxxxxx yyyyyyyy
           WRITE    WRITE      DIR
         BACKUP1  BACKUP2    UPDTS
WORK Q: zzzzzzzz 11111111 22222222
WAIT Q: 33333333 44444444 55555555
DONE:   66666666 77777777 88888888
End of Display Detail
```

In the message text, *taskname* is the name associated with the OAM storage management component task and is the same as the task name specified on the DISPLAY SMS,OSMC command. In the case of the OAM storage management cycle, *taskname* is the name of an Object storage group being processed by OSMC. In the case of the OAM MOVEVOL utility, *taskname* is the volume serial number of the volume being operated on by the utility. For the OAM Volume Recovery utility, *taskname* is the volume serial number of the optical or tape cartridge being recovered by the utility.

The fields that are displayed in each row represent the number of internal work items (*n*) that are at that stage of processing for each service:

WORK Q
: Work items queued for processing by this service

WAIT Q
: Work items for which processing has started but not completed

DONE
: Work items that have completed using this service

The values in the display for CBR9370I are defined as follows:

aaaaaa
: The number of internal work items that are queued on the work queue by the read disk sublevel 1 service.

bbbbbb
: The number of internal work items that are queued on the work queue by the read file system service

cccccc
: The number of internal work items that are queued on the work queue by the read optical service.

dddddd
: The number of internal work items that are queued on the work queue by the read tape service. This read service reads from both tape sublevel 1 and tape sublevel 2.

eeeeee
: The number of internal work items that are queued on the wait queue by the read file system service

ffffff
: The number of internal work items that are queued on the wait queue by the read optical service.

gggggg
: The number of internal work items that are queued on the wait queue by the read tape service. This read service reads from both tape sublevel 1 and tape sublevel 2.

hhhhhh
: The number of internal work items that are completed by the read disk sublevel 1 service.

iiiiii
: The number of internal work items that are completed by the read file system service.

jjjjjj
: The number of internal work items that are completed by the read optical service.

kkkkkk
: The number of internal work items that are completed by the read tape service. This read service reads from both tape sublevel 1 and tape sublevel 2.

llllll
: The number of internal work items that are queued on the work queue by the write disk sublevel 1 service.

mmmmmm
: The number of internal work items that are queued on the work queue by the write file system service.

nnnnnn
: The number of internal work items that are queued on the work queue by the write optical service.

oooooo
: The number of internal work items that are queued on the work queue by the write tape sublevel 1 service.

pppppp
: The number of internal work items that are queued on the work queue by the write tape sublevel 2 service.

qqqqqq
: The number of internal work items that are queued on the wait queue by the write file system service.

rrrrrr
: The number of internal work items that are queued on the wait queue by the write optical service.

ssssss
: The number of internal work items that are queued on the wait queue by the write tape sublevel 1 service.

tttttt
: The number of internal work items that are queued on the wait queue by the write tape sublevel 2 service.

uuuuuu
: The number of internal work items that are completed by the write disk sublevel 1 service.

vvvvvv
: The number of internal work items that are completed by the write file system service.

wwwwww
: The number of internal work items that are completed by the write optical service.

xxxxxx
: The number of internal work items that are completed by the write tape sublevel 1 service.

yyyyyy
: The number of internal work items that are completed by the write tape sublevel 2 service.

zzzzzz
: The number of internal work items that are queued on the work queue by the write first backup service.

111111
: The number of internal work items that are queued on the work queue by the write second backup service.

222222
: The number of internal work items that are queued on the work queue by the directory update service.

333333
: The number of internal work items that are queued on the wait queue by the write first backup service.

444444
: The number of internal work items that are queued on the wait queue by the write second backup service.

555555
: The number of internal work items that are queued on the wait queue by the directory update service.

666666
: The number of internal work items that are completed by the write first backup service.

777777
: The number of internal work items that are completed by the write second backup service.

888888
: The number of internal work items that are completed by the directory update service.

Figure 32 on page 322 is a sample of DISPLAY SMS,OSMC,TASK(WG360A):

```
            READ     READ     READ     READ
           DISK1    DISK2      OPT     TAPE
WORK Q:        0        0        1        0
WAIT Q:                 0        3        0
DONE:          0        0        1        1
           WRITE    WRITE    WRITE    WRITE    WRITE
           DISK1    DISK2      OPT    TAPE1    TAPE2
WORK Q:        0        0        0        0        0
WAIT Q:                 0        0        0        0
DONE:          0        0        0        0        0
           WRITE    WRITE      DIR
         BACKUP1  BACKUP2    UPDTS
WORK Q:        0        0        0
WAIT Q:        0        0        0
DONE:          0        0        0
End of Display Detail
```

Figure 32. Sample of DISPLAY SMS,OSMC,TASK(WG360A)

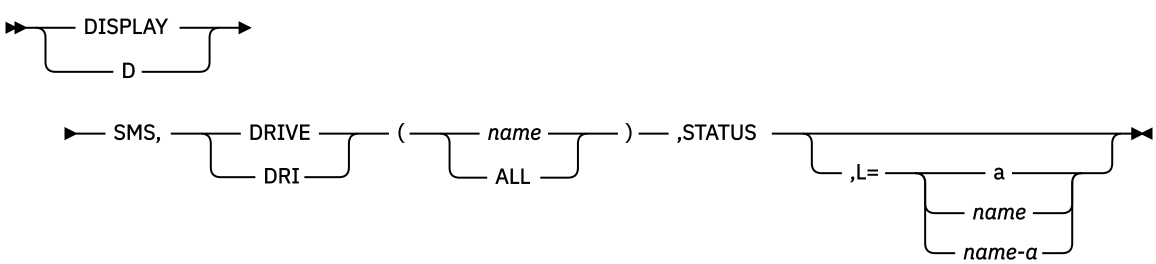
If execution of the D SMS,OSMC,TASK(RECYCLE) command is successful, the system issues the following message to hardcopy log:

```
CBR9875I Recycle Candidates:
The following volumes are candidates for OAM RECYCLE command processing using
pv=nnn, lim=nn, scope=scope, maxrecycletasks=nn, TSL=1|2|-
   VOLSER %VAL SGNAME   STAT    VOLSER %VAL SGNAME   STAT
   Volser nnn  grpname  xxxx    volser nnn  grpname  xxxx
   Volser nnn  grpname  xxxx    volser nnn  grpname  xxxx
OAM Recycle: End of OAM Recycle candidate volumes. Total volumes=nnnn.
```

Related reading: For an example of output for message CBR9370I, see Figure 32 on page 322. For descriptions of the column headings for CBR9370I, see *z/OS MVS System Messages, Vol 4 (CBD-DMO)*.

Displaying drive online/offline connectivity

The following command syntax displays OAM drive status:



DRIVE(*name* | ALL)
: Displays the system connectivity and online/offline status for optical drives. If a drive name is specified, there is one data line describing the specified optical drive. If ALL is specified, all the optical drives in the SMS configuration are displayed. To specify a drive named ALL, surround it with parentheses; for example, DISPLAY SMS,DRIVE((ALL)).

STATUS
: Displays the system connectivity and online/offline status.

L={a | *name* | *name-a*}
: Specifies the location where the results of the inquiry are to be displayed, where *name* is the console name, and *a* is the display area on the console screen.

To display SMS,DRIVE,STATUS for an individual drive, enter the following command:

```
DISPLAY SMS,DRIVE(drvname),STATUS
```

The following information is displayed:

```
IGD002I hh.mm.ss DISPLAY SMS
                                                   1 1 1 1 1 1 1
DRIVE      LIBRARY      SYSTEM=   1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
name       aaaaaaaa               s s s s s s s s s s s s s s s s


                                  1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3
DRIVE      LIBRARY      SYSTEM=   7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
name       aaaaaaaa               s s s s s s s s s s s s s s s s
```

The fields displayed in each data line are as follows:

name
Name of the optical drive for which system connectivity and the online/offline status is displayed

aaaaaaaa
Name of the optical library to which the displayed drive belongs

1–32
Numbers that appear after SYSTEM= indicating the system IDs

s
Indications of drive status:

.
Not defined

+
Online

-
Offline

Recommendation: An online status of "+" does not necessarily mean that either the drive or the library is fully functional. To determine if the drive is both online and operational, you must issue the DISPLAY SMS,DRIVE(*drvname*),DETAIL command. To determine if the library is both online and operational, you must issue the DISPLAY SMS,LIBRARY(*name*),DETAIL command.

For more information on these SMS commands, see "Displaying drive detail status" on page 324 and "Displaying library detail status" on page 332.

The following is a sample of DISPLAY SMS,DRIVE(*drvname*),STATUS:

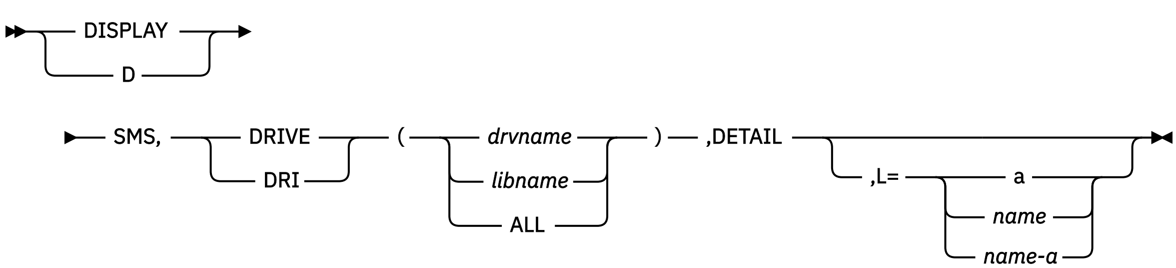
```
D SMS,DRIVE(P13D1),STATUS
IGD002  15:08:16 DISPLAY SMS 397

                                                 1 1 1 1 1 1 1
DRIVE       LIBRARY      SYSTEM=1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
P13D1       PEA13               + - . . . . . . . . . . . . . .

                                1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3
DRIVE       LIBRARY      SYSTEM=7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
P13D1       PEA13               . . . . . . . . . . . . . . . .
**********************************LEGEND*************************
. THE DRIVE IS NOT DEFINED TO THE SYSTEM
+ THE DRIVE IS ONLINE
- THE DRIVE IS OFFLINE
SYSTEM 1  = SYSTEM1    SYSTEM 2  = SYSTEM2    SYSTEM 3  = SYSTEM3
SYSTEM 4  = SYSTEM4    SYSTEM 5  = SYSTEM5    SYSTEM 6  = SYSTEM6
SYSTEM 7  = SYSTEM7    SYSTEM 8  = SYSTEM8    SYSTEM 9  = SYSTEM9
SYSTEM 10 = SYSTEM10   SYSTEM 11 = SYSTEM11   SYSTEM 12 = SYSTEM12
SYSTEM 13 = SYSTEM13   SYSTEM 14 = SYSTEM14   SYSTEM 15 = SYSTEM15
SYSTEM 16 = SYSTEM16   SYSTEM 17 = SYSTEM17   SYSTEM 18 = SYSTEM18
SYSTEM 19 = SYSTEM19   SYSTEM 20 = SYSTEM20   SYSTEM 21 = SYSTEM21
SYSTEM 22 = SYSTEM22   SYSTEM 23 = SYSTEM23   SYSTEM 24 = SYSTEM24
SYSTEM 25 = SYSTEM25   SYSTEM 26 = SYSTEM26   SYSTEM 27 = SYSTEM27
SYSTEM 28 = SYSTEM28   SYSTEM 29 = SYSTEM29   SYSTEM 30 = SYSTEM30
SYSTEM 31 = SYSTEM31   SYSTEM 32 = SYSTEM32
```

The following is a sample of DISPLAY SMS,DRIVE(ALL),STATUS:

```
D SMS,DRIVE(ALL),STATUS
IGD002I 15:08:33 DISPLAY SMS 400
                                           1 1 1 1 1 1 1
DRIVE    LIBRARY SYSTEM= 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
L1D0     LIB1            - + . . . . . . . . . . . . . .
L2D0     LIB2            - + . . . . . . . . . . . . . .
L3D0     LIB3            + - . . . . . . . . . . . . . .
L4D0     LIB4            + - . . . . . . . . . . . . . .
P13D1    PEA13           + - . . . . . . . . . . . . . .
P13D5    PCTREUSE        + - . . . . . . . . . . . . . .
P15D1    PEA15           + - . . . . . . . . . . . . . .
P15D5    PCTREUSE        + - . . . . . . . . . . . . . .
P17D1    PEA17           + - . . . . . . . . . . . . . .
P19D1    PEA19           + - . . . . . . . . . . . . . .
P21D1    PMA21           - + . . . . . . . . . . . . . .
P21D5    P3995133        - + . . . . . . . . . . . . . .
P7D1     PWA7            + - . . . . . . . - . . . . . .
P8D1     PWB8            + - . . . . . . . - . . . . . .
P9D1     PWA9            + - . . . . . . . . . . . . . .
SOU0     STDALONE        . . . . . . . . . . . . . . . .

                         1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3
DRIVE    LIBRARY SYSTEM= 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
L1D0     LIB1            . . . . . . . . . . . . . . . .
L2D0     LIB2            . . . . . . . . . . . . . . . .
L3D0     LIB3            . . . . . . . . . . . . . . . .
L4D0     LIB4            . . . . . . . . . . . . . . . .
P13D1    PEA13           . . . . . . . . . . . . . . . .
P13D5    PCTREUSE        . . . . . . . . . . . . . . . .
P15D1    PEA15           . . . . . . . . . . . . . . . .
P15D5    PCTREUSE        . . . . . . . . . . . . . . . .
P17D1    PEA17           . . . . . . . . . . . . . . . .
P19D1    PEA19           . . . . . . . . . . . . . . . .
P21D1    PMA21           . . . . . . . . . . . . . . . .
P21D5    P3995133        . . . . . . . . . . . . . . . .
P7D1     PWA7            . . . . . . . . . . . . . . . .
P8D1     PWB8            . . . . . . . . . . . . . . . .
P9D1     PWA9            . . . . . . . . . . . . . . . .
SOU0     STDALONE        . . . . . . . . . . . . . . . .
***************************** LEGEND *****************************
. THE DRIVE IS NOT DEFINED TO THE SYSTEM
+ THE DRIVE IS ONLINE
- THE DRIVE IS OFFLINE
SYSTEM 1  = SYSTEM1    SYSTEM 2  = SYSTEM2    SYSTEM 3  = SYSTEM3
SYSTEM 4  = SYSTEM4    SYSTEM 5  = SYSTEM5    SYSTEM 6  = SYSTEM6
SYSTEM 7  = SYSTEM7    SYSTEM 8  = SYSTEM8    SYSTEM 9  = SYSTEM9
SYSTEM 10 = SYSTEM10   SYSTEM 11 = SYSTEM11   SYSTEM 12 = SYSTEM12
SYSTEM 13 = SYSTEM13   SYSTEM 14 = SYSTEM14   SYSTEM 15 = SYSTEM15
SYSTEM 16 = SYSTEM16   SYSTEM 17 = SYSTEM17   SYSTEM 18 = SYSTEM18
SYSTEM 19 = SYSTEM19   SYSTEM 20 = SYSTEM20   SYSTEM 21 = SYSTEM21
SYSTEM 22 = SYSTEM22   SYSTEM 23 = SYSTEM23   SYSTEM 24 = SYSTEM24
SYSTEM 25 = SYSTEM25   SYSTEM 26 = SYSTEM26   SYSTEM 27 = SYSTEM27
SYSTEM 28 = SYSTEM28   SYSTEM 29 = SYSTEM29   SYSTEM 30 = SYSTEM30
SYSTEM 31 = SYSTEM31   SYSTEM 32 = SYSTEM32
```

Displaying drive detail status

The following command syntax displays OAM drive detail status:



DRIVE(*drvname* | *libname* | ALL)
Displays the details of the current status for optical drives. If a drive name is specified, one data line describes the specified optical drive. If a library name is specified, a data line describes each optical

drive associated with the library specified. If ALL is specified, all the optical drives in the SMS configuration are displayed. To specify a drive named ALL, surround it with parentheses, for example, DISPLAY SMS,DRIVE((ALL)).

DETAIL
Displays detail status for optical drives.

L={a | *name* | *name-a*}
Specifies the location where the results of the inquiry are to be displayed, where *name* is the console name, and *a* is the display area on the console screen.

To display detail status for an individual drive, enter the following command:

```
DISPLAY SMS,DRIVE(drvname),DETAIL
```

This results in F OAM,D,DRIVE,drivename command being issued.

The following information is displayed:

```
CBR1120I OAM drive status:
DRIVE    DEVICE TY  LIBRARY  ON OP AV WP   DEV  SC  DRV  MOUNT   PEND
NAME     TYPE       NAME                   NUM  SI  NUM  VOLUME  VOLUME
drvname  devtype a  libname   b  c  d  e  ffff   g  hhh  mntvol  pndvol
-------------------------------------------------------------
XCF MEMBER NAME: membername
-------------------------------------------------------------
```

The fields displayed in each data line are as follows:

drvname
Name of the optical drive

devtype
Device type of the optical drive:

3995-111
3995 rewritable optical disk drive

3995-112
3995 write-once optical disk drive

3995-113
3995 multifunction optical disk drive

3995-131
3995 rewritable optical disk drive

3995-132
3995 write-once optical disk drive

3995-133
3995 multifunction optical disk drive

3995-SW3
3995 multifunction optical disk drive

3995-SW4
3995 multifunction optical disk drive

a
Optical drive type:

L
Library

S
Operator-accessible (stand-alone)

libname

Name of the library to which the optical drive is attached. For an operator-accessible optical drive, this field displays the name of the pseudo optical library that this drive is associated within its SCDS definition, or one of the following defaults:

- `PCTWORM` (for 3995-132 write-once drives)
- `PCTREUSE` (for 3995-131 rewritable drives)
- `P3995133` (for 3995-133 multifunction drives)
- `P3995SW3` (for 3995-SW3 drives)
- `P3995SW4` (for 3995-SW4 drives)

b

Optical drive online status:

Y

Online

N

Offline

P

Pending offline

c

Optical drive operational status:

Y

Operational

N

Not operational

d

Optical drive availability status:

Y

Available. The optical drive is online, operational, not pending offline, and not busy.

N

Not available.

e

Write Protection status:

Y

Write protection is on. Writing to this drive is not allowed.

N

Write protection is off. Writing to this drive is allowed.

The write protection status reflects the switch setting as of the last volume mount, vary online, or drive error processing.

ffff

z/OS device number of the CTC that is used to communicate with the optical drive.

g

SCSI bus address of the optical drive on the SCSI interface. Not used for 3995 and will be blank.

hhh

Drive number of the optical disk drive.

mntvol

Volume serial number of the volume that is mounted on the optical drive. If there is no mounted volume, this field is blank.

pndvol
: Volume serial number of the volume for which a mount is pending on the optical drive. If there is no pending mount, this field is blank. This field is blank when used for library-resident 3995 optical drives.

membername
: The XCF member name associated with the instance of OAM where this drive is online. If the drive is not online to any OAM system in the OAMplex, this field contains blanks. If this OAM system is not part of the OAMplex, this field contains **N/A**.

The following is a sample of DISPLAY SMS,DRIVE(LID1),DETAIL:

```
CBR1120I OAM drive status:
DRIVE     DEVICE    TY  LIBRARY   ON   OP   AV   WP   DEV   SC   DRV  MOUNT    PEND
NAME      TYPE          NAME                          NUM   SI   NUM  VOLUME   VOLUME
L1D1      3995-133  L   LIB1      Y    Y    N    N    0922         1  OP001

----------------------------------------------------
XCF MEMBER NAME: OAMXCFMEMBER1
----------------------------------------------------
```

To display detail status for drives associated with a specific optical library, enter the following command:

```
DISPLAY SMS,DRIVE(libname),DETAIL
```

This results in F OAM,D,DRIVE,libname command being issued.

The following is a sample of DISPLAY SMS,DRIVE(LIB1),DETAIL:

```
CBR1120I OAM drive status:
DRIVE  DEVICE    TY  LIBRARY   ON   OP   AV   WP  DEV   SC   DRV  MOUNT    PEND
NAME   TYPE          NAME                         NUM   SI   NUM  VOLUME   VOLUME
L1D1   3995-133  L   LIB1       Y    Y    N    N 0922           1  OP001
L1D2   3995-133  L   LIB1       Y    Y    N    N 0923           2  OP002
L1D3   3995-133  L   LIB1       Y    Y    Y    N 0924           3  OP003
L1D4   3995-133  L   LIB1       Y    Y    Y    N 0925           4  OP004
```

To display detail status for all the optical disk drives active in the SMS configuration, enter the following command:

```
DISPLAY SMS,DRIVE(ALL),DETAIL
```

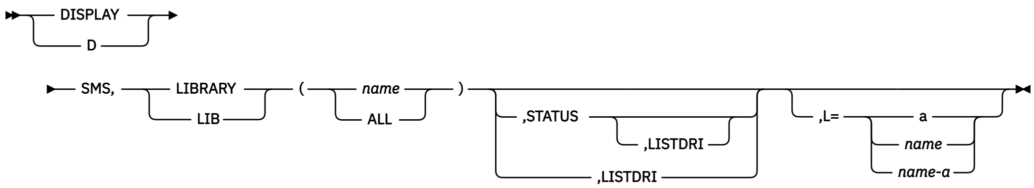
This results in F OAM,D,DRIVE command being issued.

The following is a sample of DISPLAY SMS,DRIVE(ALL),DETAIL:

```
CBR1120I OAM drive status:
DRIVE  DEVICE    TY  LIBRARY   ON   OP   AV   WP  DEV   SC   DRV  MOUNT    PEND
NAME   TYPE          NAME                         NUM   SI        VOLUME   VOLUME
L1D1   3995-133  L   LIB1       N    N    N    N 0992           1
L1D2   3995-133  L   LIB1       Y    Y    N    N 0993           2  OP002
L1D3   3995-133  L   LIB1       Y    Y    Y    N 0994           3  OP003
L1D4   3995-133  L   LIB1       Y    Y    Y    N 0995           4  OP004
L1D5   3995-133  S   PSEUDO1    N    N    N    N 0996           5
L2AD1  3995-SW3  L   LIB2       Y    Y    Y    N 19F6           1  OP012
L2AD2  3995-SW3  L   LIB2       N    Y    N    N 19F7           2
L2AD3  3995-SW3  L   LIB2       N    Y    N    N 19F8           3
L2AD4  3995-SW3  L   LIB2       N    Y    N    N 19F9           4
L2AD5  3995-SW3  L   LIB2       N    Y    N    N 19FA           5
L2AD6  3995-SW3  L   LIB2       N    Y    N    N 19FB           6
L3D1   3995-SW3  S   P3995SW3   N    Y    N    N 19F2           1
L3D1   3995-113  L   LIB2       N    Y    N    N 092A           1  OP006
L3D2   3995-113  L   LIB2       N    Y    N    N 092B           2  OP007
L3D3   3995-113  L   LIB2       Y    Y    Y    N 092C           3  OP008
L3D4   3995-113  L   LIB2       Y    Y    Y    N 092D           4  OP009
```

Displaying library online/offline connectivity

The following command syntax display OAM library status:



LIBRARY(*name* | ALL)
Displays the system connectivity and online/offline status for real libraries. If a library name is specified, there is one data line describing the specified library. If ALL is specified, there is one data line for each optical library in the configuration. To specify a library named ALL, surround it with parentheses; for example, DISPLAY SMS,LIBRARY((ALL)).

This display command applies only to real optical libraries. If the specified library is a pseudo optical library, an error message is displayed.

STATUS
Displays the system connectivity and online/offline status.

LISTDRI
Displays the online/offline status of all drives associated with this library.

L={a | *name* | *name-a*}
Specifies the location where the results of the inquiry are to be displayed, where *name* is the console name, and *a* is the display area on the console screen.

To display status for an individual library, enter the following command:

```
DISPLAY SMS,LIBRARY(libname),STATUS
```

The following information is displayed:

```
IGD002I 11.19.56    DISPLAY SMS
                                                  1 1 1 1 1 1 1
LIBRARY  CLASS      SYSTEM=    1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
name     type                  s s s s s s s s s s s s s s s s

                               1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3
LIBRARY  CLASS      SYSTEM=    7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
name     type                  s s s s s s s s s s s s s s s s
```

The fields displayed in each data line are as follows:

name
Name of the library for which system connectivity and online/offline status is displayed

type
Library type (optical or tape)

1–32
Numbers that appear after SYSTEM= indicating system IDs

s
Indications of drive status:

.
Not defined

+
Online

-
Offline

Recommendation: An online status of "+" does not necessarily mean that either the drive or the library is fully functional. To determine if the drive is both online and operational, you must issue the DISPLAY SMS,DRIVE(*drvname*),DETAIL command. To determine if the library is both online and operational, you must issue the DISPLAY SMS,LIBRARY(*name*),DETAIL command.

For more information on these SMS commands, see "Displaying drive detail status" on page 324 and "Displaying library detail status" on page 332.

The following is a sample of DISPLAY SMS,LIBRARY(*libname*),STATUS:

```
D SMS,LIBRARY(PEA13),STATUS
IGD002I 15:09:05 DISPLAY SMS 403

                                          1 1 1 1 1 1 1
LIBRARY   CLASS  SYSTEM= 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
PEA13     OPTICAL        + - . . . . . . . . . . . . . .

                         1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3
LIBRARY   CLASS  SYSTEM= 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
PEA13     OPTICAL        . . . . . . . . . . . . . . . .
***************************** LEGEND ****************************
. THE LIBRARY IS NOT DEFINED TO THE SYSTEM
+ THE LIBRARY IS ONLINE
- THE LIBRARY IS OFFLINE
P THE LIBRARY IS PENDING OFFLINE
SYSTEM 1  = SYSTEM1    SYSTEM 2  = SYSTEM2    SYSTEM 3  = SYSTEM3
SYSTEM 4  = SYSTEM4    SYSTEM 5  = SYSTEM5    SYSTEM 6  = SYSTEM6
SYSTEM 7  = SYSTEM7    SYSTEM 8  = SYSTEM8    SYSTEM 9  = SYSTEM9
SYSTEM 10 = SYSTEM10   SYSTEM 11 = SYSTEM11   SYSTEM 12 = SYSTEM12
SYSTEM 13 = SYSTEM13   SYSTEM 14 = SYSTEM14   SYSTEM 15 = SYSTEM15
SYSTEM 16 = SYSTEM16   SYSTEM 17 = SYSTEM17   SYSTEM 18 = SYSTEM18
SYSTEM 19 = SYSTEM19   SYSTEM 20 = SYSTEM20   SYSTEM 21 = SYSTEM21
SYSTEM 22 = SYSTEM22   SYSTEM 23 = SYSTEM23   SYSTEM 24 = SYSTEM24
SYSTEM 25 = SYSTEM25   SYSTEM 26 = SYSTEM26   SYSTEM 27 = SYSTEM27
SYSTEM 28 = SYSTEM28   SYSTEM 29 = SYSTEM29   SYSTEM 30 = SYSTEM30
SYSTEM 31 = SYSTEM31   SYSTEM 32 = SYSTEM32
```

The following is a sample of DISPLAY SMS,LIBRARY(ALL),STATUS:

```
D SMS,LIBRARY(ALL),STATUS
IGD002I 15:09:21 DISPLAY SMS 409
                                           1 1 1 1 1 1 1
LIBRARY   CLASS  SYSTEM= 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
ATLF4017  TAPE           + - . . . . . . . . . . . . . .
LIB1      OPTICAL        . . . . . . . . . . . . . . . .
LIB2      OPTICAL        . . . . . . . . . . . . . . . .
LIB3      OPTICAL        . . . . . . . . . . . . . . . .
LIB4      OPTICAL        . . . . . . . . . . . . . . . .
MTLA0001  TAPE           - + . . . . . . . . . . . . . .
MTLA0002  TAPE           + - . . . . . . . . . . . . . .
PCTREUSE  OPTICAL                 LIBRARY IS NOT A REAL LIBRARY
PCTWORM   OPTICAL                 LIBRARY IS NOT A REAL LIBRARY
PEA13     OPTICAL        + - . . . . . . . . . . . . . .
PEA15     OPTICAL        + - . . . . . . . . . . . . . .
PEA17     OPTICAL        + - . . . . . . . . . . . . . .
PEA19     OPTICAL        + - . . . . . . . . . . . . . .
PMA21     OPTICAL        + - . . . . . . . . . . . . . .
PWA7      OPTICAL        . . . . . . . . . + . . . . . .
PWA9      OPTICAL        + - . . . . . . . . . . . . . .
PWB8      OPTICAL        . . . . . . . . . - . . . . . .
P3995133  OPTICAL                 LIBRARY IS NOT A REAL LIBRARY
STDALONE  OPTICAL                 LIBRARY IS NOT A REAL LIBRARY

                         1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3
LIBRARY   CLASS  SYSTEM= 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
ATLF4017  TAPE           . . . . . . . . . . . . . . . .
LIB1      OPTICAL        . . . . . . . . . . . . . . . .
LIB2      OPTICAL        . . . . . . . . . . . . . . . .
LIB3      OPTICAL        . . . . . . . . . . . . . . . .
LIB4      OPTICAL        . . . . . . . . . . . . . . . .
MTLA0001  TAPE           . . . . . . . . . . . . . . . .
MTLA0002  TAPE           . . . . . . . . . . . . . . . .
PCTREUSE  OPTICAL                 LIBRARY IS NOT A REAL LIBRARY
PCTWORM   OPTICAL                 LIBRARY IS NOT A REAL LIBRARY
PEA13     OPTICAL        . . . . . . . . . . . . . . . .
PEA15     OPTICAL        . . . . . . . . . . . . . . . .
PEA17     OPTICAL        . . . . . . . . . . . . . . . .
PEA19     OPTICAL        . . . . . . . . . . . . . . . .
PMA21     OPTICAL        . . . . . . . . . . . . . . . .
PWA7      OPTICAL        . . . . . . . . . . . . . . . .
PWA9      OPTICAL        . . . . . . . . . . . . . . . .
PWB8      OPTICAL        . . . . . . . . . . . . . . . .
P3995133  OPTICAL                 LIBRARY IS NOT A REAL LIBRARY
STDALONE  OPTICAL                 LIBRARY IS NOT A REAL LIBRARY
***************************** LEGEND *****************************
. THE LIBRARY IS NOT DEFINED TO THE SYSTEM
+ THE LIBRARY IS ONLINE
- THE LIBRARY IS OFFLINE
P THE LIBRARY IS PENDING OFFLINE
SYSTEM 1  = SYSTEM1    SYSTEM 2  = SYSTEM2    SYSTEM 3  = SYSTEM3
SYSTEM 4  = SYSTEM4    SYSTEM 5  = SYSTEM5    SYSTEM 6  = SYSTEM6
SYSTEM 7  = SYSTEM7    SYSTEM 8  = SYSTEM8    SYSTEM 9  = SYSTEM9
SYSTEM 10 = SYSTEM10   SYSTEM 11 = SYSTEM11   SYSTEM 12 = SYSTEM12
SYSTEM 13 = SYSTEM13   SYSTEM 14 = SYSTEM14   SYSTEM 15 = SYSTEM15
SYSTEM 16 = SYSTEM16   SYSTEM 17 = SYSTEM17   SYSTEM 18 = SYSTEM18
SYSTEM 19 = SYSTEM19   SYSTEM 20 = SYSTEM20   SYSTEM 21 = SYSTEM21
SYSTEM 22 = SYSTEM22   SYSTEM 23 = SYSTEM23   SYSTEM 24 = SYSTEM24
SYSTEM 25 = SYSTEM25   SYSTEM 26 = SYSTEM26   SYSTEM 27 = SYSTEM27
SYSTEM 28 = SYSTEM28   SYSTEM 29 = SYSTEM29   SYSTEM 30 = SYSTEM30
SYSTEM 31 = SYSTEM31   SYSTEM 32 = SYSTEM32
```

The following is a sample of DISPLAY SMS,LIB(PEA13),LISTDRI:

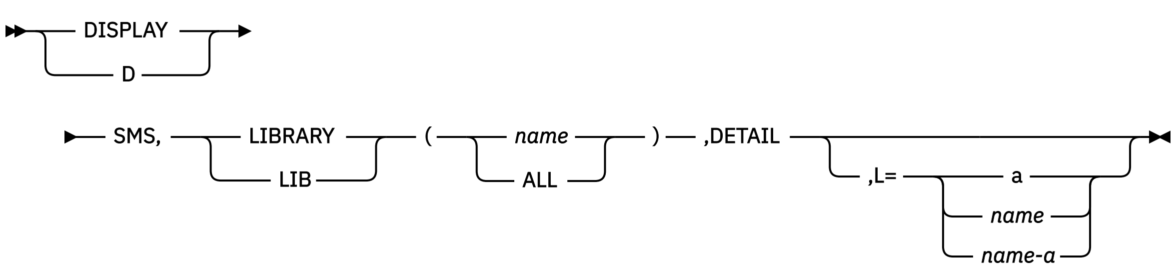
```
D SMS,LIB(PEA13),LISTDRI
IGD002I 15:09:47 DISPLAY SMS 412
                                            1 1 1 1 1 1 1
LIBRARY   CLASS  SYSTEM= 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
PEA13     OPTICAL        + - . . . . . . . . . . . . . .

                                            1 1 1 1 1 1 1
DRIVE    LIBRARY SYSTEM= 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
P13D1    PEA13           + - . . . . . . . . . . . . . .
P13D2    PEA13           + - . . . . . . . . . . . . . .
P13D3    PEA13           + - . . . . . . . . . . . . . .
P13D4    PEA13           + - . . . . . . . . . . . . . .

                         1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3
LIBRARY   CLASS  SYSTEM= 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
PEA13     OPTICAL        . . . . . . . . . . . . . . . .

                         1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3
DRIVE    LIBRARY SYSTEM= 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
P13D1    PEA13           . . . . . . . . . . . . . . . .
P13D2    PEA13           . . . . . . . . . . . . . . . .
P13D3    PEA13           . . . . . . . . . . . . . . . .
P13D4    PEA13           . . . . . . . . . . . . . . . .
***************************** LEGEND *****************************
. THE LIBRARY OR DRIVE IS NOT DEFINED TO THE SYSTEM
+ THE LIBRARY OR DRIVE IS ONLINE
- THE LIBRARY OR DRIVE IS OFFLINE
P THE LIBRARY IS PENDING OFFLINE
SYSTEM 1  = SYSTEM1    SYSTEM 2  = SYSTEM2    SYSTEM 3  = SYSTEM3
SYSTEM 4  = SYSTEM4    SYSTEM 5  = SYSTEM5    SYSTEM 6  = SYSTEM6
SYSTEM 7  = SYSTEM7    SYSTEM 8  = SYSTEM8    SYSTEM 9  = SYSTEM9
SYSTEM 10 = SYSTEM10   SYSTEM 11 = SYSTEM11   SYSTEM 12 = SYSTEM12
SYSTEM 13 = SYSTEM13   SYSTEM 14 = SYSTEM14   SYSTEM 15 = SYSTEM15
SYSTEM 16 = SYSTEM16   SYSTEM 17 = SYSTEM17   SYSTEM 18 = SYSTEM18
SYSTEM 19 = SYSTEM19   SYSTEM 20 = SYSTEM20   SYSTEM 21 = SYSTEM21
SYSTEM 22 = SYSTEM22   SYSTEM 23 = SYSTEM23   SYSTEM 24 = SYSTEM24
SYSTEM 25 = SYSTEM25   SYSTEM 26 = SYSTEM26   SYSTEM 27 = SYSTEM27
SYSTEM 28 = SYSTEM28   SYSTEM 29 = SYSTEM29   SYSTEM 30 = SYSTEM30
SYSTEM 31 = SYSTEM31   SYSTEM 32 = SYSTEM32
```

The following is a sample of DISPLAY SMS,LIBRARY(ALL),LISTDRI:

```
D SMS,LIBRARY(ALL),LISTDRI
IGD002I 15:09:21 DISPLAY SMS 409
                                           1 1 1 1 1 1 1
LIBRARY   CLASS  SYSTEM= 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
ATLF4017  TAPE           + - . . . . . . . . . . . . . .
LIB1      OPTICAL        . . . . . . . . . . . . . . . .
LIB2      OPTICAL        . . . . . . . . . . . . . . . .
LIB3      OPTICAL        . . . . . . . . . . . . . . . .
LIB4      OPTICAL        . . . . . . . . . . . . . . . .
MTLA0001  TAPE           - . . . . . . . . . . . . . . .
MTLA0002  TAPE           + - . . . . . . . . . . . . . .
PCTREUSE  OPTICAL                 LIBRARY IS NOT A REAL LIBRARY
PCTWORM   OPTICAL                 LIBRARY IS NOT A REAL LIBRARY
PEA13     OPTICAL        + - . . . . . . . . . . . . . .
PEA15     OPTICAL        + - . . . . . . . . . . . . . .
PWA9      OPTICAL        + - . . . . . . . . . . . . . .
P156A     OPTICAL        + - . . . . . . . . . . . . . .
3995133   OPTICAL                 LIBRARY IS NOT A REAL LIBRARY
STDALONE  OPTICAL                 LIBRARY IS NOT A REAL LIBRARY

                          1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3
DRIVE     LIBRARY SYSTEM= 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
P13D1     PEA13           + - . . . . . . . . . . . . . .
P13D2     PEA13           + - . . . . . . . . . . . . . .
P13D3     PEA13           + - . . . . . . . . . . . . . .
P13D4     PEA13           + - . . . . . . . . . . . . . .
P15D1     PEA15           + - . . . . . . . . . . . . . .
P15D2     PEA15           + - . . . . . . . . . . . . . .
P15D3     PEA15           + - . . . . . . . . . . . . . .
P15D4     PEA15           + - . . . . . . . . . . . . . .
P9D1      PWA9            + - . . . . . . . . . . . . . .
P9D2      PWA9            + - . . . . . . . . . . . . . .
P9D3      PWA9            + - . . . . . . . . . . . . . .
P9D4      PWA9            P . . . . . . . . . . . . . . .
P156AD1   P156A           + - . . . . . . . . . . . . . .
P156AD2   P156A           + . . . . . . . . . . . . . . .
P156AD3   P156A           - + . . . . . . . . . . . . . .
P156AD4   P156A           - + . . . . . . . . . . . . . .
P156AD5   P156A           - + . . . . . . . . . . . . . .
P156AD6   P156A           - + . . . . . . . . . . . . . .
LISTDRI IS IGNORED FOR PSEUDO AND TAPE LIBRARIES
***************************** LEGEND *****************************
. THE LIBRARY OR DRIVE IS NOT DEFINED TO THE SYSTEM
+ THE LIBRARY OR DRIVE IS ONLINE
- THE LIBRARY OR DRIVE IS OFFLINE
P THE LIBRARY IS PENDING OFFLINE
SYSTEM 1  = SYSTEM1    SYSTEM 2  = SYSTEM2    SYSTEM 3  = SYSTEM3
SYSTEM 4  = SYSTEM4    SYSTEM 5  = SYSTEM5    SYSTEM 6  = SYSTEM6
SYSTEM 7  = SYSTEM7    SYSTEM 8  = SYSTEM8    SYSTEM 9  = SYSTEM9
SYSTEM 10 = SYSTEM10   SYSTEM 11 = SYSTEM11   SYSTEM 12 = SYSTEM12
SYSTEM 13 = SYSTEM13   SYSTEM 14 = SYSTEM14   SYSTEM 15 = SYSTEM15
SYSTEM 16 = SYSTEM16   SYSTEM 17 = SYSTEM17   SYSTEM 18 = SYSTEM18
SYSTEM 19 = SYSTEM19   SYSTEM 20 = SYSTEM20   SYSTEM 21 = SYSTEM21
SYSTEM 22 = SYSTEM22   SYSTEM 23 = SYSTEM23   SYSTEM 24 = SYSTEM24
SYSTEM 25 = SYSTEM25   SYSTEM 26 = SYSTEM26   SYSTEM 27 = SYSTEM27
SYSTEM 28 = SYSTEM28   SYSTEM 29 = SYSTEM29   SYSTEM 30 = SYSTEM30
SYSTEM 31 = SYSTEM31   SYSTEM 32 = SYSTEM32
```

Displaying library detail status

The syntax of the DISPLAY SMS command for detail library status is:



LIBRARY(*name* | ALL)
Specifies the name of the optical library to be displayed. If a library name is specified, there is one data line describing the specified library. If ALL is specified, there is one data line for each optical library in the configuration. To specify a library named ALL, surround it with parentheses; for example, DISPLAY SMS,LIBRARY((ALL)).

DETAIL
Displays detail status for optical libraries.

L={a | *name* | *name-a*}
Specifies the location where the results of the inquiry are to be displayed, where *name* is the console name, and *a* is the display area on the console screen.

To display detail status for an individual library, enter the following command:

```
DISPLAY SMS,LIBRARY(name),DETAIL
```

This results in F OAM,D,LIB,libname command being issued.

The following information is displayed:

```
CBR1110I OAM library status:
OPTICAL  DEVICE  ATT  USE  AVL  TOT  EMP  SCR  PT  DEV  ON  OP  IO   LIB    READ
LIBRARY  TYPE    DRV  DRV  DRV  SLT  SLT  VOL      NUM          ST   CMD   COUNT
olibname devtype aaa  bbb  ccc  ddd  eee  fff   g  hhhh  i   j   k   lbcmd rdcnt
DEFAULT PSEUDO LIB: def-plib-name
DEFAULT MEDIA TYPE: def-mediatype
XCF MEMBER NAME: membername
3995 MICROCODE LEVEL: EC number
```

The fields displayed in each data line are as follows:

olibname
Name of the optical library

devtype
Device type of the optical library:

3995-111
3995 rewritable library, extension to a 3995-131

3995-112
3995 WORM library, extension to a 3995-132

3995-113
3995 multifunction library, extension to a 3995-133

3995-132
3995 WORM library and controller

3995-133
3995 multifunction library and controller

3995-C3A
3995 multifunction library controller

3995-C12
3995 multifunction library, extension to a 3995-C32

3995-C16
3995 multifunction library, extension to a 3995-C36

3995-C18
3995 multifunction library, extension to a 3995-C38

3995-C32
3995 multifunction library, attaches to a 3995-C3A

3995-C34
3995 multifunction library, attaches to a 3995-C3A

3995-C36
3995 multifunction library, attaches to a 3995-C3A

3995-C38
3995 multifunction library, attaches to a 3995-C3A

aaa
Number of optical drives attached to the optical library

bbb
Number of usable optical drives (online, operational, and not pending offline)

ccc
Number of available optical drives (online, operational, and not in use)

ddd
Total number of storage slots in the optical library

eee
Number of empty storage slots in the optical library

fff
Number of scratch volumes in the optical library

g
Active path to the optical library:

P
Primary

A
Alternate

BLANK
Pseudo library or 3995 library

hhhh
Device number of the active CTC, or blank for pseudo libraries

i
Optical library online status:

Y
Online

N
Offline

P
Pending offline

j
Optical library operational status:

Y
Operational

N
Not operational

k
Optical library input/output station operational status, as follows:

Y
Operational

N
Not operational

*****
An error occurred while trying to get status

BLANK
: Library not attached or library has no I/O station

lbcmd
: For the 3995: (except for 3995-C3A) REMAP indicates that a REMAP of the library is in progress, RMPND indicates that a full library audit is being processed, and AUDIT indicates that a full library audit of the library is in progress. If not REMAP, RMPND, or AUDIT, this field contains the library command most recently sent to the optical library.

rdcnt
: The number of read requests waiting or in progress for optical volumes that are resident in this optical library.

def-plib-name
: The name of the pseudo library that will be assigned to any volume that is ejected from this library if that volume does not already have a pseudo library associated with it.

def-mediatype
: The media types that can be entered into the optical library and which media types can be written to if they already reside in the library. This value is specified on the 3995 Library Define panel in ISMF.

membername
: The XCF member name associated with the instance of OAM where this library is online.

microcode-level
: This field displays the 3995 microcode level.

The following is a sample of DISPLAY SMS,LIBRARY(LIB1),DETAIL:

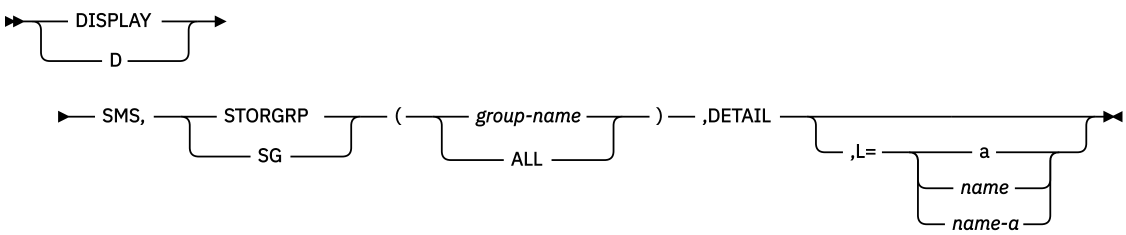
```
CBR1110I OAM library status:
OPTICAL  DEVICE   ATT  USE  AVL  TOT  EMP  SCR  PT  DEV   ON  OP  IO  LIB   READ
LIBRARY  TYPE     DRV  DRV  DRV  SLT  SLT  VOL      NUM           ST  CMD  COUNT
LIB1     3995-C36   6    1    1  156  153    4     19F4    Y   Y   Y  LM       0
-------------------------------------------------------
DEFAULT PSEUDO LIB: PLIB4
DEFAULT MEDIA TYPE: 3995REWR
XCF MEMBER NAME: OAMSYS1
3995     MICROCODE LEVEL: 14028Q
-------------------------------------------------------
```

The following is a sample of DISPLAY SMS,LIBRARY(ALL),DETAIL:

```
CBR1110I OAM library status:
OPTICAL  DEVICE   ATT  USE  AVL  TOT  EMP  SCR  PT  DEV   ON  OP  IO  LIB   READ
LIBRARY  TYPE     DRV  DRV  DRV  SLT  SLT  VOL      NUM           ST  CMD  COUNT
LIB1     3995-131  4    4    4  144   43   12   P 0900    Y   Y   Y  LM      0
LIB2     3995-111  4    4    3  144   48    8   P 0908    Y   Y   Y  LACT     0
LIB3     3995-132  4    4    3  144   45    6   P 0910    Y   Y   Y  LM       0
LIB4     3995-112  4    4    4  144   38    6   P 0918    Y   Y   Y  LM       4
LIB5     3995-133  4    4    0  144   11   26   P 0920    Y   Y   Y  LD      12
LIB6     3995-113  4    4    1  144   10   36   P 0928    Y   Y   Y  LM       0
LIB7     3995-C36  6    1    1  156  153    4     19F4    Y   Y   Y  LM       0
STDALONE 9246      0    0    0    0    0    0        0    N   Y              0
P3995SW3 3995-SW3  0    0    0    0    0    0        0    Y   Y              0
PCTREUS  3995-131  0    0    0    0    0    0        0    Y   Y              0
PCTWORM  3995-132  0    0    0    0    0    0        0    Y   Y              0
P3995133 3995-133  0    0    0    0    0    0        0    Y   Y              0
P3995SW4 3995-SW4  0    0    0    0    0    0        0    Y   Y              0
PSEUDO1            0    0    0    0    0    0        0    Y   Y              0
```

Displaying Storage Group Status

The following command syntax displays storage group status:



DISPLAY
: Specifies the OAM display command.

STORGRP(*group-name* | ALL)
: Displays the status of an Object or Object Backup storage group. If *group-name* is specified, one data line displays the status of the requested storage group. If ALL is selected, a data line will be displayed for each Object Backup storage group, and for each Tape Storage Group defined in the active configuration. If *group-name* is ALL, you must enclose the ALL parameter in two sets of parentheses, as follows:

```
D SMS,STORGRP((ALL))
```

DETAIL
: Displays detailed *object-group* status.

L={a | *name* | *name-a*}
: Specifies the location where the results of the inquiry are to be displayed, where *name* is the console name, and *a* is the display area on the console screen.

To display Object or Object Backup storage group status, enter the following command:

```
DISPLAY SMS,STORGRP(object-group),DETAIL
```

This results in F OAM,D,GROUP,groupname command being issued.

Specifying the previous command displays the status of the requested storage group:

```
CBR1130I OAM storage group status:
OBJECT   TY  REQ   OSMC     BACKUP    BACKUP    RET  DEL
STORGRP      COUNT SYSTEM   STORGRP1  STORGRP2  PRO  PRO
sgname   a   bbbbb  sysname  objbusg1  objbusg2  n    o
TAPE     DATA      L2TAPE   L2DATA
UNIT     CLASS     UNIT     CLASS
unitname dataclass unitname2 dataclass2
Library Names:   libname1 libname2 libname3 libname4
                 libname5 libname6 libname7 libname8
DSL2 Directory:  dsl2dir
DSL2 Type:       dsl2type
TAPE      LIBRARY
STORGRP   NAMES
sgname    libname1 libname2 libname3 libname4
          libname5 libname6 libname7 libname8


...
If the command issued was:
DISPLAY SMS,STORGRP(storage-group-name),DETAIL

and the storage group that is requested is an object storage group
or an object backup storage group, additional data lines are
displayed as follows:

                         OPTICAL      TAPE    TSL1    TSL2
ALL VOLUMES FULL:              c         d       u       v
WRITABLE VOLUMES:          eeeee     fffff   wwwww   xxxxx
FULL VOLUMES:              sssss     ttttt   yyyyy   zzzzz
DRIVE START THRESHOLD:     ggggg     hhhhh
Volume Full Threshold:     iiiii       jjjjj
Reinit / Recycle Mode:     kkkkkkkk    pppppppp
# of Active Drives:        lllll       mmmmm
Recall Status:             qqqqqqqq    rrrrrrrr
Recall to disk sublevel s
…
c          Indicates whether all of the optical volumes
```

```
            that belong to this storage group are marked
            full. Valid values are:
            Y          All optical volumes are full
            N          At least one optical volume has
                   available space
            -         There are no optical volumes in this
                       storage group
d           Indicates whether all of the tape volumes that
            belong to this storage group are marked full.
            Valid values are:
            Y         All tape volumes are full
            N         At least one tape volume has
                   available space
            -          There are no tape volumes in this
                   storage group
n           Indicates whether retention-protection is enabled
            for this object storage group.
            Valid Values are:
            Y     Retention-protection is enabled for this object
                   storage group. Objects stored in this storage group
                   will have a retention-protected attribute associated
                   with them for the life of the object. Retention-
                   protected objects can not be deleted prior to their
                   expiration date, and additionally, their expiration
                   dates can be moved out to a later date, but can never
                   be brought in to an earlier date.
            N     Retention-protection is disabled for this object
                   storage group. Objects stored into this storage group
                   will not have a retention-protected attribute associated
                   with them for the life of the object. Note: even if a
                   given object is not being protected from premature
                   deletion by the retention-protection attribute
                   specifically, it could possibly be protected by another
                   mechanism such as deletion-protection or deletion-hold.
            -     A dash is displayed for object backup storage groups
                   since retention-protection only applies to object
                   storage groups.
```

```
Note: Retention-protection status is determined by the OAM
            Retention Protection parameter in the SMS object storage group
            definition.
o           Indicates whether deletion-protection is enabled
            for this object storage group.
            Valid values are:
            Y     Deletion-protection is enabled for this object
                   storage group. Objects in this storage group can
                   not be deleted prior to their expiration date.
                   However, unlike retention-protection, deletion-
                   protection does not provide any safeguards for
                   preventing an object's expiration date from being
                   brought into an earlier date. Note: retention-
                   protection takes precedence over deletion-protection.
                   For example; If a given object is both retention-
                   protected and deletion-protected, then it defaults
                   to retention-protection for the life of the object,
                   and the expiration date could not be manipulated to
                   an earlier date.
            N     Deletion-protection is disabled for this object
                   storage group. Objects in this storage group are
                   not currently subject to deletion-protection. Note:
                   Although, in this case, the objects are not being
                   protected from premature deletion by the deletion-
                   protection attribute specifically, but they could
                   possibly be protected by another mechanism such as
                   retention-protection or deletion-hold.
            -     A dash is displayed for object backup storage groups
                   since deletion-protection only applies to object
                   storage groups.
u           Indicates whether all of the tape sublevel 1 volumes that
            belong to this storage group are marked full.
            Valid values are:
            Y         All tape sublevel 1 volumes are full
            N         At least one tape sublevel 1 volume has
                   available space
            -         There are no tape sublevel 1 volumes in this
                   storage group
            N/A       Not applicable if this storage group is a
                       backup group
v          Indicates whether all of the tape sublevel 2 volumes that
            belong to this storage group are marked full.
            Valid values are:
```

```
          Y         All tape sublevel 2 volumes are full
          N         At least one tape sublevel 2 volume has
                 available space
          -         There are no tape sublevel 2 volumes in this
                 storage group
          N/A       Not applicable if this storage group is a
                 backup group
```

```
eeeee      Number of optical volumes in this storage
           group that have space available for writes and
           the volume writable indicator set to 'Y'.
fffff      Number of total tape volumes in this storage group
           that have space available for writes and the
           volume writeable indicator set to 'Y'.
wwwww      Number of tape sublevel 1 volumes in this storage group
           that have space available for writes and the
           volume writeable indicator set to 'Y'. If the storage
           group is a backup group, then 'N/A' is displayed.
xxxxx      Number of tape sublevel 2 volumes in this storage group
           that have space available for writes and the
           volume writeable indicator set to 'Y'. If the storage
           group is a backup group, then 'N/A' is displayed.
sssss      Number of optical volumes in this storage
           group that have been marked full or permanently
           full with the volume full indicator set to 'Y' or 'P'.
ttttt      Number of total tape volumes in this storage group
           that have been marked full or permanently full
           with the volume full indicator set to 'Y' or 'P'.
yyyyy      Number of tape sublevel 1 volumes in this storage group
           that have been marked full or permanently full
           with the volume full indicator set to 'Y' or 'P'. If the
           storage group is a backup group, then 'N/A' is displayed.
zzzzz      Number of tape sublevel 2 volumes in this storage group
           that have been marked full or permanently full
           with the volume full indicator set to 'Y' or 'P'. If the
           storage group is a backup group, then 'N/A' is displayed.

…
lllll      Number of active optical drives, currently
           processing work for this storage group.
mmmmm      Number of tape tasks actively processing
           requests for this storage group.
```

The fields displayed in each data line are as follows:

sgname
: Name of the Object storage group or Object Backup storage group being displayed.

a
: Object storage group type:

  B
  : Object Backup storage group

  G
  : Object storage group

  N
  : Nongroup. Not used.

  S
  : Scratch

bbbbb
: Number of write requests for the storage groups that are pending in OAM.

sysname
: Name of the system where OSMC processing is run for the Object or Object Backup storage group. This system name is defined in the storage group definition in the active SMS configuration. OSMC processes storage groups automatically when the cycle start window occurs, or manually when a user requests a full OSMC cycle on that system. This field is blank if you do not define a specific OSMC system name. If this field is blank, OSMC processes storage groups when the cycle window starts, or manually when a user requests a full OSMC cycle on any system where OAM and OSMC are active.

objbusg1

Name of the Object Backup storage group that is specified in the CBROAM*xx* member of PARMLIB, where the first backup copies of objects in this object storage group are stored. This field is *--N/A--* if the displayed storage group is an Object Backup storage group. This field contains hyphens if the displayed Object storage group does not have a defined first backup storage group.

objbusg2

Name of the Object Backup storage group that is specified in the CBROAM*xx* member of PARMLIB, where the second backup copies of objects in this object storage group are stored. This field is *--N/A--* if the displayed storage group is an Object Backup storage group. This field contains hyphens if the displayed Object storage group does not have a defined second backup storage group.

unitname

MVS esoteric or generic assigned to this storage group that OAM uses for tape sublevel 1 when allocating a tape drive for a scratch volume during a write request to this storage group. This field is defined using the SETOAM statements in the CBROAM*xx* member of PARMLIB or SETOAM update operator commands. This field contains hyphens if the displayed storage group does not have a tape sublevel 1 unit name that is associated with it.

unitname2

MVS esoteric or generic assigned to this storage group that OAM uses for tape sublevel 2 when allocating a tape drive for a scratch volume during a write request to this storage group. This is defined using the SETOAM statements in the CBROAM*xx* member of PARMLIB or SETOAM update operator commands. If no L2TAPEUNITNAME is specified for this storage group and the group is not a backup storage group, this field contains hyphens. For a backup storage group, this field contains '--N/A--'.

libname1–libname8

Names of one-to-eight real optical libraries or a single pseudo library that is associated with the storage group.

dataclass

The data class that is associated with this sublevel 1 object tape volume. The data class is defined using the SETOAM statements in the CBROAM*xx* member of PARMLIB or SETOAM update operator commands. This field contains hyphens if this storage group does not have a tape sublevel 1 data class (DATACLASS) associated with it.

dataclass2

The data class that is associated with this sublevel 2 object tape volume. The data class is defined using the SETOAM statements in the CBROAM*xx* member of PARMLIB. This field contains hyphens if no L2DATACLASS is specified for this storage group and the group is not a backup storage group. For a backup storage group, this field contains *--N/A--*.

c

Indicates whether all optical volumes associated with this storage group are full. Valid values are:

Y

All optical volumes are full.

N

At least one optical volume has available space.

-

There are no optical volumes in this storage group.

d

Indicates whether all tape volumes associated with this storage group are full. Valid values are:

Y

All tape volumes are full.

N

At least one tape volume has available space.

-

There are no tape volumes in this storage group.

n

Indicates whether retention-protection is enabled for this object storage group. Valid values are:

Y

Retention-protection is enabled for this object storage group. Objects stored in this storage group will have a retention-protected attribute associated with them for the life of the object. Retention-protected objects can not be deleted prior to their expiration date, and additionally, their expiration dates can be moved out to a later date, but can never be brought in to an earlier date.

N

Retention-protection is disabled for this object storage group. Objects stored into this storage group will not have a retention-protected attribute associated with them for the life of the object.

Note: Even if a given object is not being protected from premature deletion by the retention-protection attribute specifically, it could possibly be protected by another mechanism such as deletion-protection or deletion-hold

-

A dash is displayed for object backup storage groups since retention-protection only applies to object storage groups.

Note: Retention-protection status is determined by the OAM Retention Protection parameter in the SMS object storage group definition.

o

Indicates whether deletion-protection is enabled for this object storage group. Valid values are:

Y

Deletion-protection is enabled for this object storage group. Objects in this storage group can not be deleted prior to their expiration date. However, unlike retention-protection, deletion-protection does not provide any safeguards for preventing an object's expiration date from being brought into an earlier date.

Note: Retention-protection takes precedence over deletion-protection. For example; If a given object is both retention-protected and deletion-protected, then it defaults to retention-protection for the life of the object, and the expiration date could not be manipulated to an earlier date.

N

Deletion-protection is disabled for this object storage group. Objects in this storage group are not currently subject to deletion-protection.

Note: Although, in this case, the objects are not being protected from premature deletion by the deletion-protection attribute specifically, but they could possibly be protected by another mechanism such as retention-protection or deletion-hold.

-

A dash is displayed for object backup storage groups since deletion-protection only applies to object storage groups.

u

Indicates whether all of the tape sublevel 1 volumes that belong to this storage group are marked full. Valid values are:

Y

All tape sublevel 1 volumes are full.

N

At least one tape sublevel 1 volume has available space.

-

There are no tape sublevel 1 volumes in this storage group.

N/A

Not applicable if this storage group is a backup group.

v
: Indicates whether all of the tape sublevel 2 volumes that belong to this storage group are marked full. Valid values are:

  Y
  : All tape sublevel 2 volumes are full.

  N
  : At least one tape sublevel 2 volume has available space.

  -
  : There are no tape sublevel 2 volumes in this storage group.

  N/A
  : Not applicable if this storage group is a backup group.

eeeee
: Number of optical volumes in this storage group that have space available for writes and the volume writeable indicator set to 'Y'.

fffff
: Number of total tape volumes in this storage group that have space available for writes and the volume writeable indicator set to ’Y’.

wwwww
: Number of tape sublevel 1 volumes in this storage group that have space available for writes and the volume writeable indicator set to 'Y'. If the storage group is a backup group, then 'N/A' is displayed.

xxxxx
: Number of tape sublevel 2 volumes in this storage group that have space available for writes and the volume writeable indicator set to 'Y'. If the storage group is a backup group, then 'N/A' is displayed.

sssss
: Number of optical volumes in this storage group that have been marked full or permanently full with the volume full indicator set to 'Y' or 'P'.

ttttt
: Number of total tape volumes in this storage group that have been marked full or permanently full with the volume full indicator set to 'Y' or 'P'.

yyyyy
: Number of tape sublevel 1 volumes in this storage group that have been marked full or permanently full with the volume full indicator set to 'Y' or 'P'. If the storage group is a backup group, then 'N/A' is displayed.

zzzzz
: Number of tape sublevel 2 volumes in this storage group that have been marked full or permanently full with the volume full indicator set to 'Y' or 'P'. If the storage group is a backup group, then 'N/A' is displayed.

ggggg
: Optical drive startup threshold for this storage group. When the number of requests per active optical drive exceeds the threshold, a new optical drive can be started for this storage group.

hhhhh
: Tape drive startup threshold for this storage group. When the number of requests for this storage group exceeds this threshold, a new tape drive task can be started for this storage group.

iiiii
: Optical volume full threshold. When the number of kilobytes that are free on an optical volume belonging to this storage group drops below this threshold value, the volume is marked full.

jjjjj
: Tape volume full threshold. When the number of kilobytes that are free on a tape volume belonging to this storage group drops below this threshold value, the volume is marked full.

kkkkkkkk
: The optical reinitialization mode for this storage group. When a reusable optical cartridge expires or is recycled using the RECYCLE option on a MOVEVOL command, both volumes on that cartridge are

reinitialized. The volumes can be returned to scratch, or remain assigned to their current storage group. Valid values are:

GROUP
Remain in the currently assigned storage group.

SCRATCH
Return to scratch.

pppppppp
Tape recycle mode. When the tape cartridge in this storage group no longer contains active objects because the volume is expired, or is recycled using the START RECYCLE command or the RECYCLE option on a MOVEVOL command. The tape volumes can be returned to MVS scratch, OAM scratch, or remain assigned to their current storage group. Valid values are:

GROUP
Remain in the currently assigned storage group.

OAMSCR
Return to OAM scratch.

MVSSCR
Return to MVS scratch.

lllll
The number of optical drives actively processing requests for this storage group.

mmmmm
The number of tape tasks actively processing requests for this storage group.

qqqqqqqq
Optical Immediate Recall to Disk (IRD) Status. Indicates the current IRD setting for objects residing on optical media. These values are based on SETOSMC statements in the CBROAMxx Parmlib member. Valid values are:

nnn
Implicit recalls are enabled as a result of RECALLOPTICAL or RECALLALL keywords specified in a SETOSMC statement. When an object from this storage group is retrieved from optical, it will be recalled to DB2 sublevel or file system sublevel. *nnn* represents the number of days an implicitly recalled object will reside on DB2.

EXPLICIT
Implicit recalls disabled due to RECALLNONE specified and/or RECALLOPTICAL or RECALLALL keywords NOT specified in a SETOSMC statement. Recalls will occur only by OSREQ invocation.

DISABLED
Implicit and explicit recalls disabled as a result of RECALLOFF(ON) or MAXRECALLTASKS(0) specified in a SETOSMC Statement.

If the storage group displayed is an object backup storage group, this field will contain "--N/A--".

rrrrrrrr
Tape Immediate Recall to Disk (IRD) Status. Indicates the current IRD setting for objects residing on tape media. These values are based on SETOSMC statements in the CBROAMxx Parmlib member. Valid values are:

nnn
Implicit recalls are enabled as a result of RECALLTAPE or RECALLALL keywords specified in a SETOSMC statement. When an object from this storage group is retrieved from optical, it will be recalled to DB2 sublevel or file system sublevel. *nnn* represents the number of days an implicitly recalled object will reside on DB2.

EXPLICIT
Implicit recalls disabled due to RECALLNONE specified and/or RECALLTAPE or RECALLALL keywords NOT specified in a SETOSMC statement. Recalls will occur only by OSREQ invocation.

DISABLED
: Implicit and explicit recalls disabled as a result of RECALLOFF(ON) or MAXRECALLTASKS(0) specified in a SETOSMC Statement.

If the storage group displayed is an object backup storage group, this field will contain "--N/A--".

s
: Disk sublevel in which recalled objects will be written. The disk sublevel is defined using the SETOSMC statement in the CBROAMxx member of PARMLIB or SETOSMC update operator command. For Object Backup storage groups this value will contain "--N/A--".

The following is a sample of DISPLAY SMS,STORGRP(GROUP01),DETAIL:

```
CBR1130I OAM storage group status:
OBJECT   TY  REQ   OSMC     BACKUP    BACKUP   RET  DEL
STORGRP     COUNT  SYSTEM   STORGRP1  STORGRP2 PRO  PRO
GROUP01   G     0  SYSTEM1  IMAFIRST  IMSECOND N    N
TAPE      DATA      L2TAPE    L2DATA
UNIT      CLASS     UNIT      CLASS
3490      VTSM2CU   3490      VTSM2CU
Library Names:   PEA15
DSL2 Directory:  /u/group01
DSL2 Type:       ZFS
                         OPTICAL      TAPE    TSL1    TSL2
All Volumes Full:              -         N       N       N
Writable Volumes:              0         2       1       1
Full Volumes:                  0         0       0       0
Drive Start Threshold:       200      9999
Volume Full Threshold:       900         0
Reinit / Recycle Mode:   SCRATCH    OAMSCR
# of Active Drives:            0         0
Recall Status:          EXPLICIT  EXPLICIT
Recall to disk sublevel 2
```

The following is a sample of DISPLAY SMS,STORGRP(IMAFIRST),DETAIL, where IMAFIRST is a backup storage group:

```
CBR1130I OAM storage group status:
OBJECT   TY  REQ   OSMC     BACKUP    BACKUP   RET  DEL
STORGRP     COUNT  SYSTEM   STORGRP1  STORGRP2 PRO  PRO
IMAFIRST  B     0  SYSTEM1  --N/A--   --N/A--  -    -
TAPE      DATA      L2TAPE    L2DATA
UNIT      CLASS     UNIT      CLASS
3490      VTSM2CU   --N/A--   --N/A--
Library Names:   PEA15
DSL2 Directory:  --N/A--
DSL2 Type:       --N/A--
                         OPTICAL      TAPE    TSL1    TSL2
All Volumes Full:              -         N     N/A     N/A
Writable Volumes:              0         2     N/A     N/A
Full Volumes:                  0         5     N/A     N/A
Drive Start Threshold:        99      9999
Volume Full Threshold:         1         0
Reinit / Recycle Mode:   SCRATCH    OAMSCR
# of Active Drives:            0         0
Recall Status:           --N/A--   --N/A--
Recall to disk sublevel --N/A--
```

The following is a sample of DISPLAY SMS,STORGRP(ALL),DETAIL:

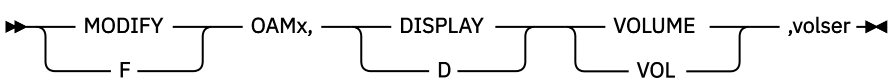
```
OBJECT   TY  REQ   OSMC     BACKUP    BACKUP   RET  DEL
STORGRP     COUNT  SYSTEM   STORGRP1  STORGRP2 PRO  PRO
GROUP01   G     0  SYSTEM1  IMAFIRST  IMSECOND N    N
TAPE      DATA      L2TAPE    L2DATA
UNIT      CLASS     UNIT      CLASS
3490      VTSM2CU   3490      VTSM2CU
Library Names:  PEA15
DSL2 Directory: /u/group01
DSL2 Type:      ZFS
OBJECT   TY  REQ   OSMC     BACKUP    BACKUP   RET  DEL
STORGRP     COUNT  SYSTEM   STORGRP1  STORGRP2 PRO  PRO
GROUP02   G     0  SYSTEM1  IMAFIRST  IMSECOND N    N
TAPE      DATA      L2TAPE    L2DATA
UNIT      CLASS     UNIT      CLASS
3490      VTSM2CU   3490      VTSM2CU
Library Names:  PEA15
```

```
DSL2 Directory:
DSL2 Type:
OBJECT   TY  REQ   OSMC     BACKUP    BACKUP   RET  DEL
STORGRP     COUNT  SYSTEM   STORGRP1  STORGRP2 PRO  PRO
IMAFIRST  B     0  SYSTEM1  --N/A--   --N/A--  -    -
TAPE      DATA      L2TAPE    L2DATA
UNIT      CLASS     UNIT      CLASS
3490      VTSM2CU   --N/A--   --N/A--
Library Names:  PEA15
DSL2 Directory: --N/A--
DSL2 Type:      --N/A--
OBJECT    TY  REQ   OSMC     BACKUP    BACKUP   RET  DEL
STORGRP     COUNT  SYSTEM   STORGRP1  STORGRP2 PRO  PRO
IMSECOND  B     0  SYSTEM1  --N/A--   --N/A--  -    -
TAPE      DATA      L2TAPE    L2DATA
UNIT      CLASS     UNIT      CLASS
3490      VTSM2CU   --N/A--   --N/A--
Library Names:  PEA15
DSL2 Directory: --N/A--
DSL2 Type:      --N/A--
TAPE      LIBRARY
STORGRP   NAMES
SGATL     ATL10001
SGATLOBJ  ATL10001  ATLCA030
SGMIXED   ATL10001
SGMTL     MTL13590
SGMTLOBJ  MTL13590
SGVTS     ATLBA999
SGVTSOBJ  ATLBA999  ATLCA030
```

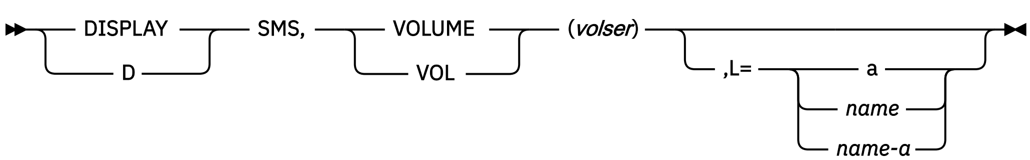
Displaying volume status

The OAM display volume command shows information about tape and optical volumes associated with object support.

The following command displays volume status:



In a classic OAM configuration, or when issuing the command to the Tape Library address space in a multiple OAM configuration, the following command can also be used and will be transformed into the above version of the command:



In both cases, volser specifies the volume for which information is to be displayed. For an optical volume (only supported in a classic configuration), information about the opposite-side volume will also be shown.

L={*a* | *name* | *name-a*}
: Specifies the location where the results of the inquiry are to be displayed, where *name* is the console name, and *a* is the display area on the console screen.

There is no option to display all volumes known to the system. However, you can use ISMF to display the optical and tape volume lists.

Optical Volume Status

For an optical volume, the following information is displayed in message CBR1140I:

```
CBR1140I OAM volume status

VOLUME  STORAGE RD  WR  WP LOST  FREE SPACE     MOUNTED  PENDING  REQ
        GROUP             FLAG  (KB)      (%)  DRIVE    MOUNT     CT
volser  sgname  a   b   c  d     freespac fff%  mdrvname pdrvname ggg
oppvol  sgname  a   b   c  d     freespac fff%  mdrvname pdrvname ggg
MEDIA TYPE: mediatyp
media descript {WORM|rewritable|unknown} optical disk media.
LIBRARY: libname
SHELF LOC: shelfloc
PSEUDO LIBRARY: plibname
OWNER: owner-information
XCF MEMBER NAME: member-name
BACKUP TYPE: {BACKUP1 | BACKUP2}
CREATION DATE: createdate     ENTER-EJECT DATE: eedate
VOLSER:            volser     oppvol

LAST WRITTEN DATE: lwdate     lwdate
LAST MOUNTED DATE: lmdate     lmdate
EXPIRATION DATE:   expdate    expdate
status
```

The fields displayed in each data line are as follows:

volser
: Volume serial number of the requested optical volume.

oppvol
: Volume serial number of the opposite side of the requested optical volume.

sgname
: Name of the storage group to which the optical volume belongs.

a
: Optical volume label readability status:

  Y
  : Readable

  N
  : Unreadable

b
: Optical volume writability status:

  Y
  : Writable

  N
  : Not writable

c
: Optical volume write protection status:

  Y
  : Write protected

  N
  : Not write protected

d
: Volume lost indicator:

  Y
  : Volume is marked lost

  N
  : Volume is not marked lost

freespac
Remaining volume space of the requested optical volume in kilobytes (KB).

fff%
Percentage of free space on the optical volume. For a full optical volume, this field contains "FULL".

mdrvname
Name of the drive where this optical volume is mounted. If the optical volume is not mounted, this field contains blanks.

pdrvname
For a 3995 drive, this field displays "YES" if a mount is pending for this optical volume.

ggg
Number of read requests for this optical volume that are pending in OAM.

mediatyp
8-character media type of the requested optical volume. Valid values are as follows:

3995-1RW
3995 5.25-inch, single-density (650 MB), rewritable optical disk media

3995-1WO
3995 5.25-inch, single-density (650 MB), WORM optical disk media

3995-2RW
3995 5.25-inch, double-density (1300 MB), rewritable optical disk media

3995-2WO
3995 5.25-inch, double-density (1300 MB), WORM optical disk media

3995-4RW
3995 5.25-inch, quad-density (2600 MB) rewritable optical disk media

3995-4WO
3995 5.25-inch, quad-density (2600 MB) WORM optical disk media

3995-8RW
3995 5.25-inch, 8*x*-density (5.2 GB) rewritable optical disk media

3995-8WO
3995 5.25-inch, 8*x*-density (5.2 GB) WORM optical disk media

media_descript
72-character description for the requested optical volume.

libname
Name of the library in which the optical volume resides. This field appears only for a library-resident optical volume.

shelfloc
Shelf location where the optical volume is found. This field appears only for a shelf-resident optical volume.

plib-name
The pseudo library name that this volume is assigned to when it is shelf resident.

owner-information
Owner information from the optical volume label.

member-name
The XCF member name of the OAM that is controlling the optical volume, blank, or -N/A-.

BACKUP1 | BACKUP2
The backup volume type for this volume. This line is displayed only if the volume type is "B" (a backup volume). This volume contains first (BACKUP1) or second (BACKUP2) backup copies written to the Object Backup storage group to which this volume belongs.

volser
The volume serial number of the requested optical volume.

oppvol
The volume serial number of the opposite side of the requested optical volume.

createdate
Date the optical volume was created in the format YYYY-MM-DD.

lwdate
Date the optical volume was last written to in the format YYYY-MM-DD.

lmdate
Date the optical volume was last mounted in the format YYYY-MM-DD.

eedate
Date the optical volume was last entered or ejected from the library in the format YYYY-MM-DD.

expdate
Expiration date of the optical volume in the format YYYY-MM-DD.

status
Additional optical volume status:

- Optical volumes are not in their assigned optical library slot.
- The optical library slot assigned to these optical volumes is empty or contains different optical volumes.

For an optical volume belonging to an Object storage group, the command `D SMS,VOLUME(WG830B)` displays the following information:

```
CBR1140I OAM volume status:
VOLUME STORAGE  RD WR WP LOST  FREE SPACE   MOUNTED  PENDING   REQ
       GROUP             FLAG   (KB)   (%)  DRIVE    MOUNT     CT
WG830B GROUP22  Y  Y  N  N     194044  65%  P9D3     --------    0
WG830A GROUP22  Y  Y  N  N     194656  65%           --------    0
MEDIA TYPE: 3995-1WO
3995 (650 MB) WORM optical disk media.
LIBRARY: PWA9
PSEUDO LIBRARY:
OWNER: K. G. SMITH
XCF MEMBER NAME: -N/A-
CREATION DATE: 2000-12-19  ENTER-EJECT DATE:  2000-12-19
VOLSER:            - WG830B -  - WG830A -
LAST WRITTEN DATE: 2001-03-08  2000-12-19
LAST MOUNTED DATE: 2001-03-08  2000-12-19
EXPIRATION DATE:   0001-01-01  0001-01-01
```

For an optical volume belonging to an Object Backup storage group, the command `D SMS,VOLUME(WG920A)` displays the following information:

```
CBR1140I OAM volume status:
VOLUME STORAGE  RD WR WP LOST  FREE SPACE    MOUNTED  PENDING   REQ
       GROUP             FLAG    (KB)   (%)  DRIVE    MOUNT     CT
WG920A BACKUP1  Y  Y  N  N     60838    20%           --------    0
WG920B BACKUP1  Y  Y  N  N    166181    55%           --------    0
MEDIA TYPE: 3995-1WO
3995 (650 MB) WORM optical disk media.
LIBRARY: PWA9
PSEUDO LIBRARY:
OWNER: K. G. SMITH
XCF MEMBER NAME: -N/A-
BACKUP TYPE: BACKUP1
CREATION DATE: 2000-12-07  ENTER-EJECT DATE:  2000-12-07
VOLSER:            - WG920A -  - WG920B -
LAST WRITTEN DATE: 2001-01-08  2000-12-07
LAST MOUNTED DATE: 2001-02-14  2000-12-07
EXPIRATION DATE:   9999-12-31  0001-01-01
```

Object Tape Volume Status

For an object tape volume, the following information is displayed:

```
CBR1240I  OAM object tape volume status
VOLUME  STORAGE RD  WR  CM  IN   MED  FREE-SPACE        %  FULL LOST  REQ
        GROUP               USE  TYPE                FULL     FLAG  CT

volser  sgname   a   b   c   d   ee   fffffffff{K|M}   gg   h   i     jjj
Volume is WORM tape.
Volume is logical WORM.
XCF MEMBER NAME: member-name
BACKUP TYPE: {BACKUP1 | BACKUP2}
CAPACITY:          capacity{K|M}  UNITNAME:          unitname
ERDS PHYSICAL ID:  epi
CREATION DATE:     createdate   EXPIRATION DATE:   expdate
LAST MOUNTED DATE: lmdate       LAST WRITTEN DATE: lwdate
DATACLASS:  dataclass           SUBLEVEL:          sublevel
```

The fields displayed in each data line are as follows:

volser
: Volume serial number of the requested tape volume.

sgname
: Name of the Object storage group or Object Backup storage group to which the tape volume belongs.

a
: Tape volume readability status:

  Y
  : Readable

  N
  : Unreadable

b
: Tape volume writability status:

  Y
  : Writable

  N
  : Unwritable

c
: Compaction indicator for this tape volume:

  Y
  : Tape volume written in compacted format.

  N
  : Tape volume written in noncompacted format.

d
: Tape volume in use indicator for this tape volume:

  Y
  : Tape volume currently in use by an OAM drive task.

  N
  : Tape volume not currently in use by an OAM drive task.

ee
: Media type of the requested tape volume:

  02
  : IBM Cartridge System Tape

  04
  : IBM Enhanced Capacity Cartridge System Tape

05
IBM High Performance Cartridge System Tape

06
IBM Extended High Performance Cartridge System Tape

07
IBM Enterprise Tape Cartridge

08
IBM Enterprise WORM Tape Cartridge

09
IBM Enterprise Economy Tape Cartridge

10
IBM Enterprise Economy WORM Tape Cartridge

12
IBM Enterprise Extended Tape Cartridge

14
IBM Enterprise Extended WORM Tape Cartridge

16
IBM Enterprise Advanced Tape Cartridge

17
IBM Enterprise Advanced WORM Tape Cartridge

18
IBM Enterprise Advanced Economy Tape Cartridge

***ffffffffff*{K|M}**
Remaining space on the requested tape volume in kilobytes (KB). If *ffffffffff* is followed by a 'K' then *ffffffffff* is in KB and the amount of KB is less than 2GB. If *ffffffffff* is followed by an 'M' then the freespace shown is in MB because the amount of KB is equal to or greater than 2GB.

gg
Percentage that the requested tape volume is full (percentage of the tape that has been used).

h
Volume full indicator:

Y
Volume is marked full.

N
Volume is not marked full.

P
Volume is marked permanently full.

When a volume is marked Y or N, OAM initialization will reevaluate this volume's full status based on the calculation of free space and percent valid. When a volume is marked P, it will remain P during the OAM initialization.

i
Volume lost indicator:

Y
Volume is marked lost.

N
Volume is not marked lost.

jjj
Number of read requests for this tape volume that are pending in OAM.

Volume is WORM tape
This text is displayed if the volume is physical WORM.

Volume is logical WORM
This text is displayed if the volume is logical WORM.

member-name
The XCF member name of the OAM which is controlling this tape volume, or -N/A-.

BACKUP1 | BACKUP2
The backup volume type for this volume. This volume contains first (BACKUP1) or second (BACKUP2) backup copies written to the Object Backup storage group to which this volume belongs.

***capacity*{K|M}**
Approximate number of millimeters of tape or approximate number of kilobytes of data which can be written to the volume, allowing variance for different manufacturers. If capacity is followed by a 'K' then capacity is in KB and the amount of KB is less than 2GB. If capacity is followed by an 'M' the capacity shown is in MB because the amount of KB is equal to or greater than 2GB.

unitname
MVS unit name used when the tape volume is allocated. If the tape volume is in an IBM tape library, this value is ignored.

epi
The ERDS Physical Identifier which indicates the real underlying device type that is used to write OAM objects to this volume. You can use this value to diagnose problems in a mixed device environment where native and emulated devices coexist.

createdate
Date the tape volume was created, in the format YYYY-MM-DD.

expdate
Expiration date of the tape volume, in the format YYYY-MM-DD.

lmdate
Date the tape volume was last mounted, in the format YYYY-MM-DD.

lwdate
Date the tape volume was last written to, in the format YYYY-MM-DD.

dataclass
This field represents the data class associated with this object tape volume. If no DATACLASS is specified, this field contains hyphens.

sublevel
This field indicates which tape sublevel this volume is associated with. Valid values are 1 or 2 for volumes associated with object storage groups, and --*N/A*-- for volumes associated with OAM SCRATCH or object backup storage groups.

For an object tape volume belonging to an Object storage group, the command F OAMx,D,VOL,J11981 or D SMS,VOLUME(J11981) displays the following information:

```
CBR1240I Object tape vol status: 135
 VOLUME STORAGE   RD WR CM IN  MED  FREE-SPACE  %   FULL LOST  REQ
        GROUP            USE TYPE           FULL      FLAG  CT
 J11981 GROUP01   Y  Y  N  N   07   292968448K   0   N    N      0
----------------------------------------------------------------------
Extended    LRG
Attributes: BLK
             N
----------------------------------------------------------------------
XCF MEMBER NAME: -N/A-
 CAPACITY:           292968448K   UNITNAME: 3590-1
 ERDS PHYSICAL ID:        0013
 CREATION DATE:     2008-03-20    EXPIRATION DATE:    0001-01-01
 LAST MOUNTED DATE: 2008-03-20    LAST WRITTEN DATE:  2008-03-20
 DATACLASS:         ATL51         SUBLEVEL:           1
 CBR1180I OAM tape volume status: 136
 VOLUME  MEDIA     STORAGE   LIBRARY   USE  W  C  SOFTWARE  LIBRARY
         TYPE      GROUP     NAME      ATR  P  P  ERR STAT  CATEGORY
 J11981  MEDIA5    SGATLOBJ  MTL13590  P    N  N  NOERROR   NONE
 -------------------------------------------------------------------
 RECORDING TECH:      EFMT1            COMPACTION:         NO
 SPECIAL ATTRIBUTE:  RDCOMPAT          ENTER/EJECT DATE:   2008-03-20
 CREATION DATE:      2007-11-30        EXPIRATION DATE:    2020-07-18
 LAST MOUNTED DATE:  2008-03-20        LAST WRITTEN DATE:  2008-03-20
 SHELF LOCATION:
 OWNER:
 -------------------------------------------------------------------
```

For an object tape volume belonging to an Object Backup storage group, the command F OAMx,D,VOL,E28212 or D SMS,VOLUME(E28212) displays the following information:

```
CBR1240I Object tape vol status: 097
VOLUME STORAGE   RD WR CM IN  MED  FREE-SPACE  %   FULL LOST  REQ
       GROUP            USE TYPE           FULL      FLAG  CT
E28212 IMAFIRST  Y  Y  N  N   04      873940K   0   N    N      0
----------------------------------------------------------------------
Extended    LRG
Attributes: BLK
             N
----------------------------------------------------------------------
XCF MEMBER NAME: -N/A-
BACKUP TYPE: BACKUP1
CAPACITY:              874218K   UNITNAME: 3490
ERDS PHYSICAL ID:        0002
CREATION DATE:     2008-01-12    EXPIRATION DATE:    2008-02-16
LAST MOUNTED DATE: 2008-01-16    LAST WRITTEN DATE:  2008-01-16
DATACLASS:         VTSM2CU       SUBLEVEL:           N/A
----------------------------------------------------------------------
CBR1180I OAM tape volume status: 098
VOLUME  MEDIA     STORAGE   LIBRARY   USE  W  C  SOFTWARE  LIBRARY
        TYPE      GROUP     NAME      ATR  P  P  ERR STAT  CATEGORY
E28212  MEDIA2    SGVTSOBJ  ATLBA999  P    N     NOERROR   PRIVATE
-------------------------------------------------------------------
RECORDING TECH:     36 TRACK         COMPACTION:         NO
SPECIAL ATTRIBUTE:  NONE             ENTER/EJECT DATE:   2006-06-26
CREATION DATE:      2006-06-26       EXPIRATION DATE:
LAST MOUNTED DATE:  2008-01-16       LAST WRITTEN DATE:  2008-01-16
SHELF LOCATION:
OWNER:
LM SG: SGVTSOBJ  LM SC: SCVTSOBJ  LM MC:           LM DC: VTSM2CU
-------------------------------------------------------------------
Logical volume.
```

If the object tape volume resides in an IBM automated tape library, you receive both the CBR1240I display for object tape volumes and the CBR1180I display for tape library volumes. The command F

OAMx,D,VOL,XCF011 or D SMS,VOLUME(XCF011) displays the following information for an OAM scratch tape volume that resides in an automated tape library:

```
CBR1180I OAM tape volume status:
VOLUME  MEDIA  STORAGE  LIBRARY  USE  W  C  SOFTWARE  LIBRARY
        TYPE   GROUP    NAME     ATR  P  P  ERR STAT  CATEGORY
XCF011  MEDIA2 *SCRTCH* ATLBA999 S    N  N  MEDIAMNT  NOTAVAIL
-------------------------------------------------------------
RECORDING TECH:    36 TRACK     COMPACTION:       UNKNOWN
SPECIAL ATTRIBUTE: NONE         ENTER/EJECT DATE: 2001-02-14
CREATION DATE: 2001-02-14       EXPIRATION DATE:
LAST MOUNTED DATE:              LAST WRITTEN DATE:
SHELF LOCATION:
OWNER:
```

Related reading: For more examples of CBR1180I, see the *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Displaying volumes that have LOSTFLAG set

The OAM display lost volumes command displays the volume serial numbers of the optical and tape volumes that have LOSTFLAG set. OAM requests might fail if this flag is set. This command allows you to determine which volumes are "lost" and to perform the necessary troubleshooting on those volumes.

The following command syntax displays the optical and tape volumes that have LOSTFLAG set:

```
>>─┬─ MODIFY ─┬─ OAM, ─┬─ DISPLAY, ─┬─ LOSTVOL ─><
   └─ F ──────┘        └─ D, ───────┘
```

The following is a description of the keyword for this command:

LOSTVOL
Displays the optical and tape volumes that have LOSTFLAG set.

To display the optical and tape volumes that have LOSTFLAG set, enter the following command:

```
F OAM,DISPLAY,LOSTVOL
```

The output of the F OAM,DISPLAY,LOSTVOL command shows optical volumes that are lost, but no tape volumes that are lost:

```
F OAM,DISPLAY,LOSTVOL
CBR1154I OPTICAL volume OPTVOL1 is a lost volume.
CBR1154I OPTICAL volume OPTVOL2 is a lost volume.
CBR1155I Total number of OPTICAL volumes marked lost is 2.
CBR1155I Total number of TAPE volumes marked lost is 0.
```

The output of the F OAM,DISPLAY,LOSTVOL command shows both optical and object tape volumes that are lost:

```
F OAM,DISPLAY,LOSTVOL
CBR1154I OPTICAL volume OPTVOL1 is a lost volume.
CBR1154I OPTICAL volume OPTVOL2 is a lost volume.
CBR1155I Total number of OPTICAL volumes marked lost is 2.
CBR1154I TAPE volume TAPEVOL1 is a lost volume.
CBR1155I Total number of TAPE volumes marked lost is 1.
```

To clear the status of a lost volume, use either of the following commands:

```
F OAM,UPDATE,VOLUME,VOLSER,LOSTFLAG,OFF
```

or the

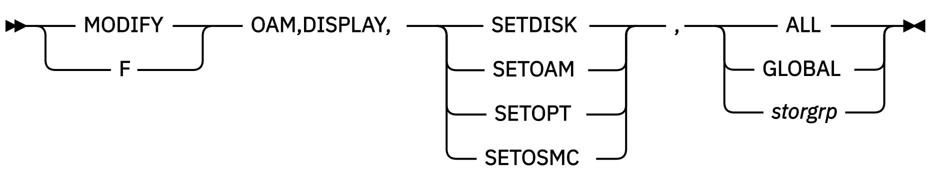
```
F OAM,RESTART
```

Displaying SETDISK, SETOAM, SETOPT, SETOSMC, and SETTLIB parameters

Use the F OAM,DISPLAY command to display the current settings of the SETDISK, SETOAM, SETOPT, or SETOSMC statement for the OAM address space.

For information on displaying the SETTLIB statement, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*

The following is the command syntax:



The following are the descriptions of the keywords used in this command:

SETDISK | SETOAM | SETOPT | SETOSMC

Specifies the statement whose settings are to be displayed. Use:

- SETDISK to display values of storage group level settings for the file system sublevel of the OAM storage hierarchy.
- SETOAM to display values of settings for the tape level of the OAM storage hierarchy.
- SETOPT to display general rules or options that span the disk, optical, and tape levels of the OAM storage hierarchy. In a multiple OAM configuration, optical-specific values will not be shown.
- SETOSMC to display the values of settings for OSMC processing.

ALL | GLOBAL | *storgrp*

Specifies the kind of information that OAM is to display.

For the SETDISK parameter, *storgrp* (where *storgrp* specifies a storage group name) and ALL are the only valid values.

For the SETOAM, SETOPT, and SETOSMC parameters, the valid values are:

ALL

Displays the settings for each valid storage group as well as the global default settings. If ALL is specified, the global default, if applicable, is displayed as well as the settings for each valid storage group in the active SMS configuration.

GLOBAL

Displays only the OAM global keywords.

storgrp

- Displays only the settings for the specified storage group name. Up to fifteen storage groups can be displayed with a single DISPLAY command, however the operating system command length limit might reduce that depending on the length of the storage group names.
- Displays the settings for the FIRSTBACKUPGROUP and SECONDBACKUPGROUP for the specified storage group name.

The following is a sample of F OAM,DISPLAY,SETDISK,*storgrp* command:

```
F OAM,DISPLAY,SETDISK,FSGR123

CBR1075I FSGR123 value for L2DIR is /oam/fsgr123/
CBR1075I FSGR123 value for L2TYPE is ZFS
```

The following is a sample of F OAM,DISPLAY,SETOAM,*storgrp* command:

```
F OAM,DISPLAY,SETOAM,GROUP22

CBR1075I GROUP22 value for SGMAXTPS is  55
```

```
CBR1075I GROUP22 value for SGMAXTPR is 22
CBR1075I GROUP22 value for EXPDATE is 2019/031
CBR1075I GROUP22 value for TFULLTHR is 240
CBR1075I GROUP22 value for TFULLPER is 100
CBR1075I GROUP22 value for TAPEUNIT is 3490
CBR1075I GROUP22 value for L2TAPEUN is 3490
CBR1075I GROUP22 value for DMWT is 120
CBR1075I GROUP22 value for DATACL is ATLM2CU
CBR1075I GROUP22 value for L2DATACL is VTSM2CU
CBR1075I GROUP22 value for TCOMP is N
CBR1075I GROUP22 value for TDRVSTRT is 9999
CBR1075I GROUP22 value for SGMAXREC is 1
```

The following is a sample of F OAM,DISPLAY,SETOAM,ALL command:

```
F OAM,DISPLAY,SETOAM,ALL

CBR1075I GLOBAL value for MAXTAPES is 68
CBR1075I GLOBAL value for MAXTAPER is 42
CBR1075I GLOBAL value for DSNSGNAM is Y
CBR1075I GLOBAL value for EXPDATE is 2011/115
CBR1075I GLOBAL value for TAPESDB is SMALL
CBR1075I GLOBAL value for TFULLTHR is 240
CBR1075I GLOBAL value for MWT is 5
CBR1075I GLOBAL value for DMWT is 7
CBR1075I GLOBAL value for DATACL is ATLM2CU
CBR1075I GLOBAL value for L2DATACL is ATLM2CU
CBR1075I GLOBAL value for TRECYC is GROUP
CBR1075I GLOBAL value for OSCRSYNC is DISABLED
CBR1075I GLOBAL value for TAPEDISP is 3
CBR1075I GLOBAL value for MAXRECYC is nn
CBR1075I GLOBAL value for PERCENTV is nnn
CBR1075I GLOBAL value for ALLOCRET is 5
CBR1075I BACKUP1 value for EXPDATE is /
CBR1075I BACKUP1 value for TFULLTHR is 0
CBR1075I BACKUP1 value for TFULLPER is 0
CBR1075I BACKUP1 value for TAPEUNIT is 3490
CBR1075I BACKUP1 value for L2TAPEUN is N/A
CBR1075I BACKUP1 value for DMWT is 300
CBR1075I BACKUP1 value for DATACL is ATLM2CU
CBR1075I BACKUP1 value for L2DATACL is N/A
CBR1075I BACKUP1 value for TCOMP is N
CBR1075I BACKUP1 value for TDRVSTRT is 0
CBR1075I BACKUP1 value for SGMAXREC is 1
CBR1075I BACKUP2 value for EXPDATE is /
CBR1075I BACKUP2 value for TAPEUNIT is 3490
CBR1075I BACKUP2 value for L2TAPEUN is N/A
CBR1075I BACKUP2 value for DMWT is 50
CBR1075I BACKUP2 value for DATACL is ATLM2CU
CBR1075I BACKUP2 value for L2DATACL is N/A
CBR1075I BACKUP2 value for TFULLTHR is 0
CBR1075I BACKUP2 value for TFULLPER is 0
```

The following is a sample F OAM,D,SETOAM,GLOBAL command:

```
F OAM,D,SETOAM,GLOBAL
CBR1075I GLOBAL VALUE FOR MAXTAPES IS 3
CBR1075I GLOBAL value for MAXTAPER is 3
CBR1075I GLOBAL value for DSNSGNAM is Y
CBR1075I GLOBAL value for EXPDATE is 2006/364
CBR1075I GLOBAL value for TAPESDB is SMALL
CBR1075I GLOBAL value for TFULLTHR is 9999
CBR1075I GLOBAL value for MWT is 3
CBR1075I GLOBAL value for DATACL is VTSM2CU
CBR1075I GLOBAL value for L2DATACL is DATACLASS
CBR1075I GLOBAL value for DMWT is 300
CBR1075I GLOBAL value for TRECYC is GROUP
CBR1075I GLOBAL value for OSCRSYNC is DISABLED
CBR1075I GLOBAL value for TAPEDISP is 3
CBR1075I GLOBAL value for MAXRECYC is 0
CBR1075I GLOBAL value for PERCENTV is 100
CBR1075I GLOBAL value for ALLOCRET is 5
```

The following is a sample of F OAM,D,SETOPT,*storgrp* command in a classic OAM configuration:

```
F OAM,DISPLAY,SETOPT,GROUP22

CBR1075I GROUP22 value for OPREINIT is GROUP
```

In a multiple OAM configuration, there are no storage group specific values for SETOPT so the F OAMC,D,SETOPT,storgrp command results in the following message: **CBR1089I There are no storage group level values to display.**

The following is a sample of F OAM,DISPLAY,SETOPT,ALL command in a classic OAM configuration:

```
F OAM,DISPLAY,SETOPT,ALL

CBR1075I GLOBAL value for OPREINIT is GROUP
CBR1075I GLOBAL value for OPDISDLY is 0
CBR1075I GLOBAL value for MWT is 3
CBR1075I GLOBAL value for ABUNREAD is INACTIVE
CBR1075I GLOBAL value for ABOFFLIN is BACKUP2
CBR1075I GLOBAL value for ABNOTOPE is INACTIVE
CBR1075I GLOBAL value for ABDB2ERR is INACTIVE
CBR1075I GLOBAL value for ABLOST is BACKUP1
CBR1075I GLOBAL value for ABFSERR is INACTIVE
CBR1075I GLOBAL value for MULTISYS is N
CBR1075I GROUP22 value for OPREINIT is GROUP
CBR1075I GROUP26 value for OPREINIT is GROUP
CBR1075I GROUP28 value for OPREINIT is GROUP
```

The following is a sample of F OAMC,DISPLAY,SETOPT,ALL command in a multiple OAM configuration:

```
F OAMC,DISPLAY,SETOPT,ALL

CBR1075I GLOBAL value for ABUNREAD is INACTIVE
CBR1075I GLOBAL value for ABOFFLIN is INACTIVE
CBR1075I GLOBAL value for ABNOTOPE is INACTIVE
CBR1075I GLOBAL value for ABDB2ERR is INACTIVE
CBR1075I GLOBAL value for ABLOST is INACTIVE
CBR1075I GLOBAL value for ABFSERR is INACTIVE
CBR1075I GLOBAL value for MULTISYS is N
```

The following is a sample F OAM,D,SETOPT,GLOBAL command in a classic OAM configuration:

```
F OAM,D,SETOPT,GLOBAL
CBR1075I GLOBAL value for OPREINIT is GROUP
CBR1075I GLOBAL value for OPDISDLY is 0
CBR1075I GLOBAL value for MWT is 5
CBR1075I GLOBAL value for UNLOADD is 1
CBR1075I GLOBAL value for UNLOADT is 9999
CBR1075I GLOBAL value for ABUNREAD is INACTIVE
CBR1075I GLOBAL value for ABOFFLIN is BACKUP2
CBR1075I GLOBAL value for ABNOTOPE is INACTIVE
CBR1075I GLOBAL value for ABDB2ERR is INACTIVE
CBR1075I GLOBAL value for ABLOST is BACKUP1
CBR1075I GLOBAL value for ABFSERR is INACTIVE
CBR1075I GLOBAL value for MULTISYS is N
```

The following is a sample F OAMC,D,SETOPT,GLOBAL command in a multiple OAM configuration:

```
F OAMC,D,SETOPT,GLOBAL
CBR1075I GLOBAL value for ABUNREAD is INACTIVE
CBR1075I GLOBAL value for ABOFFLIN is INACTIVE
CBR1075I GLOBAL value for ABNOTOPE is INACTIVE
CBR1075I GLOBAL value for ABDB2ERR is INACTIVE
CBR1075I GLOBAL value for ABLOST is INACTIVE
CBR1075I GLOBAL value for ABFSERR is INACTIVE
CBR1075I GLOBAL value for MULTISYS is N
```

The following is a sample F OAM,DISPLAY,SETOSMC,*storgrp* command in a classic OAM configuration:

```
F OAM,DISPLAY,SETOSMC,GROUP22
CBR1075I GROUP22 value for BACKUP1 is IMAFIRST
CBR1075I GROUP22 value for BACKUP2 is IMSECOND
CBR1075I GROUP22 value for RECALLO is 30,ON
CBR1075I GROUP22 value for RECALLT is 30,ON
CBR1075I GROUP22 value for RECALLF is OFF
CBR1075I GROUP22 value for RECALLD is 2
CBR1075I GROUP22 value for CLEAROLD is OPT
```

The following is a sample F OAMC,DISPLAY,SETOSMC,*storgrp* command in a multiple OAM configuration:

```
F OAMC,DISPLAY,SETOSMC,GROUP22

CBR1075I GLOBAL value for ABUNREAD is INACTIVE
CBR1075I GLOBAL value for ABOFFLIN is INACTIVE
CBR1075I GLOBAL value for ABNOTOPE is INACTIVE
CBR1075I GLOBAL value for ABDB2ERR is INACTIVE
CBR1075I GLOBAL value for ABLOST is INACTIVE
CBR1075I GLOBAL value for ABFSERR is INACTIVE
CBR1075I GLOBAL value for MULTISYS is N
```

The following is a sample F OAM,DISPLAY,SETOSMC,ALL command in a classic OAM configuration:

```
F OAM,DISPLAY,SETOSMC,ALL
CBR1075I GLOBAL value for BACKUP1 is IMAFIRST
CBR1075I GLOBAL value for BACKUP2 is IMSECOND
CBR1075I GLOBAL value for CYCLEW is STRTSTOP
CBR1075I GLOBAL value for MAXRECAL is 10
CBR1075I GLOBAL value for RECALLO is 15,ON
CBR1075I GLOBAL value for RECALLT is 15,ON
CBR1075I GLOBAL value for RECALLD is 1
CBR1075I GLOBAL value for BACKUPDE is OFF
CBR1075I GLOBAL value for CLEAROLD is OPT
CBR1075I GROUP00 value for BACKUP1 is IMAFIRST
CBR1075I GROUP00 value for BACKUP2 is IMSECOND
CBR1075I GROUP01 value for BACKUP1 is IMAFIRST
CBR1075I GROUP01 value for BACKUP2 is IMSECOND
CBR1075I GROUP02 value for BACKUP1 is IMAFIRST
CBR1075I GROUP02 value for BACKUP2 is IMSECOND
CBR1075I GROUP03 value for BACKUP1 is IMAFIRST
CBR1075I GROUP03 value for BACKUP2 is IMSECOND
CBR1075I GROUP04 value for BACKUP1 is IMAFIRST
CBR1075I GROUP04 value for BACKUP2 is IMSECOND
CBR1075I GROUP05 value for BACKUP1 is IMAFIRST
CBR1075I GROUP05 value for BACKUP2 is IMSECOND
CBR1075I GROUP06 value for BACKUP1 is IMAFIRST
CBR1075I GROUP06 value for BACKUP2 is IMSECOND
CBR1075I GROUP07 value for BACKUP1 is IMAFIRST
```

The following is a sample F OAMC,DISPLAY,SETOSMC,ALL command in a multiple OAM configuration:

```
F OAMC,DISPLAY,SETOSMC,ALL
CBR1075I GLOBAL value for ABUNREAD is INACTIVE
CBR1075I GLOBAL value for ABOFFLIN is INACTIVE
CBR1075I GLOBAL value for ABNOTOPE is INACTIVE
CBR1075I GLOBAL value for ABDB2ERR is INACTIVE
CBR1075I GLOBAL value for ABLOST is INACTIVE
CBR1075I GLOBAL value for ABFSERR is INACTIVE
CBR1075I GLOBAL value for MULTISYS is N
```

The following is a sample F OAM,D,SETOSMC,GLOBAL command in a classic OAM configuration:

```
F OAM,D,SETOSMC,GLOBAL
CBR1075I GLOBAL value for BACKUP1 is IMAFIRST1
CBR1075I GLOBAL value for BACKUP2 is SECOND1
CBR1075I GLOBAL value for CYCLEW is STRTONLY
CBR1075I GLOBAL value for MAXRECAL is 0
CBR1075I GLOBAL value for RECALLO is 1, ON
CBR1075I GLOBAL value for RECALLT is 1, ON
CBR1075I GLOBAL value for RECALLD is 1
CBR1075I GLOBAL value for CLEAROLD is BOTH
CBR1075I GLOBAL value for BACKUPDE is OFF
```

The following is a sample F OAMC,D,SETOSMC,GLOBAL command in a multiple OAM configuration:

```
F OAMC,D,SETOSMC,GLOBAL
CBR1075I GLOBAL value for ABUNREAD is INACTIVE
CBR1075I GLOBAL value for ABOFFLIN is INACTIVE
CBR1075I GLOBAL value for ABNOTOPE is INACTIVE
CBR1075I GLOBAL value for ABDB2ERR is INACTIVE
CBR1075I GLOBAL value for ABLOST is INACTIVE
CBR1075I GLOBAL value for ABFSERR is INACTIVE
CBR1075I GLOBAL value for MULTISYS is N
```

Displaying outstanding OAM messages

To display outstanding OAM messages, enter the following command:

```
DISPLAY R,L,KEY=OAM
```

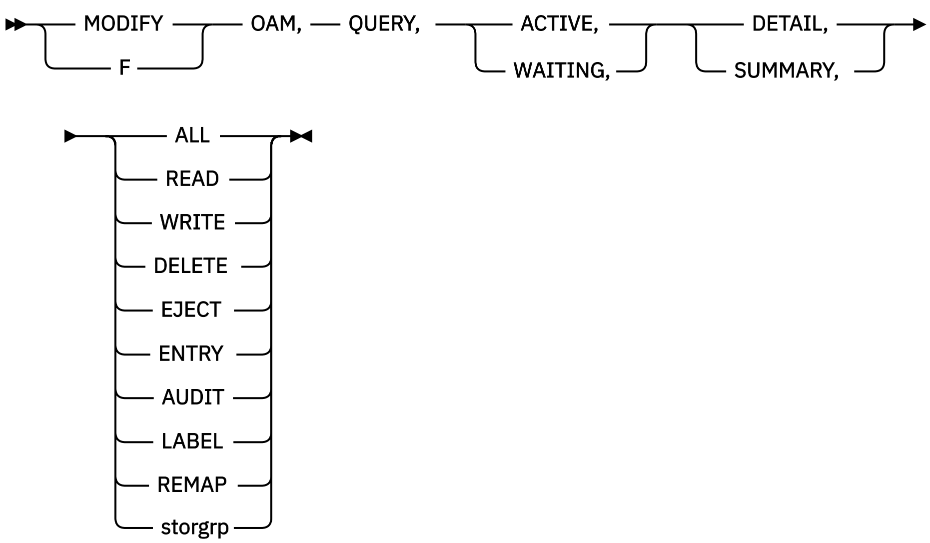
Use this command to display the message identification numbers and texts of all immediate action, eventual action messages, and messages waiting for replies that OAM issued.

Querying summary and detail information for pending and active requests

Use the F OAM,QUERY,*options* command to display:

- A summary of active file system requests
- A summary of active optical requests
- A summary of active tape requests
- A summary of waiting file system requests
- A summary of waiting optical requests
- A summary of waiting tape requests
- Detailed information concerning active file system requests
- Detailed information concerning active optical requests
- Detailed information concerning active tape requests
- Detailed information concerning waiting file system requests
- Detailed information concerning waiting optical requests
- Detailed information concerning waiting object tape requests
- A summary of active file system and tape requests for a storage group
- A summary of waiting file system and tape requests for a storage group
- Detailed information concerning active file system and tape requests for a storage group
- Detailed information concerning waiting file system and tape requests for a storage group

The following command syntax shows the QUERY command:



Recommendation: OAM is the default name of the cataloged procedure in your SYS1.PROCLIB. If a name other than OAM is used for the cataloged procedure, use that name in the QUERY statement. For example, MODIFY *procname_name*,QUERY,ACTIVE,SUMMARY.

The following are the keyword descriptions of the QUERY command:

QUERY
: Specifies a request to display information about active and waiting requests. The abbreviation for this command, Q, can also be used.

ACTIVE
: Indicates that only information about active requests, those currently being processed, is displayed. The abbreviation for this command, A, can also be used. Either the ACTIVE or WAITING keyword must be specified on the QUERY statement.

WAITING
: Indicates that only information about requests waiting to be processed are to be displayed. The abbreviation for this command, W, can also be used. Either the WAITING or ACTIVE keyword must be specified on the QUERY command. This information includes any waiting requests that have been sent to other instances of OAM in the OAMplex for processing.

SUMMARY
: Indicates that only summary information about the requested category is displayed. If neither the SUMMARY nor the DETAIL keyword is specified on the QUERY command, only summary information is displayed for the requested category. The abbreviation for this command, S, can also be used. This summary information also includes any waiting requests that have been sent to other instances of OAM in the OAMplex for processing. SUMMARY is the default. *storgrp* keyword can be specified after SUMMARY keyword:

  storgrp
  : Indicates summary information of all READS and WRITES (active or waiting) for the specified storage group.

DETAIL
: Indicates that only detailed information about the requested category (ACTIVE or WAITING) is displayed. If neither the DETAIL nor the SUMMARY keyword is specified on the QUERY command, only summary information is displayed for the requested category. The abbreviation for this command, D, can also be used. Query detail messages are written to the hard copy log so that the WTO buffers are not overrun, which causes a system degradation. One of the following keywords is required when the DETAIL keyword is specified:

  ALL
  : Indicates that detail information for either all active or all waiting (depending on the prior specification in the command) requests are to be displayed.

READ
Indicates that detail information for all READS (active or waiting) is displayed.

WRITE
Indicates that detail information for either all WRITES (active or waiting) is displayed.

DELETE
Indicates that detail information for all DELETES (active or waiting) is displayed.

EJECT
Indicates that detail information for all EJECTS (active or waiting) is displayed.

ENTRY
Indicates that detail information for all ENTRIES (active or waiting) is displayed.

AUDIT
Indicates that detail information for all AUDITS (active or waiting) is displayed.

LABEL
Indicates that detail information for all LABELS (active or waiting) is displayed.

REMAP
Indicates that detail information for all REMAPS (active or waiting) is displayed.

storgrp
Indicates detail information of all READS and WRITES (active or waiting) for the specified storage group.

Note: The OAM QUERY command is passed to the OAM address space through the MVS MODIFY system command. All messages that are sent as a result of the OAM QUERY command are sent to the system console from which the command originated, except for the QUERY DETAIL messages. They are sent to the system log only, not the console. The system operator must be aware that entering a F OAM,QUERY,ACTIVE,DETAIL or a F OAM,QUERY,WAITING,DETAIL command can result in a significant number of messages if there is a significant backlog of OAM requests processing or waiting for execution.

Related reading: For further information about messages that are associated with the OAM QUERY command, see *z/OS MVS System Messages, Vol 4 (CBD-DMO)*.

To query OAM to provide information about active file system, optical, and tape requests, enter the following command:

```
F OAM,QUERY,ACTIVE

       or

F OAM,QUERY,ACTIVE,SUMMARY
```

The following information is displayed only if optical libraries are defined in the active configuration:

```
CBR1720I OPTICAL ACTIVE SUMMARY
----------- OPTICAL REQUESTS CURRENTLY BEING PROCESSED--------------
 READS  WRITES  DELETES  ENTERS  EJECTS  AUDITS  LABELS
aaaaaa   bbbbbb  cccccc  dddddd  eeeeee  ffffff  gggggg
```

aaaaaa
Total number of object read requests from an optical volume currently being processed. This includes read requests being processed on this system that originated from another instance of OAM in an OAMplex.

bbbbbb
Total number of object write requests to an optical volume currently being processed. This includes write requests being processed on this system that originated from another instance of OAM in an OAMplex.

cccccc
Total number of object delete requests from an optical volume currently being processed.

dddddd
Total number of optical volume enter requests currently being processed.

eeeeee
Total number of optical volume eject requests currently being processed. This number also includes system initiated ejects.

ffffff
Total number of optical volume audit requests currently being processed.

gggggg
Total number of optical cartridge label requests currently being processed.

The following information is displayed only if a file system is defined in the active configuration:

```
CBR1745I FILE SYSTEM ACTIVE SUM:
---- FILE SYSTEM REQUESTS CURRENTLY BEING PROCESSED ----
READS WRITES
aaaaaa bbbbbb
```

aaaaaa
Total number of object read requests from a file system directory currently being processed.

bbbbbb
Total number of object write requests to a file system directory currently being processed.

The following information is displayed only if a file system is defined in the active configuration:

```
CBR1745I FILE SYSTEM ACTIVE SUM:
FOR STORAGE GROUP: ccccccc
---- FILE SYSTEM REQUESTS CURRENTLY BEING PROCESSED ----
READS WRITES
aaaaaa bbbbbb
```

aaaaaa
Total number of object read requests from a file system directory currently being processed.

bbbbbb
Total number of object write requests to a file system directory currently being processed.

ccccccc
The storage group name that this summary information is for.

The following information is displayed for object tape requests only if there are SETOAM statements in the current OAM invocation:

```
CBR1730I TAPE OBJECT ACTIVE SUM:
---- OBJECT TAPE REQUESTS CURRENTLY BEING PROCESSED--------------
 READS  WRITES
aaaaaa   bbbbbb
```

The following fields in the data line specify the number of each resource currently being processed:

aaaaaa
Total number of object read requests from a tape volume currently being processed. This includes read requests being processed on this system that originated from another instance of OAM in an OAMplex.

bbbbbb
Total number of object write requests to a tape volume currently being processed.

The following information is displayed for object tape requests only if there are SETOAM statements in the current OAM invocation and *storgrp* keyword is specified in the QUERY command:

```
CBR1730I TAPE OBJECT ACTIVE SUM:
FOR STORAGE GROUP: ccccccc
---- OBJECT TAPE REQUESTS CURRENTLY BEING PROCESSED--------------
 READS  WRITES
aaaaaa   bbbbbb
```

The following fields in the data line specify the number of each resource currently being processed:

aaaaaa
: Total number of object read requests from a tape volume currently being processed. This includes read requests being processed on this system that originated from another instance of OAM in an OAMplex.

bbbbbb
: Total number of object write requests to a tape volume currently being processed.

ccccccc
: The storage group name that this summary information is for.

To query OAM to provide information about waiting file system, optical, and tape requests, enter the following command:

```
F OAM,QUERY,WAITING

      OR

F OAM,QUERY,WAITING,SUMMARY
```

The following information is displayed only if optical libraries are defined in the active configuration:

```
CBR1700I OPTICAL WAITING SUMMARY:

------------- OPTICAL REQUESTS WAITING FOR PROCESSING---------------------
READS    WRITES    DELETES   ENTERS    EJECTS    AUDITS    LABELS
aaaaaa   bbbbbb    cccccc    dddddd    eeeeee   ffffff     gggggg
```

The following fields in the data line specify the number of each resource waiting for execution:

aaaaaa
: Total number of object read requests from an optical volume waiting to be processed. This includes read requests waiting to be processed on this system that originated from another instance of OAM in the OAMplex or read requests originated by this system, waiting to be processed by another instance of OAM in the OAMplex.

bbbbbb
: Total number of object write requests to an optical volume waiting to be processed. This includes write requests waiting to be processed on this system that originated from another instance of OAM in the OAMplex or write requests originated by this system, waiting to be processed by another instance of OAM in the OAMplex.

cccccc
: Total number of object delete requests from an optical volume waiting to be processed.

dddddd
: Total number of optical volume enter requests waiting to be processed.

eeeeee
: Total number of optical volume eject requests waiting to be processed. This number also includes system initiated ejected.

ffffff
: Total number of optical volume audit requests waiting to be processed.

gggggg
: Total number of optical cartridge label requests waiting to be processed.

The following information is also displayed for object tape requests only if there are SETOAM statements in the current OAM invocation:

```
CBR1710I TAPE OBJECT WAITING SUM:

----- OBJECT TAPE REQUESTS WAITING FOR PROCESSING----------
 READS  WRITES
aaaaaa   bbbbbb
```

The following fields in the data line specify the number of each resource waiting for execution:

aaaaaa
Total number of object read requests from a tape volume waiting to be processed. This includes read requests waiting to be processed on this system that originated from another instance of OAM in the OAMplex or read requests originated by this system, waiting to be processed by another instance of OAM in the OAMplex.

bbbbbb
Total number of object write requests to a tape volume waiting to be processed.

The following information is also displayed for object tape requests only if there are SETOAM statements in the current OAM invocation and *storgrp* keyword is specified in the QUERY command.

```
CBR1710I TAPE OBJECT WAITING SUM:
FOR STORAGE GROUP: ccccccc
----- OBJECT TAPE REQUESTS WAITING FOR PROCESSING----------
 READS  WRITES
aaaaaa   bbbbbb
```

The following fields in the data line specify the number of each resource waiting for execution:

aaaaaa
Total number of object read requests from a tape volume waiting to be processed. This includes read requests waiting to be processed on this system that originated from another instance of OAM in the OAMplex or read requests originated by this system, waiting to be processed by another instance of OAM in the OAMplex.

bbbbbb
Total number of object write requests to a tape volume waiting to be processed.

ccccccc
The storage group name that this summary information is for.

```
CBR1715I TAPE LIBRARY WAITING SUM:

----- TAPE LIBRARY REQUESTS WAITING FOR PROCESSING----------
 ENTERS  EJECTS  AUDITS
aaaaaa   bbbbbb   cccccc
```

The fields in the data line are defined as follows:

aaaaaa
Total number of tape volume entry requests waiting for processing. This is the total number of volumes recognized by OAM as being in the library manager insert category waiting to be processed. If OAM has not received the attention interrupt signaling the addition of cartridges to the insert category, the entered volumes will not be included in the summary count even though they have physically been entered into a library.

bbbbbb
Total number of user initiated tape volume eject requests waiting for processing in the OAM address space that have not yet been sent to the library manager.

cccccc
Total number of tape volume audit requests waiting for processing in the OAM address space that have not yet been sent to the library manager.

Note: All counts are a snapshot-in-time, so the counts can quickly change.

The following information is also displayed only if there is a file system defined in the current OAM invocation:

```
CBR1725I FILE SYSTEM WAITING SUM:

----- FILE SYSTEM REQUESTS WAITING FOR PROCESSING----------
 READS  WRITES
aaaaaa   bbbbbb
```

A display of file system work requests waiting for execution in the OAM address space is generated. The fields displayed in the data line of the multi-line message are as follows:

aaaaaa
: Total number of object read requests from a file system directory waiting to be processed.

bbbbbb
: Total number of object write requests to a file system directory waiting to be processed.

The following information is also displayed only if there is a file system defined in the current OAM invocation and *storgrp* keyword is specified:

```
CBR1725I FILE SYSTEM WAITING SUM:
FOR STORAGE GROUP: ccccccc
----- FILE SYSTEM REQUESTS WAITING FOR PROCESSING----------
 READS  WRITES
aaaaaa   bbbbbb
```

A display of file system work requests waiting for execution in the OAM address space is generated. The fields displayed in the data line of the multi-line message are as follows:

aaaaaa
: Total number of object read requests from a file system directory waiting to be processed.

bbbbbb
: Total number of object write requests to a file system directory waiting to be processed.

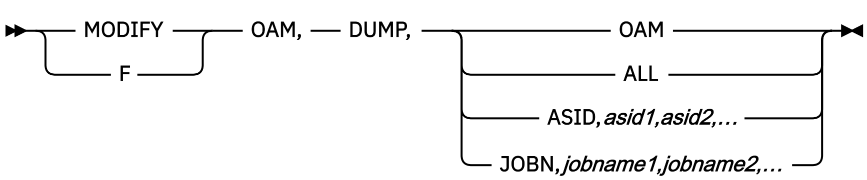
ccccccc
: The storage group name that this summary information is for.

Note: All counts are a snapshot-in-time, so the counts can quickly change.

Scheduling an SVC dump for the OAM address space

OAM uses SVC dumps as a diagnostic tool for system hangs or performance problems. In order to capture this data, the operator must issue the DUMP command after the completion of a recreate to obtain all the required data needed for diagnostics. OAM provides a streamlined version of the DUMP command. The F OAM,DUMP,(*operands*) command automatically collects all the pertinent data needed for diagnostic purposes without the operator having to key in all the correct parameters.

The following command syntax shows the DUMP command:



OAM
: Specifies an SVC dump is scheduled for the OAM address space. If the first operand after the DUMP verb is either OAM or blank, OAM schedules a SVC dump for the OAM address space.

ALL
: An SVC dump is scheduled for the OAM address space and any address spaces which have work queued to the OAM address space, up to 14 address spaces in addition to OAM.

  If the first operand after the DUMP verb is ALL, OAM scans all queues to identify address spaces that are not the OAM address space. OAM scans until all queues are searched or 14 address are found. OAM schedules an SVC dump for the OAM address space and up to 14 other address spaces that have work queued in the OAM address space.

ASID,*asid1, asid2, asid3...*
: An SVC dump is scheduled for the OAM address space and any address spaces separated by commas specified after the ASID operand. A valid ASID is a 1 to 4 hexadecimal (0–9, A–F) value. From one to 14 ASIDs can be specified with the ASID operand. If more than 14 ASIDs are specified, the first 14 will be used.

If the first operand after the DUMP verb is ASID, OAM validates that any ASIDs specified following the ASID operand are valid hexadecimal characters (0–9, A–F). If they are valid, OAM, schedules an SVC dump for the OAM address space and any additional address spaces specified (up to 14 address spaces in addition to OAM).

JOBN,*jobname1,jobname2,jobname3...*
An SVC dump is scheduled for the OAM address space and any job spaces specified after the JOBN operand separated by commas. A valid job name is a 1 to 8 character value of the following character set:

- Alphanumeric characters (A–Z, 0–9)
- National characters (&, $, @)
- Wildcard characters (*, ?) where '*' can stand for 0 or more characters, up to the maximum length of the job name string (8) and '?' can stand for one character.

From 1 to 14 job names can be specified with the JOBN operand. If more than 14 job names are specified, the first 14 will be used.

If the first operand after the DUMP verb is JOBN, OAM validates that any job names specified following the JOBN operand contain the valid character set. If they are valid, OAM schedules an SVC dump for the OAM address space and any job names specified (up to 14 jobs in addition to OAM).

OAM issues messages for any errors found in the DUMP command at SVC scheduling time and at SVC DUMP data capture completion.

Related reading: For more information concerning these messages, see *z/OS MVS System Messages, Vol 4 (CBD-DMO)*.

Restarting the OAM address space

Use the OAM RESTART command to restart the OAM address space. During restart processing, OAM matches the constructs and definitions that are used to those that are found in the active SMS configuration.

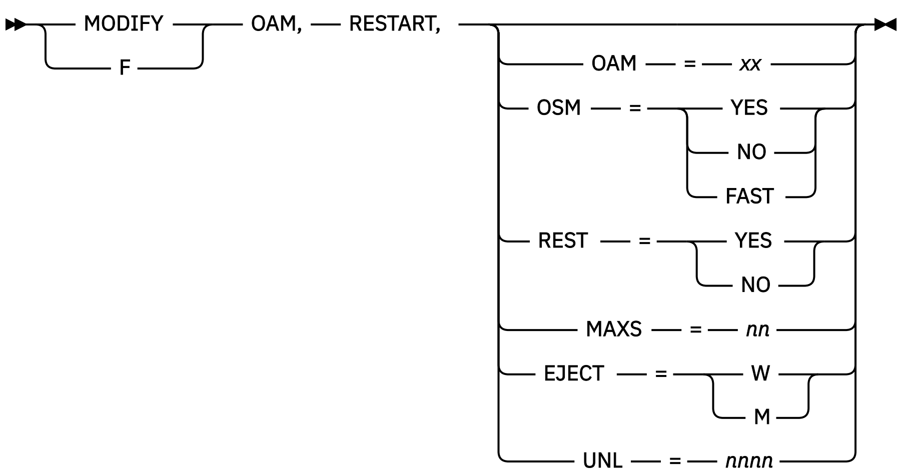
This command provides the ability to avoid having to do a STOP and START of the OAM address space and allows the OAM address space to retain its current ASID.

To restart the OAM address space without first stopping OAM, enter the following RESTART command:

```
F OAM,RESTART,parm=value,parm=value,...
```

You can issue this command when the parameter RESTART=NO is specified in the PARM string on the EXEC statement in the OAM procedure statement, which prevents OAM from automatically restarting the address space after a change is made to the SMS configuration. The time that OAM is notified of the SCDS activation depends on the time interval specified with the INTERVAL keyword in the IGDSMS*xx* PARMLIB member. If the change to the SMS configuration affects OAM and a restart of the OAM address space is required, you can issue this command in place of issuing a STOP and START command.

The following command syntax shows the RESTART command. Use the OAM parameter to specify which CBROAM*xx* PARMLIB member is used during initialization.



OAM

OAM=*xx* specifies the suffix of the CBROAM*xx* PARMLIB member that OAM should process during OAM address space initialization. The two alphanumeric characters (*xx*) must immediately follow the OAM=*keyword* in the PARM field. If the two characters immediately following the OAM=*keyword* are invalid or not specified, error message CBR0025I is issued. OAM only reads PARMLIB member CBROAM*xx* if the OAM=*keyword* is specified on the PARM field of the JCL EXEC statement in the OAM cataloged procedure. If no OAM=*keyword* is specified on the PARM field of the JCL EXEC statement, no PARMLIB member is read by OAM and object tape storage is not active. If the object tape storage is not active, OAM cannot read any objects back or write any new objects to tape until OAM is initialized with a valid OAM=*xx* specification, and a valid corresponding CBROAM*xx* PARMLIB member. OAM processes PARMLIB member CBROAM*xx* during OAM address space initialization.

OSM

Indicates whether or not OSMC is initialized at the same time as OAM after the RESTART command has been issued. The valid values for this parameter are as follows:

YES

OAM initializes with OSMC.

NO

OAM initializes without OSMC

FAST

OAM initializes with OSMC. However, OAM bypasses the collection audit, delaying it until OSMC storage group processing is done.

REST

Determines whether OAM should automatically restart when it receives notification that a new SCDS is activated. The valid values for this parameter are as follows:

YES

OAM automatically restarts when a new SCDS is activated.

NO

OAM does not automatically restart when a new SCDS is activated.

MAXS

MAXS=*nn* specifies the maximum number of OSMC storage management tasks that can be active at one time, where *nn* is a number value 1–99. The value given for MAXS must not exceed the number of optical and tape drives that is available for storage management processing. If the MAXS parameter is not specified, a default of 2 is assigned.

If concurrent processing includes Object storage groups writing to tape volumes, the correct corresponding (global level) MAXTAPERETRIEVETASKS and MAXTAPESTORETASKS values on the SETOAM statement must be specified.

EJECT
Used to determine which volumes are ejected from an optical library when the library is full and there is a request to add additional volumes to the library. The valid values for this parameter are as follows:

W
This value indicates that Library Space Management is to use the *least recently written* algorithm when selecting a volume to be ejected from an optical library.

M
This value indicates that Library Space Management is to use the *least recently mounted* algorithm when selecting a volume to be ejected from an optical library.

UNL
UNL=*nnnn* specifies the number of seconds of inactivity that OAM waits before unloading an optical drive in a library in order to keep at least one drive empty. This unload only occurs if there are no available optical drives within this library. That is, there are no empty online and operational drives. Thus, during periods of inactivity, you can cause at least one drive to be ready to accept the next mount request without first having to do a demount.

For more information about the MAXS parameter and related SETOAM keywords, see "OAM cataloged procedure parameter (MAXS)" on page 166.

Using the UPDATE command to set SETOAM, SETOSMC, and SETOPT values

You can change most of the current values of the SETOAM, SETOSMC, and SETOPT statements using the F OAM,UPDATE command without having to restart OAM. This command can also define settings of most of the SETOAM, SETOSMC, and SETOPT values if a CBROAM*xx* PARMLIB member was not used at OAM startup. This command provides dynamic processing that provides another method (other than using the CBROAM*xx* PARMLIB member) for customizing your object tape and optical storage.

If you used a CBROAM*xx* PARMLIB member at OAM initialization, you can use the OAM UPDATE command to change the current settings for SETOAM, SETOSMC, and SETOPT during the OAM session. If OAM is restarted, in most cases, the settings from the CBROAM*xx* PARMLIB member override any SETOAM, SETOSMC, or SETOPT values modified by an update command.

Exception: If you restart OAM and no specific statements exist in the CBROAM*xx* PARMLIB member for a storage group and a previous UPDATE SETOPT, SETOSMC, or SETOAM command was issued to update a field for that storage group, the value that is assigned by the update might be retained.

The following is the syntax for the F OAM,UPDATE,SETOAM | SETOPT | SETOSMC command:

```
>>-+- MODIFY -+- OAM,UPDATE, -+- SETOAM --+- ,scope,field,value,field,value -><
   +- F ------+               +- SETOPT --+
                              +- SETOSMC -+
```

Note: There is a maximum of seven pairs of field and value updates allowed on one UPDATE command.

The following are the descriptions of the keywords used in this command:

SETOPT | SETOAM | SETOSMC
Specifies the command parameter being updated. Use:

- SETOAM to update settings for the tape level of the OAM storage hierarchy.
- SETOPT to update general rules or options that span the disk, optical, and tape levels of the OAM storage hierarchy. The only settings that can be updated are specific to the optical level,and SETOPT is not supported on UPDATE in a multiple OAM configuration.
- SETOSMC to update settings for OSMC processing.This does not support optical-related keywords in a multiple OAM configuration.

Note: The values of the SETDISK statement, which specify the file system settings, are static values and cannot be changed with the UPDATE command.

scope
Specifies which storage groups will be affected by the update. Valid values are ALL or the name of a storage group that is defined in the active SMS configuration. If ALL is specified, the global default, if applicable, is updated for each object storage group in the active SMS configuration. If the name of a storage group is specified, only that storage group is updated with the setting value changes. Only one storage group can be specified with a single UPDATE command.

field
Specifies specific keyword in the SETOAM or SETOPT statement that is being updated.

value
Specifies the new value for the specified field. This value must conform to the same conditions and restrictions that apply to the CBROAM*xx* PARMLIB member in SYS1.PARMLIB, with the following exceptions:

- The OPREINIT (OPTICALREINITMODE) keyword values are GROUP and OAMSCR when this keyword is issued with the UPDATE command. This is due to an eight-character restriction on the input values.
- The TCOMP (TAPECOMPACTION / NOTAPECOMPACTION) keyword values are Y (indicating TAPECOMPACTION) or N (indicating NOTAPECOMPACTION) when this keyword is issued with the UPDATE command.
- The TRECYC (TAPERECYCLEMODE) keyword values are GROUP, OAMSCR, and MVSSCR when this keyword is issued with the update command. OAMSCR represents the OAMSCRATCH value. MVSSCR represents the MVSSCRATCH value.

Updating SETOAM values

Table 43 on page 367 lists the valid UPDATE command keywords and their associated SETOAM keywords. If a SETOAM keyword is not on the list, it is not modifiable on the F OAM,UPDATE command.

Table 43. Valid SETOAM keywords on the UPDATE command

UPDATE keyword	Associated SETOAM keyword	Associated value	Value range
ALLOCRET	ALLOCRETRYMINUTES	Minutes	0 - 5
DATACL	DATACLASS	Name	
DMWT	DEMOUNTWAITTIME	Seconds	1 - 9999
EXPDATE	TAPEEXPIRATION	YYYY/DDD	
L2DATACL	L2DATACLASS	Name	
L2TAPEUN	L2TAPEUNITNAME	Unit name	
MAXRECYC	MAXRECYCLETASKS	Number	0 - 15
MWT	MOUNTWAITTIME	Minutes	1 - 120
PERCENTV	PERCENTVALID	Percent	0 - 100
SGMAXREC	SGMAXRECYCLETASKS	Number	0 - 15
SGMAXTPR	SGMAXTAPERETRIEVETASKS	Number	1 - 100
SGMAXTPS	SGMAXTAPESTORETASKS	Number	1 - 100
TAPEDISP	TAPEDISPATCHERDELAY	Seconds	0 - 60
TAPEUNIT	TAPEUNITNAME	Unit name	
TCOMP	TAPECOMPACTION / NOTAPECOMPACTION	Y/N	

Table 43. Valid SETOAM keywords on the UPDATE command (continued)			
UPDATE keyword	**Associated SETOAM keyword**	**Associated value**	**Value range**
TDRVSTRT	TAPEDRIVESTARTUP	Megabytes	1 - 9999
TFULLPER	TAPEPERCENTFULL	Percent	1 - 100
TFULLTHR	TAPEFULLTHRESHOLD	Kilobytes	0 - 999999
TRECYC	TAPERECYCLEMODE	GROUP | OAMSCR | MVSSCR	

The following is an example of the syntax to update or set the number of recycle tasks that can be run concurrently:

F OAM,UPDATE,SETOAM,[ALL],MAXRECYC,*max_recyc_tasks*

- MAXRECYC represents the MAXRECYCLETASKS keyword of the SETOAM statement in the CBROAMxx PARMLIB member.
- *max_recyc_tasks* represents the maximum number of MOVEVOL tasks that can run concurrently as a result of a RECYCLE command. Valid values are 0 to 15.
- If a value of 0 is specified while there is an active START RECYCLE command processing, the update command fails.

If the execution of this command is successful, the system issues the following message:

```
CBR1074I Update successful for SETOAM parameter MAXRECYC, new
value max_recyc_tasks, scope (ALL). The previous value was old_value.
```

The following example is the syntax to update or set the number of recycle tasks that can be run concurrently for a specific storage group, or to set the number of recycle tasks that can be run concurrently for all storage groups:

F OAM,UPDATE,SETOAM,[ALL|*group-name*],SGMAXREC,*sgmax_recyc_tasks*

- SGMAXREC represents the SGMAXRECYCLETASKS keyword of the SETOAM statement in the CBROAMxx PARMLIB member.
- *sgmax_recyc_tasks* represents the maximum number of MOVEVOL tasks for a specific storage group that can run concurrently as a result of a RECYCLE command. Valid values are 0 to 15. If the SGMAXRECYCLETASKS associated with a given storage group exceeds the MAXRECYCLETASKS specified at the global level, OAM processes *only*concurrent MOVEVOL tasks for the storage group up to the MAXRECYCLETASKS value.
- If a value of 0 is specified while there is an active START RECYCLE command processing, the update command fails.

If the execution of this command is successful, the system issues the following message:

```
CBR1074I Update successful for SETOAM parameter SGMAXREC, new value
sgmax_recyc_tasks,
scope (ALL | group-name). The previous value was old_value..
```

The following example is the syntax to update or set the default percent valid threshold for RECYCLE:

```
F OAM,UPDATE,SETOAM,ALL,PERCENTV,percent_valid
```

- PERCENTV represents the PERCENTVALID keyword of the SETOAM statement in the CBROAMxx PARMLIB member
- *percent_valid* represents the threshold of valid data on a full tape volser to be considered a candidate for recycle selection. Valid values are 0 to 100.

If the execution of this command is successful, the system issues the following message:

```
CBR1074I Update successful for SETOAM parameter PERCENTV, new value
percent_valid, scope ALL. The previous value was previous_percent_valid.
```

The following example is the syntax to update the L2TAPEUNITNAME value:

```
F OAM,UPDATE,SETOAM,scope,L2TAPEUN,unitname
```

- L2TAPEUN represents the L2TAPEUNITNAME keyword of the SETOAM statement in the CBROAMxx PARMLIB member.
- *unitname* represents the unitname to be used for *scope*.

If the execution of this command is successful, the system issues the following message:

```
CBR1074I Update successful for SETOAM parameter L2TAPEUN, new value
unitname, scope scope. The previous value was old-unitname.
```

The following example is the syntax to update the L2DATACLASS value:

```
F OAM,UPDATE,SETOAM,scope,L2DATACL,dataclass
```

- L2DATACL represents the L2DATACLASS keyword of the SETOAM statement in the CBROAMxx PARMLIB member.
- *dataclass* represents the dataclass to be used for *scope*.

If the execution of this command is successful, the system issues the following message:

```
CBR1074I Update successful for SETOAM parameter L2DATACL, new value
dataclass, scope scope. The previous value was old-dataclass.
```

The following examples show some F OAM,UPDATE,SETOAM commands:

```
F OAM,UPDATE,SETOAM,GROUP22,TAPEUNIT,TAPEESO
```

Assuming TAPEESO is a valid tape esoteric name defined to the system, this command sets the TAPEUNITNAME setting for only GROUP22 to the tape esoteric, TAPEESO.

TAPEUNITNAME is the MVS unit name that OAM uses to initially allocate a scratch tape when an object is stored to this Object or Object Backup storage group and stored on a tape volume. If tape volumes already belong to the object or object backup storage group, they are used first before allocating a scratch volume. Even though a tape unit name is specified for the group, the ACS routines (for environment ALLOC) can override the TAPEUNITNAME specification by assigning the allocation to a Tape storage group, thereby, steering the allocation into an ATLDS or an MTL.

```
F OAM,UPDATE,SETOAM,ALL,TFULLPER,90,TFULLTHR,100
```

This command updates all global values and every storage group name in the active SMS configuration to a TAPEPERCENTFULL value of 90 and a TAPEFULLTHRESHOLD value of 100.

```
F OAM,UPDATE,SETOAM,GROUP22,MWT,2,DMWT,4,TFULLPER,95
```

This command updates the values for MOUNTWAITTIME, DEMOUNTWAITTIME, and TAPEPERCENTFULL for only the GROUP22 storage group.

```
F OAM,UPDATE,SETOAM,ALL,TRECYC,GROUP
```

This command updates the value for TAPERECYCLEMODE for expired object tape volumes in all Object and Object Backup storage groups. The expired tape volumes remain in their current storage group. When this command executes successfully, OAM issues the following message:

```
CBR1074I Update successful for SETOAM parameter TRECYC, new value GROUP,
scope ALL. The previous value was MVSSCR.
```

```
F OAM,UPDATE,SETOAM,ALL,TRECYC,MVSSCR
```

This command updates the value for TAPERECYCLEMODE for expired object tape volumes in all Object and Object Backup storage groups. The expired tape volumes are returned to the MVS scratch pool. When this command executes successfully, OAM issues the following message:

```
CBR1074I Update successful for SETOAM parameter TRECYC, new value MVSSCR,
scope ALL. The previous value was GROUP.
```

This command updates the value for TAPEDISPATCHERDELAY to 3 seconds for object tape volumes in all Object and Object Backup storage groups.

```
F OAM,UPDATE,SETOAM,ALL,TAPEDISP,3
```

This instructs OAM to wait 3 seconds before demounting a tape volume even if other work is available for this drive. When this command executes successfully, OAM issues the following message:

```
CBR1074I Update successful for SETOAM parameter TAPEDISP, new value 3,
scope ALL. The previous value was 0.
```

Updating SETOPT values

Table 44 on page 370 lists the valid UPDATE command keywords and their associated SETOPT keywords. If a SETOPT keyword is not on the list, it is not modifiable on the F OAM,UPDATE command.

Table 44. Valid SETOPT keywords on the UPDATE command

UPDATE keyword	Associated SETOPT keyword	Associated value	Value range
OPREINIT	OPTICALREINITMODE	GROUP | OAMSCR	
OPDISDLY	OPTICALDISPATCHERDELAY	Seconds	1 - 60
MWT	MOUNTWAITTIME	Minutes	1 - 9999
UNLOADD	UNLOADDRIVES	Number	1 - 6
UNLOADT	UNLOADTIMER	Seconds	1 - 9999

The following are some examples of F OAM,UPDATE,SETOPT commands:

```
F OAM,UPDATE,SETOPT,ALL,OPREINIT,OAMSCR,MWT,3,OPDISDLY,50
```

This command updates the values for OPTICALREINITMODE, MOUNTWAITTIME, and OPTICALDISPATCHERDELAY for all global values and for each storage group in the active SMS configuration.

```
F OAM,UPDATE,SETOPT,GROUP22,OPREINIT,GROUP
```

This command updates the values for OPTICALREINITMODE only for storage group GROUP22

```
F OAM,UPDATE,SETOPT,GROUP22,OPREINIT,GROUP,OPREINIT,OAMSCR
```

This updates the values for OPTICALREINITMODE twice for storage group GROUP22. The first update takes place followed by the second. The final result is OAMSCR.

```
F OAM,UPDATE,SETOPT,ALL,UNLOADD,4,UNLOADT,600
```

This updates the values of UNLOADDRIVES and UNLOADTIMER such that inactive optical drives are unloaded for all storage groups after 10 minutes idle time.

Updating SETOSMC values

Table 45 on page 371 lists the valid UPDATE command keywords and their associated SETOSMC keywords. If a SETOSMC keyword is not on the list, it is not modifiable on the F OAM,UPDATE command. The keywords in the table are specific to optical level and thus are not supported in a multiple OAM configuration.

Table 45. Valid SETOSMC keywords on the UPDATE command

UPDATE keyword	Associated SETOSMC keyword	Associated value	Value range
CLEAROLD	CLEAROLDLOC	OPT[1] | TAPE | BOTH[2]| NONE[2]	
CYCLEW	CYCLEWINDOW	STRTONLY | STRTSTOP	
MAXRECAL	MAXRECALLTASKS	Number	0 - 255
RECALLA[2]	RECALLALL	Days	0 - 255
RECALLD	RECALLDISKSUBLEVEL	Disk sublevel	1 - 2
RECALLF[2]	RECALLOFF	ON|OFF	
RECALLN[2]	RECALLNONE	ON	
RECALLO[1]	RECALLOPTICAL	Days	0 - 255
RECALLT	RECALLTAPE	Days	0 - 255

[1] Keyword or value relates only to the optical level and is not supported in a multiple OAM configuration

[2] Keyword or value relates to both the optical and tape levels. In a multiple OAM configuration, only the tape level setting will be updated.

The following is an example of the F OAM,UPDATE,SETOSMC command:

```
F OAMPLEX,UPDATE,SETOSMC,ALL,CYCLEW,STRTSTOP
```

This command updates the value for the SETOSMC CYCLEWINDOW keyword for all Object and Object Backup storage groups. STRTSTOP represents the STARTSTOP value, where the start and end cycle times represent a window in which to begin and end the OSMC cycle.

When this command executes successfully, OAM issues the following message:

```
CBR1074I Update successful for SETOSMC parameter CYCLEW, new value STRTSTOP,
scope ALL. The previous value was STRTONLY.
```

This command updates the value for CLEAROLDLOC specifications for all storage groups when objects transition to the DB2 sublevel or file system sublevel.

```
F OAM,UPDATE,SETOSMC,ALL,CLEAROLD,BOTH
```

This instructs OAM not to retain the original volume location information for objects on tape or optical media for ALL storage groups, when the objects have been transitioned to the DB2 sublevel or file system sublevel.

When this command executes successfully, OAM issues the following message:

```
CBR1074I Update successful for SETOSMC parameter CLEAROLD, new value BOTH,
scope ALL. The previous value was NONE.
```

This command updates the value for RECALLALL to 124 days for all storage groups.

```
F OAM,UPDATE,SETOSMC,ALL,RECALLA,124
```

This implicit recall parameter instructs OAM to copy to disk any object retrieved from an optical device or a tape device and keep it there for 124 days after the day of retrieval.

When this command executes successfully, OAM issues the following message:

```
CBR1074I Update successful for SETOSMC parameter RECALLA, new value RECALLO=124,ON
scope ALL. The previous value was RECALLO=0,OFF, RECALLT=0,OFF.
```

This command updates the value for RECALLOPTICAL for all storage groups.

```
F OAM,UPDATE,SETOSMC,ALL,RECALLO,10
```

This instructs OAM to copy to disk, and keep there for 10 days, any object retrieved from an optical device in any storage group.

When this command executes successfully, OAM issues the following message:

```
CBR1074I Update successful for SETOSMC parameter RECALLO, new value RECALLO=10,ON,
scope ALL. The previous value was RECALLO=233,ON.
```

This command updates the value for RECALLTAPE for storage group GROUP22.

```
F OAM,UPDATE,SETOSMC,GROUP22,RECALLT,10
```

This instructs OAM to copy to disk, and keep there for 10 days, any object retrieved from a tape device in storage group GROUP22.

When this command executes successfully, OAM issues the following message:

```
CBR1074I Update successful for SETOSMC parameter RECALLT, new value RECALLT=10,ON,
scope GROUP22. The previous value was RECALLT=0,OFF.
```

This command updates the value for RECALLNONE and activates it for storage group GROUP14.

```
F OAM,UPDATE,SETOSMC,GROUP14,RECALLN,ON
```

This implicit recall parameter instructs OAM not to copy to disk any GROUP14 object retrieved from an optical device or a tape device.

When this command executes successfully, OAM issues the following message:

```
CBR1074I Update successful for SETOSMC parameter RECALLN, new value RECALLO=3,OFF,
RECALLT=3,OFF, scope GROUP14. The previous value was RECALLO=3,ON, RECALLT=3,ON.
```

This command updates the value for RECALLOFF to ON for storage group GROUP24.

```
F OAM,UPDATE,SETOSMC,GROUP24,RECALLF,ON
```

This instructs OAM not to copy to disk any GROUP24 object retrieved from an optical device or a tape device, even if recall is specified on an OSREQ RETRIEVE request.

When this command executes successfully, OAM issues the following message:

```
CBR1074I Update successful for SETOSMC parameter RECALLF, new value ON, scope GROUP04.
The previous value was OFF.
```

This command updates the value for MAXRECALLTASKS to 8.

```
F OAM,UPDATE,SETOSMC,ALL,MAXRECAL,8
```

This instructs OAM process a maximum of 8 recalls to disk of tape or optical objects being retrieved from any storage group. This can only be issued at the global level.

When this command executes successfully, OAM issues the following message:

```
CBR1074I Update successful for SETOSMC parameter MAXRECAL,8
scope ALL. The previous value was MAXRECAL=27.
```

Using the UPDATE command to set OAMXCF values

Use the F OAM,UPDATE command to define or update the OAMXCF timeout parameters and value settings without restarting OAM. This command provides dynamic processing as an alternative to the CBROAM*xx* PARMLIB member for customizing your object tape and optical storage. OAMXCF keywords MEMBERNAME and GROUPNAME must first be specified in the CBROAM*xx* PARMLIB to establish an OAMplex.

The OAM UPDATE command can be used to change the current settings for OAMXCF timeout parameters during the OAM session. If OAM is restarted, the settings from the CBROAM*xx* PARMLIB member will not override any OAMXCF values that are changed by an update command.

The following is the syntax for the F OAM,UPDATE,OAMXCF command:

```
►►─┬─ MODIFY ─┬─ OAM,UPDATE,OAMXCF, parm,value,parm,value ─►◄
   └─── F ────┘
```

The following are the descriptions of the keywords used in this command:

OAMXCF
: Specifies the command parameter being updated. OAMXCF is used to update settings values in a Parallel Sysplex environment.

parm
: Specifies the timeout parameter setting that is being updated.

  The following are valid timeout parameters:

  OPTREADA
  : Indicates the number of seconds that an OAM originating an optical read request, which is shipped to another OAM within the OAMplex that owns the library where the object resides for processing, should wait for completion of the read request. This parameter equates to the XCFOPTREADA parameter of the OAMXCF statement in the CBROAM*xx* member of PARMLIB.

  OPTREADM
  : Indicates the number of seconds that an OAM originating an optical read request for a shelf-resident volume, which is shipped to another OAM within the OAMplex that owns the library where the object resides for processing, should wait for completion of the read request. This parameter equates to the XCFOPTREADM parameter of the OAMXCF statement in the CBROAM*xx* member of PARMLIB.

  OPTWRITA
  : Indicates the number of seconds that an OAM originating an optical write request that is targeted for an object storage group that contains real (automated) optical libraries, which are shipped to another OAM within the OAMplex that owns the optical library that is defined to the object storage group for processing, should wait for completion of the write request. This parameter equates to the XCFOPTWRITEA parameter of the OAMXCF statement in the CBROAM*xx* member of PARMLIB.

  OPTWRITM
  : Indicates the number of seconds that an OAM originating an optical write request that is targeted for an object storage group that contains pseudo libraries, which are shipped to another OAM

within the OAMplex that owns the pseudo library that is defined to the object storage group for processing, should wait for completion of the write request. This parameter equates to the XCFOPTWRITEM parameter of the OAMXCF statement in the CBROAM*xx* member of PARMLIB.

Note: This parameter is only valid in a classic OAM configuration.

TAPREADA
Indicates the number of seconds that an OAM originating a tape read request that is targeted for an Automated Tape Library Dataserver, which is shipped to another OAM within the OAMplex that owns the library in which the object exists for processing, should wait for completion of the read request. This parameter equates to the XCFTAPEREADA parameter of the OAMXCF statement in the CBROAM*xx* member of PARMLIB.

TAPREADM
Indicates the number of seconds that an OAM originating a tape read request that is targeted for a manual tape library, which is shipped to another OAM within the OAMplex that owns the library in which the object resides for processing, should wait for completion of the read request. This parameter equates to the XCFTAPEREADM parameter of the OAMXCF statement in the CBROAM*xx* member of PARMLIB.

value
Specifies the numeric value, in seconds, that is used for the specified timeout parameter.

Valid input values for the OAMXCF timeout parameters are numeric values in the range of 0–999999. A timeout value of zero causes the originating OAM to wait until a request is completed and a response is returned from the target OAM without ever timing out the request.

Updating fields in the DB2 Volume Table and the Tape Volume Table

Use the F OAM,UPDATE,VOLUME command to allow users to update specific fields in a volume record that are related to an optical or tape volume that is used for OAM object storage. You can use this command in place of the SPUFI option to make the changes.

Requirement: The volume requested for the update request must reside in a library that is either controlled by the OAM on which the command was entered or not controlled by any OAM at this time.

The following MVS command syntax updates valid fields in the DB2 Volume Table or the Tape Volume Table:

```
>>--+-MODIFY-+--OAM,UPDATE,--+-VOLUME-+--,volser,field,value,field,value-><
    '-F------'               '-VOL----'
```

The following are the descriptions of the keywords used in this command:

UPDATE
Specifies that an update is requested to a volume record related to an optical or tape volume used for object storage.

VOLUME
Indicates whether this update command is for an optical or tape volume.

volser
Indicates the specific volume serial number targeted for the update.

field
Specifies the specific field in the DB2 Volume Table or the Tape Volume Table that is targeted for the update.

value
Specifies the value to be assigned to the field as a result of the update.

Table 46 on page 375 shows the valid fields, field values, and value descriptions for updating optical volumes:

Table 46. Field values for optical volumes

Field	Valid values	Definition of value
LOSTFLAG	OFF	The volume lost flag will be reset. This is the only valid value for the LOSTFLAG operand.
EXPDATE	*yyyymmdd*	A valid date that specifies the scheduled expiration date for the volume. If all objects on this volume are not yet expired or deleted, OSMC automatically recalculates this date when the volume is selected for expiration, unless this volume belongs to the Object Backup storage group. This value might be recalculated if an object is stored to this volume prior to the expiration of the volume.
FULL	Y	Indicates that this volume is full, however, this volume's full status will be reevaluated during subsequent OAM initialization and might be marked not full (FULL='N').
	N	Indicates that this volume is not full, however, this volume’s full status will be reevaluated during subsequent OAM initialization and might be marked full (FULL='Y').
	P	Indicates that this volume has been marked permanently full by the `F OAM,UPDATE,VOLUME` command, so this volume’s full status will not be reevaluated during subsequent OAM initialization and is retained across OAM initialization.
READABLE	Y N	Indicates that this volume is readable. Indicates that this volume is not readable.
WRITABLE	Y N	Indicates that this volume can be written to. Indicates that this volume cannot be written to.
WRITPROT	Y N	Indicates that this volume is write protected. Indicates that this volume is not write protected.

Table 47 on page 375 shows the valid fields, field values, and value descriptions for updating object tape volumes:

Table 47. Field values for object tape volumes

Field	Valid values	Definition of value
LOSTFLAG	OFF	The volume lost flag will be reset. OFF is the only valid value for the LOSTFLAG operand.
EXPDATE	*yyyymmdd*	A valid date that specifies the scheduled expiration date for the volume. If all objects on this volume are not yet expired or deleted, OSMC automatically recalculates this date when the volume is selected for expiration, unless this volume belongs to the Object Backup storage group. This value might be recalculated if an object is stored to this volume prior to the expiration of the volume.

Table 47. Field values for object tape volumes (continued)

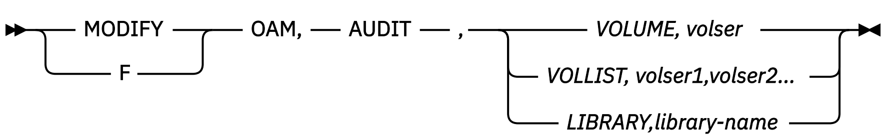
Field	Valid values	Definition of value
FULL	Y	Indicates that this volume is full.
	N	Indicates that this volume is not full, or that it is not writable. (See "SETOAM keyword definitions for STORAGEGROUP subparameters" on page 121 for a discussion of the TAPEPERCENTFULL parameter.) **Note:** OAM recalculates the PERCENTFULL and FULL status for volumes during initialization.
	P	Indicates that this volume has been marked permanently full by the F OAM,UPDATE,VOLUME command, so this volume's full status will not be reevaluated during subsequent OAM initialization and is retained across OAM initialization.
PFULL	0–100	Decimal value in the range of 0 to 100 that specifies the current percentage full for this object tape volume. This value might be recalculated by OAM initialization or after a write request to this volume.
READABLE	Y N	Indicates that this volume is readable. Indicates that this volume is not readable.
WRITABLE	Y N	Indicates that this volume is writable. Indicates that this volume is not writable.

Auditing a volume

OAM provides an AUDIT command that enables the system operator to audit a library resident tape or optical volume.

This command is valid only in a classic OAM configuration, or when directed to the Tape Library OAM address space in a multiple OAM configuration.

The syntax of the command for the AUDIT function is:



The following are the descriptions of the keywords used in this command:

VOLUME | VOLLIST | LIBRARY
Specifies the scope (single volume, volume list, or all volumes in a library) of the audit to be performed.

volser
Specifies the single tape or optical volume to be audited when the scope is VOLUME.

volser1, volser2...
Specifies up to 15 tape or optical volumes to be audited when the scope is VOLLIST.

library-name
Specifies the name of the tape or optical library to be audited when the scope is LIBRARY.

Remapping an optical library

OAM provides a REMAP command that enables the system operator to remap an optical library.

The syntax of the command for the REMAP function is:

```
►►─┬─ MODIFY ─┬── OAM, ── REMAP, ── library-name ─►◄
   └─ F ──────┘
```

library-name
Specifies the name of the optical library that is remapped.

Removing one or all OAM subsystems from the OAM configuration

Support is added to the MODIFY OTIS command to allow OAM subsystems to be removed from the current OAM configuration. This is intended to provide support for interchanging classic and multiple OAM configurations or removing an incorrectly defined OAM subsystem without requiring an IPL.

Use the MODIFY OTIS,DELSUB,*subsys* command to remove the specified subsystem (previously defined in the IEFSSNxx PARMLIB member or by a SETSSI ADD command) from the OAM configuration.

Use the MODIFY OTIS,DELSUB,ALL command to remove all subsystems that are currently in the OAM configuration leaving an OAM configuration with no subsystems.

MODIFY OTIS,DELSUB,[subsys|ALL]

subsys
Specifies the subsystem name of the OAM subsystem to be removed from the OAM configuration. Any OAM address space associated with the subsystem must be stopped prior to issuing this command. It will fail if an OAM address space is active for the subsystem.

ALL
Specifies that all OAM subsystems that are in the OAM configuration should be removed from it. All OAM address spaces associated with any OAM subsystem in the configuration must be stopped prior to issuing this command. If any OAM address spaces are active, the subsystem associated with each active address space will not be removed.

This command can be used to remove one or all OAM subsystems from the OAM configuration. The subsystem(s) will remain defined to z/OS but will no longer be a part of the OAM configuration and will no longer be used. This command can be used to remove from the OAM configuration OAM subsystems that were defined in the IEFSSNxx member of PARMLIB or with a SETSSI ADD command but are no longer needed. Once removed the subsystem(s) cannot be reconnected to the OAM configuration without an IPL. A new OAM subsystem (or multiple OAM subsystems in a multiple OAM configuration) can be defined and added to the OAM configuration using the SETSSI ADD command, but because the removed subsystem remains defined to z/OS, any newly added subsystem must use a different subsystem name.

Note: The OAM address space, if any, and all other activity (OSREQ applications) associated with the OAM subsystem to be deleted must be stopped prior to issuing this command. Any OAM subsystem with an active OAM address space will not be removed. Unpredictable results can occur if applications have an open OSREQ connection at the time a subsystem is removed.

Refreshing OTIS DB2 Tasks

OTIS maintains a separate subtask for each DB2 subsystem defined in the OAM configuration. If the connection to a particular DB2 subsystem cannot be established or terminates due to a condition other than DB2 not being started, the OTIS subtask for that DB2 subsystem terminates. Once the problem has been resolved, use the MODIFY OTIS,REFRESH command to re-establish a connection to that DB2 subsystem.

Note: Stopping and restarting OTIS with the STOP OTIS and START OTIS commands will refresh all DB2 subtasks but is not recommended in a multiple OAM configuration with more than one object instance because OTIS services will be unavailable to all instances during the time that OTIS is not active.

MODIFY OTIS,REFRESH

This command causes OTIS to check the status of its DB2 subtasks. For any DB2 subsystems defined in the OAM configuration for which it has no DB2 subtask, OTIS will attach a new DB2 subtask and attempt to connect to the DB2 subsystem.

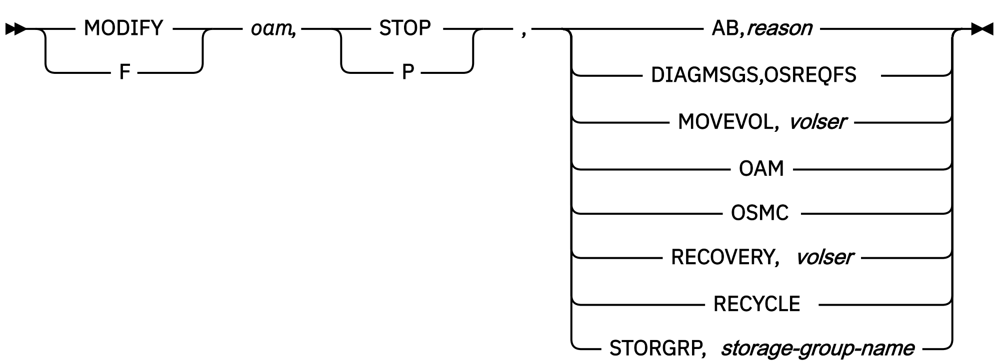
Note: This is only needed after a connection to a DB2 subsystem fails for a reason other than DB2 not being active. If a connection to DB2 fails only because DB2 was not active or was stopped, OTIS will detect when that DB2 is again available and automatically connect to it.

Stopping OAM functions

Use the OAM STOP command to stop the following processes:

- OAM
- OSMC
- OSMC processing for a particular storage group
- Move volume processing for a particular volume
- Volume recovery processing for a particular volume
- Access backup processing
- RECYCLE command processing.
- Display of diagnostic messages for file system related errors originating from an OSREQ request.

The following command syntax stops OAM:



Attention: To stop OAM, DB2 must be active. If you plan to stop DB2, stop OAM first.

The F *oam* , STOP command takes the following operands:

AB,*reason*
: Automatic access to backup copies of objects on removable media is discontinued for the specified reason.

DIAGMSGS,OSREQFS
: Stops the display of diagnostic messages for file system related errors originating from an OSREQ request.

MOVEVOL,*volser*
OSMC completes movement of objects that are currently in the process of being moved before stopping.

OAM
Stops OAM and stops OSMC if it is running. OSMC does not complete work in process before stopping.

OSMC
OSMC completes all work currently in process before stopping. OSMC continues processing DISPLAY commands.

RECOVERY,*volser*
OSMC completes the recovery of all objects currently in process before stopping the recovery.

RECYCLE
Stop the RECYCLE process.

Note: To stop a MOVEVOL of a specific volume that is currently being processed on behalf of a F OAM,START,RECYCLE command, you can issue the following command to terminate the MOVEVOL processing of the volume:

```
F OAM,STOP,MOVEVOL,volser
```

The processing of the F OAM,START,RECYCLE command continues.

STORGRP,*storage-group-name*
OSMC completes all work currently in process for this Object or Object Backup storage group before stopping.

Stopping OAM

To stop OAM, enter one of the following commands:

```
F OAM,STOP,OAM
or
STOP OAM
```

The system displays messages that indicates OAM termination status. If you stop OAM and the OSMC cycle is running, you receive the following messages:

```
CBR1000I OAM STOP command execution scheduled.

CBR0098I OAM termination starting.

CBR9011I OAM requested OSMC to terminate.

CBR9012I OSMC completed termination.

CBR0074I OAM XCF member xcf-member-name successfully left OAM XCF group
        xcf-group-name

CBR0099I OAM termination completed.
```

Note:

1. If OSMC is running and OSMC processing is still completing, you also receive other OSMC messages. OSMC does not complete work in process before stopping.
2. CBR0074I is issued only when OAM is stopped in a sysplex environment.

Stopping OSMC

To stop OSMC, enter the following command:

```
F OAM,STOP,OSMC
```

The system displays messages that indicates OSMC completion status. When OSMC is running, you also see OSMC messages as processing completes, in addition to the following messages:

```
CBR9047I Operator requested OSMC to stop processing.

CBR1000I OAM STOP command execution scheduled.

CBR9010I OSMC has stopped.
```

You can issue the following command to stop all OSMC processing:

```
F OAM,STOP,OSMC,FORCE
```

All OSMC processing is terminated immediately, so work that is currently processing is not completed. If FORCE is not specified, OSMC completes all work currently in process before stopping.

Stopping OSMC processing for a storage group

To stop OSMC for a storage group, enter the following command:

```
F OAM,STOP,STORGRP,storgrp
```

where *storgrp* is the name of the Object or Object Backup storage group.

The system displays messages that indicates OSMC completion status. When OSMC is running, you also see OSMC messages as processing completes, in addition to the following messages:

```
CBR1000I OAM STOP command execution scheduled.
 CBR9201I Object processing completed for storage group.
CBR9048I Storage group has successfully completed processing.
```

OSMC completes all work currently in process for this Object or Object Backup storage group before stopping.

Stopping the Move Volume utility

To stop the Move Volume utility for a volume, enter the following command:

```
F OAM,STOP,MOVEVOL,volser
```

where *volser* is the volume serial of the source volume from which objects are being moved.

The system issues the following messages:

```
CBR9856I Move Volume Utility stopping for volume volser.

CBR9858I Move Volume Utility status for volume volser. Total: total
         Attempted: attempted, Successful: successful,
         Unsuccessful:  unsuccessful.

CBR9859I Move Volume Utility ending for volume  volser.
```

The Move Volume utility completes any work it is currently processing, but does not move more objects than those already completed or those the utility is currently processing. If OAM is unable to process the STOP MOVEVOL command, it issues one of the following messages: CBR9093I, CBR9094I, or CBR9095I.

Stopping a volume recovery that is in progress

The system operator can stop a volume recovery that is in progress without also stopping OAM or OSMC processing. OAM allows any work (reads, writes, and updates) that is already scheduled to complete before ending the volume recovery.

Example: A volume recovery can run for hours for full volumes with small objects. This type of volume recovery can affect users trying to retrieve OAM data from the DB2 database. If a volume recovery is affecting users, you can stop the recovery in the morning and resume the recovery where it left off in the evening.

To stop a recovery that is already in progress, enter the following command:

1. Enter the following RECOVERY command:

   ```
   F OAM,STOP,RECOVERY,volser
   ```

   where *volser* is the volume serial number of the volume that is being recovered.

   ____

2. The system issues the following messages:

   ```
   CBR10001 OAM STOP command execution scheduled.
   CBR9862I Volume Recovery status for volumes volser1 and volser2 is not available.
   CBR9865I Volumes volser1 and volser2 will not be scheduled to be deleted
   because one or more objects could not be recovered.
   CBR9819I OAM Volume Recovery is ending for volumes volser1 and volser2.
   ```

   ____

Result: The volume recovery stops.

If a volume recovery requests mounts of volumes that an operator cannot find or an operator is unavailable to mount the volume, you can cancel the volume recovery until the volume or operator is available. You can resume a stopped volume recovery where it left off by issuing the `MODIFY OAM,START,RECOVERY,volser` command. Wait for the volume recovery to stop before starting the Volume Recovery utility again for the same volume.

If OAM is unable to stop the volume recovery, it issues one of the following messages: CBR9093I or CBR9095I. If the Volume Recovery utility is not active, OAM issues message CBR9094I.

In the following example, a volume recovery with the DELETE option was started. The operator replies with GO to proceed with the recovery, and then stops the volume recovery. Because the volume recovery is stopped before all the objects can be recovered, the optical volumes are not deleted.

```
F OAM,START,RECOVERY,PW801A,DELETE

CBR1000I OAM START command execution scheduled.
CBR9800I OAM Volume Recovery Delete starting for volumes PW801A and PW801B.
CBR9824I OAM Volume Recovery.
The following OPTICAL volumes are needed for recovery:
R8003A R8003B R8033A
*06 CBR9810D Reply 'QUIT' to terminate or 'GO' to
proceed with recovery.
6,go
IEE600I REPLY TO 06 IS;GO

F OAM,STOP,RECOVERY,PW801A

CBR10001 OAM STOP command execution scheduled.
CBR9862I Volume Recovery status for volumes PW801A and PW801B is not available.
CBR9865I Volumes PW801A and PW801B will not be scheduled to be deleted
because one or more objects could not be recovered.
CBR9819I OAM Volume Recovery is ending for volumes PW801A and PW801B.
```

Related reading: For more information about these system messages, see *z/OS MVS System Messages, Vol 4 (CBD-DMO)*.

Stopping automatic access to backup

To stop processing for automatic access to backup, enter the following command:

```
F OAM,STOP,AB,reason
```

The following are valid values and descriptions for the *reason* keywords:

ALL
: ALL is the default. Automatic access to backup processing is stopped for all object retrieves. When a retrieve for an object is attempted and the volume (optical or tape media) on which the object resides is not available for any of the following reasons, the retrieve request fails.

DB2ERROR
: Automatic access to backup processing is stopped for object retrieves from the 4 KB, 32 KB, or LOB object storage table. When a retrieve for an object is attempted and the object storage table on which the object resides is not available, the retrieve request fails.

FSERROR
: Automatic access to backup processing is stopped for object retrieves from the file system. When a retrieve for an object is attempted and the file system on which the object resides is not available, the retrieve request fails.

LOST
: Automatic access to backup processing is stopped for object retrieves from lost media as well as media that is not-defined. When a retrieve for an object is attempted and the volume (optical or tape media) on which the object resides is marked lost or is not-defined, the retrieve request fails.

NOTOPER
: Automatic access to backup processing is stopped for object retrieves from resident volumes (optical or tape media) in libraries that are not operation. When a retrieve for an object is attempted and the media on which the object resides is in a library that is marked nonoperational, the retrieve request fails.

OFFLINE
: Automatic access to backup processing is stopped for object retrieves from resident volumes (optical or tape media) in libraries that are not online. When a retrieve for an object is attempted and the volume on which the object resides is in a library that is offline or pending offline, the retrieve request fails.

UNREAD
: Automatic access to backup processing is stopped for object retrieves from unreadable media. When a retrieve for an object is attempted and the volume (optical or tape media) on which the object resides is marked not readable, the retrieve request fails.

The system displays the following messages that indicates access backup processing is stopped:

```
CBR1000I OAM STOP command execution scheduled.

CBR1091I OAM Access Backup processing stopped for reason.
```

If access to backup is already stopped, the following message displays:

```
CBR1093I OAM Access Backup processing already stopped for reason.
```

Where reason is one of the following values:

- DB2 OBJECT TABLE ERRORS
- FILE SYSTEM ERRORS
- LOST VOLUMES
- NOT OPERATIONAL LIBRARIES
- OFFLINE LIBRARIES
- UNREADABLE VOLUMES

Stopping recycle processing

To stop recycle processing, enter the following command:

```
F OAM,STOP,RECYCLE

CBR9856I Move Volume Utility stopping for volume CMW242.
CBR9858I Move Volume Utility status for volume CMW242. Total: 16828,
Attempted: 2864, Successful: 2864, Unsuccessful: 0.
CBR9859I Move Volume Utility ending for volumes CMW242 and N/A.
CBR9881I OAM START RECYCLE command ending unsuccessfully. Reason is
stopped by RECYCLE STOP command.
```

Stopping display of diagnostic messages for file system related errors originating from an OSREQ request

To stop the display of diagnostic messages for file system related errors originating from an OSREQ request.

```
F OAM,STOP,DIAGMSGS,OSREQFS
```

When diagnostic messages are stopped successfully for file system related errors originating from an OSREQ request. the following message is issued:

```
CBR1991I OAM diagnostic messages will no longer be issued for OSREQFS.
```

Message CBR1991I is also issued automatically by OAM whenever the current limit of messages to be issued is reached.

If diagnostic messages are already stopped for file system related errors originating from an OSREQ request, the following message is issued:

```
CBR1993I OAM diagnostic messages already stopped for OSREQFS.
```

Stopping OTIS

To stop OTIS, enter one of the following commands:

Note: Remember that if any OAM applications are processing, the OTIS address space must be active.

```
F OTIS,STOP
    or
STOP OTIS
```

When the OTIS address space has ended, the system issues the following message:

```
CBR8511I OTIS subsystem has terminated.
```

Appendix A. Sample optical hardware configurations

Note: If you are in a multiple OAM configuration and need to list or delete Optical libraries, the DB2SSID needs to be specified in the IGDSMSxx parmlib member in order for ISMF to connect to the correct DB2 subsystem and for the deletion of the optical library to be successful.

This appendix provides information on the following topics:

- "3995 Optical Library Dataserver" on page 385
- "Sample ISMF session for an IBM 3995 Optical Library Dataserver" on page 395
- "Defining optical drives" on page 406
- "Maintaining and modifying optical libraries and optical drives" on page 410

These topics discuss the IBM 3995 Optical Library Dataserver. This appendix provides the following information for each device:

- The major hardware components
- An overview of the physical connections between the various components
- Information important to the system software

3995 Optical Library Dataserver

There are various models of the IBM 3995 Optical Library Dataserver. In a nonsysplex environment where OAM is **not** running in an OAMplex with DB2 data sharing, each model can be connected to a single host processor operating in basic mode or to one logical partition of a processor complex operating in LPAR mode.

Attention: Multihost attachment is not supported and causes unpredictable results.

The 3995 models with two parallel channel adapters (PCA) are attachable to the host through parallel channels, while 3995 models with one or two ESCON channel adapters (ECA) are attachable to the host through ESCON channels.

Up to two ESCON directors can be connected between the 3995 Optical Library Dataserver and the host, but only one ESCON director can use dynamic link connection. The 3995 ESCON models can connect to the host in one of the following ways.

When using dynamic link, the host side port number of the dynamic link is required by the 3995 configuration program.

When using static link, or when not using an ESCON director, the host side port number is not required by the 3995 configuration program.

In an OAMplex with DB2 data sharing, it is recommended that you use ESCON and the System Automation for z/OS for connectivity as opposed to using parallel channels.

In a Parallel Sysplex, it is possible to establish a multisystem connection (logical not physical) that is controlled so that there is only one physical library-host connection at a time. Multiple instances of OAM within an OAMplex can be connected to a single library-host connection. The logical connection to these instances of OAM can be changed by specifying which library on a specific OAM system should be connected to the host system. This logical connection allows data to be accessed from and shared between various libraries associated with multiple instances of OAM within an OAMplex.

Related reading: See *IBM 3995 Optical Library Dataserver: Maintenance Information* for more information regarding supplying the port numbers to the 3995 configuration file.

Configurations for the 3995-1xx models

Figure 33 on page 386 shows a sample hardware configuration for the IBM 3995 Optical Library Dataserver with two parallel channel adapters.

Note: Although Figure 33 on page 386 shows a configuration with two parallel channel adapters, most 3995-133 and 3995-113 Optical Library Dataservers are ESCON attached.

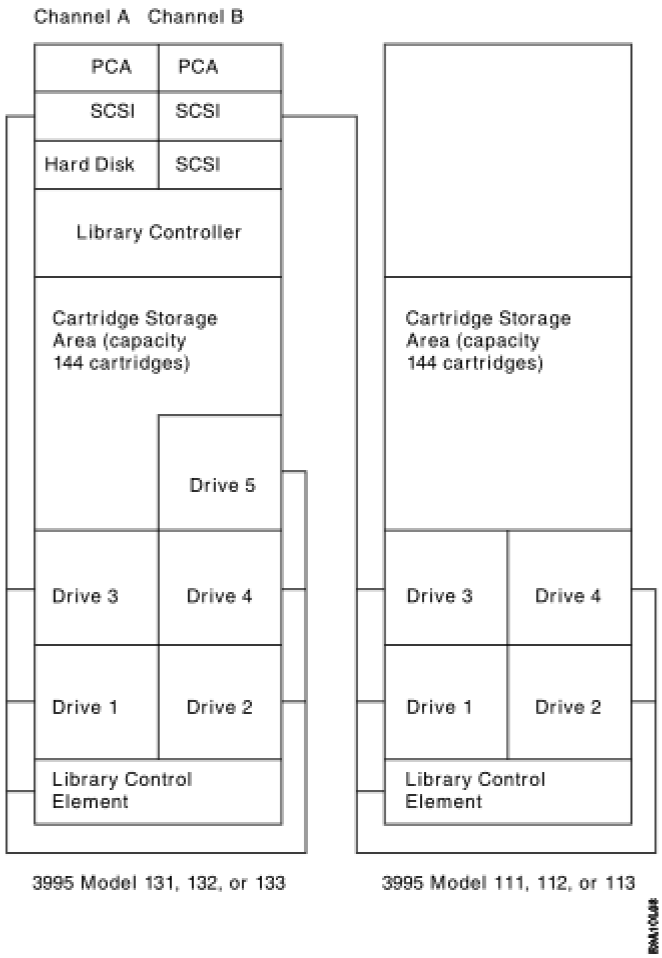


Figure 33. Sample Hardware Configuration—IBM 3995 Models 131, 132, 133, 111, 112, 113

Each channel adapter can be configured to run in one of the following modes, as shown in Table 48 on page 386 via the IBM 3995 Optical Dataserver's RAS package.

Table 48. PCA card speed setting to match CPU channel speeds.

CPU rated channel speed	Optical Library Controller PCA speed setting
High speed DC interlock	1.5 MB/second
2.0 MB/second data streaming	1.9 MB/second
3.0 MB/second data streaming	2.7 MB/second
3.5 MB/second data streaming	3.4 MB/second
4.5 MB/second data streaming	4.5 MB/second
ESCON channel speed	17.5 MB/second

Configurations for the 3995-Cxx models

Figure 34 on page 387, Figure 35 on page 388, Figure 36 on page 389, and Figure 37 on page 390 show sample hardware configurations for the 3995 C-Series Models. The C32, C34, C36, and C38 are attached to a single host processor through two ESCON channels or two parallel channels. While neither attachment supports multiple hosts, ESCON attachment does support remote connection of the 3995 up to 3 kilometers to the first director or host. For a configuration example for the 3995-11*x* and 3995-13*x* models see “Configurations for the 3995-1xx models” on page 386.

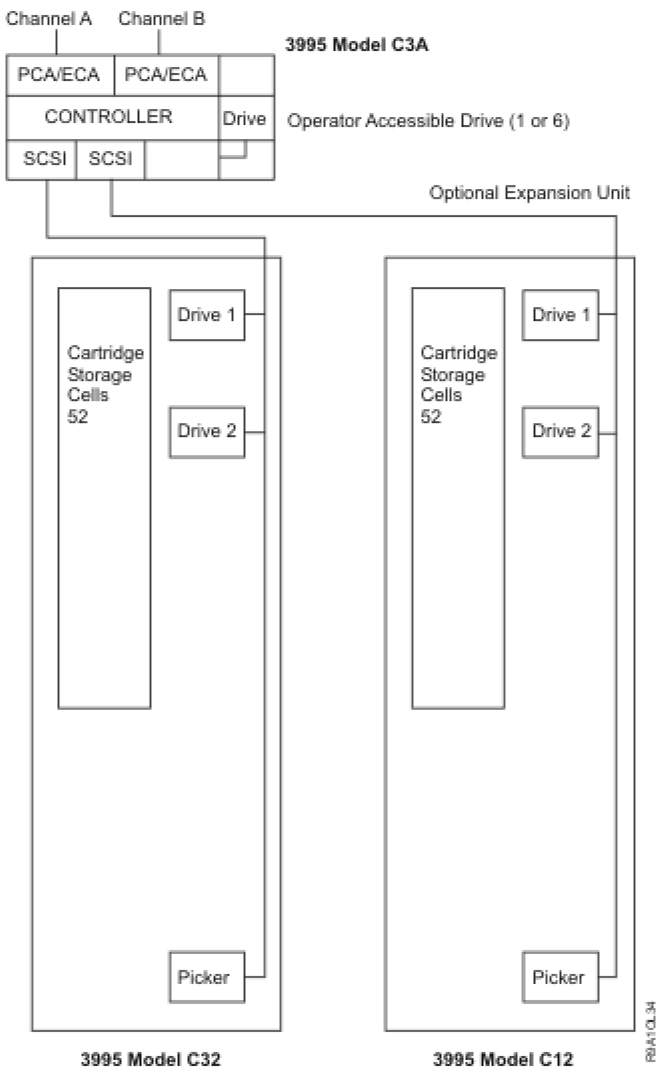


Figure 34. Sample Hardware Configuration—IBM 3995 Models C3A, C32, and C12

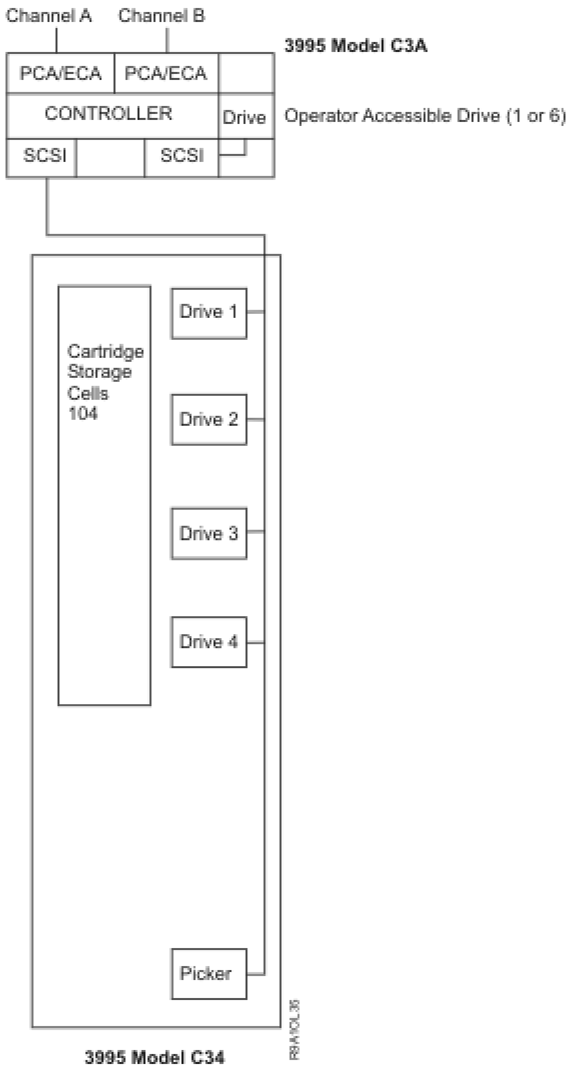


Figure 35. Sample Hardware Configuration—IBM 3995 Models C3A and C34

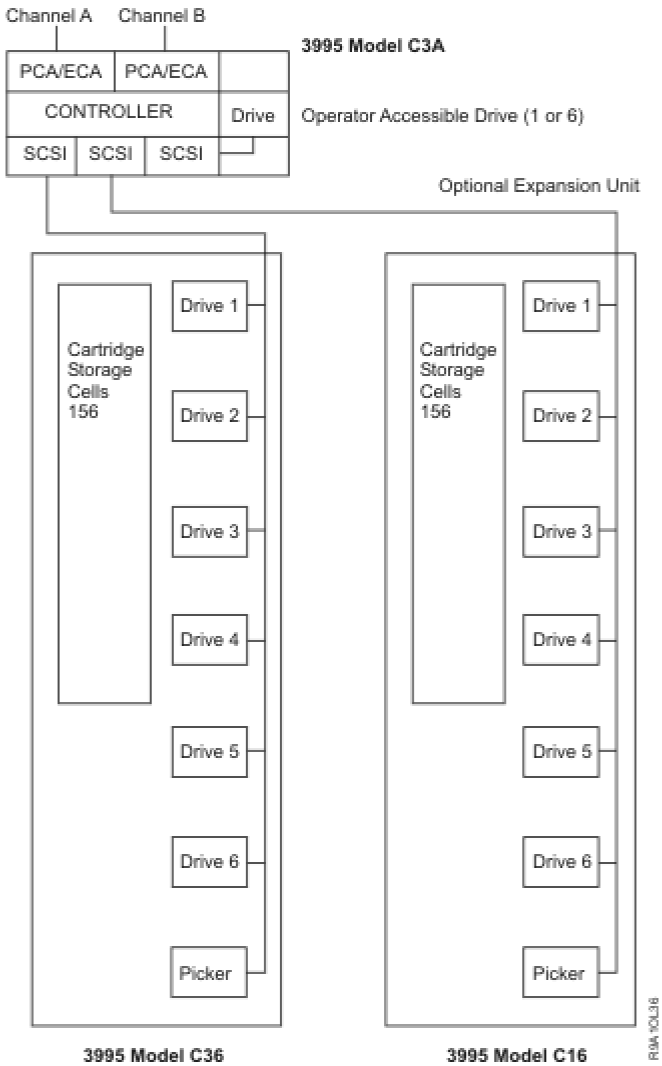


Figure 36. Sample Hardware Configuration—IBM 3995 Model C3A, C36, and C16

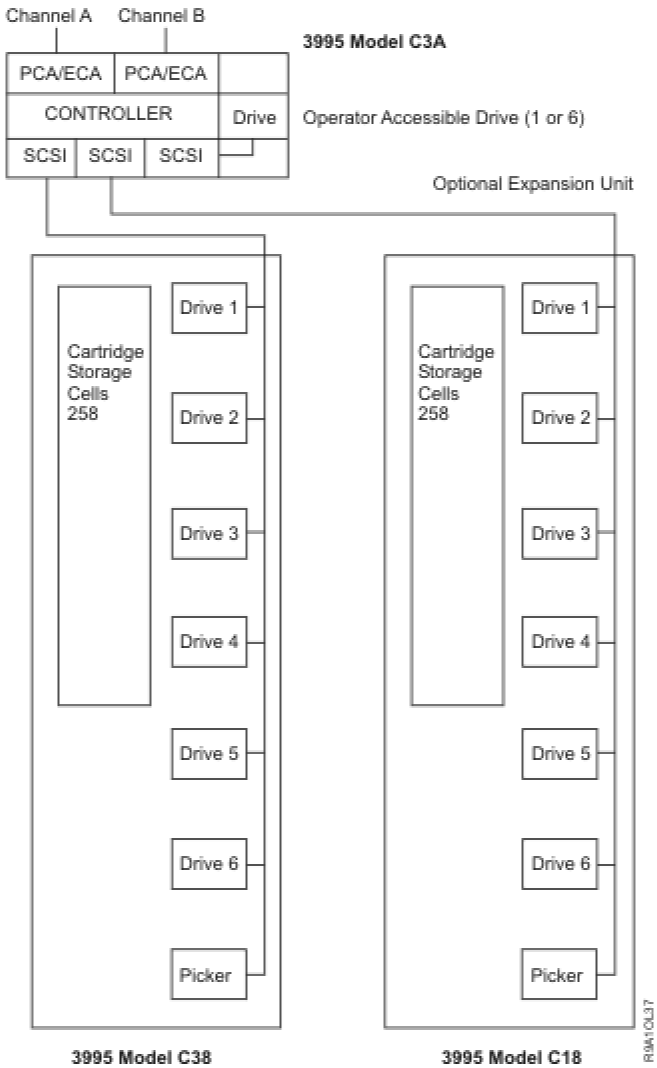


Figure 37. Sample Hardware Configuration—IBM 3995 Model C3A, C38 and C18

Defining 3995 device numbers

OAM uses specific base device numbers for communicating the following with the optical disk library and the optical disk drives within the library:

- Sending library related commands to the 3995 library
- Receiving unsolicited attention interrupts from the input output station
- Performing input and or output to specific optical disk drives

Each IBM 3995 library configuration is required to have device numbers defined to the MVS operating system and channel subsystem. A library configuration can consist of a controller, a single library, or a single library with an expansion unit. *The number of device numbers depends on the hardware configuration.*

Attention: Multihost attachment from two separate processors or two separate partitions of the same processor is not supported in a nonOAMplex environment. The multihost logical connection environment of an OAMplex uses only one active host-library connection.

The base device number for the IBM 3995-13*x* or 3995-C3A controlling library models must be a multiple of 16, meaning the low order digit of the base device number must be zero, for example, 0940.

The base device number for the IBM 3995-11*x* expansion unit models must be equal to the base device number of the controlling library (the IBM 3995-13*x* models) + 8, for example, 0948. The base device number for the expansion units (3995-11*x* models) is automatically calculated and does not need to be specified when defining these libraries.

The base device number for the IBM 3995-C*xx* models must match the base device number assigned to the device when it was installed. For further information on defining the 3995-C*xx* libraries, see Figure 43 on page 401.

Use the HCD panels to define the new 3995 device numbers. Defining 3995 device numbers with HCD provides the following capabilities:

- Dynamic I/O capability
- Specially designed 3995 Error Recovery Processing
- 3995 optical service information messages
- Ability to have MVS Unit Control Blocks for 3995 devices reside above the 16 MB line in 31-bit addressable storage
- Assigning the device type of 3995 for all 3995 device numbers

Using 3995-SDA definition in HCD

3995-SDA stands for a 3995 with self-description architecture. Self-description architecture allows the 3995 hardware to provide information concerning the optical library and optical drives to the OAM. This allows OAM to depend on information provided by the hardware without having to perform validity checking on its own. Although the 3995-13*x* models are not self-descriptive devices, the preferred method for defining all 3995 libraries is to use the 3995-SDA definition method.

To define the device numbers for a channel attached 3995 library, perform the following steps:

1. Use the Add Control Unit panel to define 3995-SDA type control units. Specify the base device number of each 3995 controller (3995-13*x* or 3995-C3A model) as a control unit. The base device number for 3995 library controller must be a multiple of 16, meaning the low order digit of this device number must end in zero, for example, 0940 or 0AC0. This is the control unit number that must be used when attaching the channel paths to the 3995. The unit addresses for the 3995 control unit must begin with 00 when attached to ESCON CHPIDs.
2. Use the Add Device panel to define the 3995. You must supply the starting device number; the low order digit of this device number must end in zero, for example 0940 or 0AC0. The device type must be a 3995-SDA. You can reserve device numbers for future drive expansions of the library. This allows you to best utilize channel address resources while allowing for the planning of future upgrades of the drives. The number of devices must be greater than zero and less than or equal to 256. The default is 16. The control unit number is the same as the base device number. Using the default number of devices as an example, devices 0AC0–0ACF are generated.

   Note: The 3995 does not belong to any esoteric device group and is not reserved through device allocation.

Using 3995 definition in HCD

This section describes defining device numbers for 3995-1*xx* models.

Note: The 3995-SDA definition method is the preferred method for defining *any* 3995 library. For information concerning this definition method, see "Using 3995-SDA definition in HCD" on page 391.

To define the device numbers for IBM 3995-1*xx* models, perform the following steps:

1. Use the Add Control Unit panel to define 3995 type control units. Specify the base device number of each 3995-13*x* model as a control unit. The base device number for the 3995-13*x* library must be a multiple of 16, meaning the low order digit of this device number must end in zero, for example, 0940 or 0AC0. This is the control unit number that must be used when attaching the channel paths to the 3995 Optical Dataserver. The unit addresses for the 3995 control unit must begin with 00 when attached to ESCON CHPIDs.
2. Use the Add Device panel to define the 3995-13*x* model. You must supply the base device number; the low order digit of this device number must end in zero, for example 0940 or 0AC0. The number of

devices must be 1 and the device type must be a 3995. The control unit number is the same as the base device number. Using 0AC0 as an example, devices 0AC0–0AC7 are generated.

3. Use the Add Device Panel to define the 3995-11*x* model. You must supply the base device number; the low order digit of this device number must be 8 (the base device number for the controlling library, IBM 3995-13*x* models + 8), for example, 0948 or 0AC8. The number of devices must be 1 and the device type must be 3995. The control unit number is the same as the base number of the 3995-13*x* model to which this 3995-11*x* model is attached. Using 0AC8 as an example, devices 0AC8–0ACF are generated, with 0AC0 as the control unit device number.

   Note: The 3995 does not belong to any esoteric device group and is not reserved through device allocation.

Migrating the configuration from IOCDS into IODF

The OAM optical device statements in the IOCP/MVSCP input data set must be upgraded for a migration to HCD/IODF for the following reasons:

- In contrast to IOCP, HCD validates device and control unit types, and associated parameters such as the protocols for a given control unit type and the connection of a device to a control unit.
- HCD introduced the specification of the SDA model for the 3995.
- HCD uses the true 3995 device type that had to be defined as a "look-like" device for MVSCP (3995 was defined as a "look-like" CTC, CTCS, or 3088 in MVSCP).

There are really two options that can be used to assist you in migrating to the HCD/IODF structure. The first method, and the preferred option, is to redefine your optical devices using HCD control unit and device definition panels. An alternate method is to change your MVSCP/IOCP statements and migrate them to IODF. Both methods are discussed in this section.

The *preferred* method of migrating your configuration from IOCDS to IODF is to perform the following steps:

1. Delete the CNTLUNIT and IODEVICE statements for the OAM 3995 optical devices from the old IOCP/MVSCP input data set.
2. Use the HCD migration panels or run a batch job to create a migrated IODF work file using the updated IOCP/MVSCP input from the previous step. This will create an IODF work file that does not include OAM 3995 optical devices.
3. Add the OAM 3995 optical devices to the IODF work file generated from the previous step by using the appropriate HCD panels. Use the HCD Control Unit Panel to add the OAM 3995 optical library controllers. Use the HCD I/O Device Panel to add the OAM 3995 optical devices.

An alternate method of migrating from MVSCP/IOCP to IODF is to change the device statements within the IOCP input data set. The HCD panels or a batch job can then be used to perform the migration.

Note: It is recommended that the *preferred* method be used instead of this alternate method. The changes made to the device and control unit statements in the alternate method are device specific and therefore might cause errors if not modified correctly.

The following examples are provided to assist you in changing your MVSCP/IOCP statements for 3995 optical devices for migration purposes only. These examples can be used when migrating an old MVSCP/IOCP input data set into an IODF file using the HCD migration panel or batch job. These statements should not be used as an input example for MVSCP or IOCP programs.

Here are some examples of changing the device and control unit statements for 3995-C*xx* optical devices:

```
* ============================================================ *
* A. 3995 Cxx Series                                           *
****************************************************************
****************************************************************
*                                                              *
* A1. ESCON 3995-C3x/C1x optical devices                       *
*     using ESCON (serial attached to Host)                    *
*                                                              *
*           (such as:  3995-C32/C12, 3995-C34/C16,             *
*                      3995-C38/C18)                           *
```

```
**************************************************************
*                                                            *
*  >>> ESCON 3995-SDA CHPID    statement               <<<   *
*        --->Note:                                           *
*              SWITCH=ss (ss: ESCON Director Number)         *
*              TYPE=CNC                                      *
*                                                            *
*              Also define the PARTITION                     *
*              Reconfigurable (REC) if running in LPAR mode  *
*                                                            *
**************************************************************
 CHPID PATH=((7B)),SWITCH=0A,TYPE=CNC,PARTITION=(PC6,REC)
 CHPID PATH=((7F)),SWITCH=0B,TYPE=CNC,PARTITION=(PC6,REC)
**************************************************************
*  >>> ESCON 3995-SDA CNTLUNIT statement               <<<   *
*        --->Note:                                           *
*              UNITADD=((00,nnn))  (nnn: # of devices        *
*                                    1 < nnn =< 256 )        *
*              UNIT=3995-SDA                                 *
*              LINK=(l1,l2)        (l1: ESCON Link Address)  *
*                                  (l2: ESCON Link Address)  *
**************************************************************
 CNTLUNIT CUNUMBR=0005,PATH=(7B,7F),UNITADD=((00,032)),        +
       UNIT=3995-SDA,LINK=(C4,C5)
**************************************************************
*  >>> ESCON 3995-SDA I/O device (IODEVICE) statement  <<<   *
*        --->Note:                                           *
*              ADDRESS=((xxx0,nnn))  (nnn: # of devices      *
*                                      1 < nnn =< 256 )      *
*              UNITADD=00                                    *
*              TIMEOUT=N                                     *
*              UNIT=3995                                     *
*              MODEL=SDA                                     *
*              DYNAMIC=YES                                   *
*              LOCANY=YES                                    *
**************************************************************
 IODEVICE ADDRESS=(19D0,032),UNITADD=00,CUNUMBR=(0005),        +
       UNIT=3995,MODEL=SDA,DYNAMIC=YES,LOCANY=YES
**************************************************************
**************************************************************
*                                                            *
* A2. OEMI 3995-C3x/C1x optical devices                      *
*     using OEMI (parallel attached to Host)                 *
**************************************************************
*                                                            *
*  >>> OEMI  3995-SDA CHPID    statement               <<<   *
*        --->Note:                                           *
*              TYPE=BL                                       *
*                                                            *
**************************************************************
 CHPID PATH=((9D)),TYPE=BL
 CHPID PATH=((9E)),TYPE=BL
**************************************************************
*  >>> OEMI 3995-SDA control unit (CNTLUNIT) statement <<<   *
*        --->Note:                                           *
*              UNITADD=((00,nnn))  (nnn: # of devices        *
*                                    1 < nnn =< 256 )        *
*              SHARED=N                                      *
*              PROTOCL=S4                                    *
*              UNIT=3995-SDA                                 *
**************************************************************
 CNTLUNIT CUNUMBR=0004,PATH=(9D,9E),UNITADD=((C0,016)),        +
       SHARED=N,PROTOCL=S4,UNIT=3995-SDA
**************************************************************
*  >>> OEMI 3995-SDA IO device (IODEVICE) statement    <<<   *
*        --->Note:                                           *
*              ADDRESS=((xxx0,nnn))  (nnn: # of devices      *
*                                      1 < nnn =< 256 )      *
*              TIMEOUT=N                                     *
*              UNIT=3995                                     *
*              MODEL=SDA                                     *
*              DYNAMIC=YES                                   *
*              LOCANY=YES                                    *
**************************************************************
 IODEVICE ADDRESS=(19C0,016),CUNUMBR=(0004),                   +
       TIMEOUT=N,UNIT=3995,MODEL=SDA,DYNAMIC=YES,LOCANY=YES
* ========================================================== *
```

Here are some examples of changing the device and control unit statements for 3995-1*xx* optical devices:

```
* ========================================================== *
*                                                            *
* B. 3995 1xx Series                                         *
. . .
**************************************************************
*                                                            *
* B1. ESCON 3995-13x/11x optical devices                     *
*     using ESCON (serial attached to Host)                  *
*                                                            *
*     (such as:  3995-133/113, 3995-132/112,                 *
*                3995-131/111.)                              *
*                                                            *
**************************************************************
*                                                            *
*  >>> ESCON 3995 channel path(CHPID) statement         <<<  *
*        --->Note:                                           *
*              SWITCH=ss  (ss: ESCON Director Number)        *
*              TYPE=CNC                                      *
*                                                            *
*              Also must define the PARTITION as             *
*              reconfigurable (REC) if running in LPAR mode  *
**************************************************************
 CHPID    PATH=((A6)),TYPE=CNC,PART=(PC6,REC),SWITCH=07
 CHPID    PATH=((A7)),TYPE=CNC,PART=(PC6,REC),SWITCH=08
**************************************************************
*                                                            *
*  >>> ESCON 3995 control unit(CNTLUNIT) statement      <<<  *
*        --->Note:                                           *
*              UNITADD=((00,016))                            *
*              UNIT=3995                                     *
*              LINK=(l1,l2)           (ESCON Link Addresses) *
*                                                            *
**************************************************************
 CNTLUNIT CUNUMBR=0164,PATH=(A6,A7),UNITADD=((00,016)),        +
       UNIT=3995,LINK=(ED,EF)
**************************************************************
*  >>> ESCON 3995-13x (A BOX) IODEVICE  statement:      <<<  *
*       (When the HCD migration completed successfully,      *
*        the following example statement will generate       *
*        8 device numbers for 3995 from 0E40 to 0E47)        *
*        --->Note:                                           *
*              ADDRESS=(xxx0,001)                            *
*              UNITADD=00                                    *
*              TIMEOUT=N                                     *
*              UNIT=3995                                     *
*              DYNAMIC=YES                                   *
*              LOCANY=YES                                    *
**************************************************************
 IODEVICE ADDRESS=(0E40,001),UNITADD=00,CUNUMBR=(0164),        +
       TIMEOUT=N,UNIT=3995,DYNAMIC=YES,LOCANY=YES
**************************************************************
*  >>> ESCON 3995-11x (B BOX) IODEVICE  statement:      <<<  *
*         (When the HCD migration completed successfully,    *
*          the following example statement will generate     *
*          8 device numbers for 3995 from 0E48 to 0E4f)      *
*          --->Note:                                         *
*              ADDRESS=(xxx8,001)                            *
*              UNITADD=08                                    *
*              TIMEOUT=N                                     *
*              UNIT=3995                                     *
*              DYNAMIC=YES                                   *
*              LOCANY=YES                                    *
*                                                            *
*              Also the controller unit number (CUNUMBR)     *
*              of A and B boxes must be the same.            *
**************************************************************
 IODEVICE ADDRESS=(0E48,001),UNITADD=08,CUNUMBR=(0164),        +
       TIMEOUT=N,UNIT=3995,DYNAMIC=YES,LOCANY=YES
**************************************************************
**************************************************************
* B2. OEMI 3995-13x/11x optical devices                      *
*     using OEMI (parallel attached to Host):                *
*     (such as:  3995-133/113, 3995-132/112,                 *
*                3995-131/111.)                              *
**************************************************************
*                                                            *
**************************************************************
*  >>> OEMI 3995 channel path (CHPID) statement        <<<   *
**************************************************************
```

```
 CHPID PATH=((1A)),TYPE=BL
 CHPID PATH=((9C)),TYPE=BL
**************************************************************
*  >>> OEMI 3995     control unit (CNTLUNIT) statement <<<   *
*        --->Note:                                           *
*              UNITADD=((x0,016)),                           *
*              SHARED=N                                      *
*              PROTOCL=S4                                    *
*              UNIT=3995                                     *
**************************************************************
 CNTLUNIT CUNUMBR=00C5,PATH=(9C,1A),UNITADD=((50,016)),        +
       SHARED=N,PROTOCL=S4,UNIT=3995
**************************************************************
*  >>> OEMI 3995-13x (A BOX) IODEVICE  statement:      <<<  *
*                                                            *
*       (When the HCD migration completed successfully,      *
*        the following example statement will generate       *
*        8 device numbers for 3995 from 0950 to 0957)        *
*        --->Note:                                           *
*              ADDRESS=(xxx0,001)                            *
*              UNITADD=x0                                    *
*              TIMEOUT=N                                     *
*              UNIT=3995                                     *
*              DYNAMIC=YES                                   *
*              LOCANY=YES                                    *
**************************************************************
 IODEVICE ADDRESS=(0950,001),CUNUMBR=(00C5),                   +
       TIMEOUT=N,UNIT=3995,DYNAMIC=YES,LOCANY=YES
**************************************************************
*                                                            *
*  >>> OEMI 3995-11x (B BOX) IODEVICE  statement:      <<<  *
*                                                            *
*         (When the HCD migration completed successfully,    *
*          the following example statement will generate     *
*          8 device numbers for 3995 from 0958 to 095f)      *
*          --->Note:                                         *
*              ADDRESS=(xxx8,001)                            *
*              UNITADD=x8                                    *
*              TIMEOUT=N                                     *
*              UNIT=3995                                     *
*              DYNAMIC=YES                                   *
*              LOCANY=YES                                    *
*                                                            *
*              Also the controller unit number (CUNUMBR)     *
*              of A and B boxes must be the same.            *
**************************************************************
 IODEVICE ADDRESS=(0958,001),CUNUMBR=(00C5),                   +
       TIMEOUT=N,UNIT=3995,DYNAMIC=YES,LOCANY=YES
* ========================================================== *
```

Sample ISMF session for an IBM 3995 Optical Library Dataserver

The following series of panels illustrates an ISMF session defining a sample 3995 library and drive configuration. These panels will illustrate how to define libraries and drives for 3995-1*xx*, 3995-C*xx*, and pseudo libraries. From the ISMF Primary Option menu for storage administrators (as shown in Figure 38 on page 396), select option 10 to start the library management dialog.

```
  Panel  Help
------------------------------------------------------------------------------
                           ISMF PRIMARY OPTION MENU - z/OS DFSMS V1R13
Enter Selection or Command ===>

Select one of the following options and press Enter:

 0  ISMF Profile              - Specify ISMF user profile
 1  Data Set                  - Perform Functions Against Data Sets
 2  Volume                    - Perform Functions Against Volumes
 3  Management Class          - Specify Data Set Backup and Migration Criteria
 4  Data Class                - Specify Data Set Allocation Parameters
 5  Storage Class             - Specify Data Set Performance and Availability
 6  Storage Group             - Specify Volume Names and Free Space Thresholds
 7  Automatic Class Selection - Specify ACS Routines and Test Criteria
 8  Control Data Set          - Specify System Names and Default Criteria
 9  Aggregate Group           - Specify Data Set Recovery Parameters
 10 Library Management        - Specify Library and Drive Configurations
 11 Enhanced ACS Management   - Perform Enhanced Test/Configuration Management
 C  Data Collection           - Process Data Collection Function
 L  List                      - Perform Functions Against Saved ISMF Lists
 P  Copy Pool                 - Specify Pool Storage Groups for Copies
 R  Removable Media Manager   - Perform Functions Against Removable Media
 X  Exit                      - Terminate ISMF

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

Figure 38. ISMF Primary Option Menu

When you select Library Management, the Library Management Selection menu is displayed, as shown in Figure 39 on page 396. At this point, you select either an optical library configuration or an optical drive configuration. Because a library must be defined before any drives associated with that library can be defined, you must first define libraries and then define drives. Select option 1 to display the Optical Library Application Selection panel, as shown in Figure 40 on page 397.

```
  Panel  Help
------------------------------------------------------------------------------
                      LIBRARY MANAGEMENT SELECTION MENU
Enter Selection or Command ===>_

 1  Optical Library          - Optical Library Configuration
 2  Optical Drive            - Optical Drive Configuration
 3  Tape Library             - Tape Library Configuration

Use HELP Command for Help; Use END Command to Exit.
```

Figure 39. Library Management Selection Menu

The ISMF screens that follow provide examples for defining 3995-11*x*, 3995-13*x*, 3995-C*xx*, and pseudo libraries. These examples define the pseudo libraries based on the media type and device type association; however, you can define pseudo libraries to best suit the requirements of your environment (based, for example, on location, or all backup objects with primary active objects, or data categories, and so forth). Where applicable, the ISMF screens are duplicated with examples for the different library definitions. The following libraries are defined in our sample configuration:

- LIBRARY1—3995-133

- LIB1C—3995-C3A
- LIBRARY2—3995-113
- LIBRARYA—3995-C38
- LIBRARYB—3995-C18
- PSEUD01—operator-accessible drives.

The name of the source control data set (SCDS) that contains the library and drive definitions in these examples is SCDS.PRIMARY. The default CDS NAME is the single-quoted word 'ACTIVE', which represents the currently active configuration. Whenever you define a new optical library within an SCDS, the library definition added to the SCDS is identified by the system to which the storage administrator is logged on. You can use the Optical Library Configuration application to add an optical library to an SCDS. The Optical Library Configuration application must be running on the system on which the OAM configuration database resides. The OAM configuration database cannot be shared among systems.

The OPTICAL LIBRARY APPLICATION SELECTION panel, Figure 40 on page 397, provides an example of selecting the define option to define a 3995-133 optical library (LIBRARY1). The following information must be provided on this screen:

- CDS Name—'SCDS.PRIMARY'
- Library Name—LIBRARY1
- Library Device Number—3995-133
- Library Type—REAL
- Select Option 3 (Define) to continue the definition

```
   Panel  Utilities  Help
 ------------------------------------------------------------------------------
                     OPTICAL LIBRARY APPLICATION SELECTION
Command ===>_

To Perform Library Operations, Specify:
  CDS Name  . . . . . . . 'SCDS.PRIMARY'
                                 (1 to 44 character data set name or 'ACTIVE')
  Library Name  . . . . . LIBRARY1  (For Optical Library List, fully or
                                     Partially Specified or * for All)
  Library Device Type . . 3995-133  (For Optical Library List, fully or
                                     Partially Specified or * for All)
  Library Type . . . . .  REAL  (REAL, PSEUDO, or * for ALL)

Select one of the following Options:
  3  1. List    - Generate a list of Libraries
     2. Display - Display a Library
     3. Define  - Define a Library
     4. Alter   - Alter a Library
If List Option is Chosen,
   Enter "/" to select option    _ Respecify View Criteria
                                 _ Respecify Sort Criteria
 Use ENTER to Perform Selection;
 Use HELP Command for Help; Use END Command to Exit.
```

Figure 40. Optical Library Application Selection Panel

The following are output fields specified in the Optical Library Application Selection panel:

SCDS NAME
The name of the source control data set that this library is defined in.

LIBRARY NAME
The name of the library being defined.

Rules:

1. To avoid confusion, do not assign the same name to both libraries and Object storage groups when defining them to the OAM configuration database.

2. You must include library names in the Object and Object Backup storage group definition so that optical writes are performed for the Object or Object Backup storage group.

LIBRARY TYPE

The type of library you are addressing. This can be either REAL or PSEUDO. A REAL optical library is a physical library containing optical disk drives and optical volumes that reside physically inside the library. A PSEUDO optical library is a set of shelf-resident optical disk volumes associated with operator-accessible optical disk drives, or both.

In order to process read and write requests from optical shelf-resident volumes, you need to define compatible optical operator-accessible drives to the pseudo optical library when defining it to the OAM configuration database. If you do not define any optical operator-accessible drives to the pseudo optical library, OAM cannot process read and write requests for optical shelf-resident volumes.

OAM stills uses pseudo optical libraries for each optical drive device type. If no pseudo optical library is defined in the active SMS configuration for a valid optical drive device type, OAM defines a default using the following names that can be used for assigning shelf-resident optical volumes:

- PCTREUSE—3995-131 device types
- PCTWORM—3995-132 device types
- P3995133—3995-133 device types
- P3995SW3—3995-SW3 device types
- P3994SW4—3995-SW4 device types

Note: After the first library definition, ISMF primes CDS NAME, LIBRARY NAME, LIBRARY DEVICE TYPE, and LIBRARY TYPE with the last used reference values on the Optical Library Application Selection panel.

For more information on pseudo libraries, see "Pseudo optical library concept" on page 42.

Defining real 3995 libraries

Choose option 3 from the Optical Library Application Selection panel to display the 3995 Library Define panel with all the input fields blanked out. You can then enter a sample definition, as shown in Figure 41 on page 398 and Figure 42 on page 400 (for LIBRARY1), and as shown in Figure 43 on page 401 and Figure 44 on page 402 (for LIB1C). When the panel is complete, press the END key.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                             3995 LIBRARY DEFINE               Page 1 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARY1
Library Type :REAL

To DEFINE Library, Specify:

  Description ==>
              ==>

  Model Number . . . . .  133
  Base Device Number  . . 0940      (Valid base device number)
  Controlling Library . .           (Library Name)        (if expansion unit)
  Default Media Type  . . 3995
  Default Pseudo Library. PSEUDO1   (Pseudo Library name)

Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 41. 3995 Library Define Panel for LIBRARY1 (Page 1 of 2)

The following are field descriptions for the 3995 Library Define panel (1 of 2). The information provided on this panel is stored in the OAM configuration database and an entry is made in the current source control data set:

DESCRIPTION
A 120-byte field that allows you to enter a description of the library definition for use by the installation. There are no restrictions on its content.

BASE DEVICE NUMBER
Specify the base device number of a IBM 3995 controlling library model. This field is required for all *controlling* library models. The low order digit of this base device number must be 0 (for example, 1AC0).

The base device number for the IBM 3995-11*x* expansion unit models must be equal to the base device number of the controlling library (the IBM 3995-13*x* models) + 8, for example, 0948. The base device number for the expansion units (3995-11*x* models) is automatically calculated and does not need to be specified when defining these libraries.

The base device number for the IBM 3995-C*xx* models must match the base device number assigned to the device when it was installed. See "Defining 3995 device numbers" on page 390 for more information on the device numbers.

CONTROLLING LIBRARY
Specify the name of the 3995 control unit (3995-13x or 3995-C3A) to which the 3995 library expansion unit is connected. This field is required for all connected 3995 library expansion unit models. The following lists the controlling library and the associated expansion unit:

3995-131
3995-111

3995-132
3995-112

3995-133
3995-113

3995-C3A
3995-C32, 3995-C34, 3995-C36, 3995-C38, 3995-C12, 3995-C16, 3995-C18

Note: The 3995-C12, 3995-C16, and 3995-C18 must be attached to its corresponding 3995-C32, -C36, and -C38 before being attached to the 3995-C3A controlling library. See Table 3 on page 36 for more details concerning these devices.

The system connectivity defined for the controlling library is inherited by the libraries attached to it.

DEFAULT MEDIA TYPE
Specifying a default media type limits the type of media that can be entered into the specified optical library dataserver. It is also used as a criteria for the output volume selection for a grouped write request when using a multifunction library that is referenced by the Object or the Object Backup storage group to which the grouped write request is being written. The valid values for the default media type are:

3995
Any 3995 5.25-inch single-, double-, quad-, or 8*x*-density, WORM, or rewritable optical disk media. This is the default.

3995REWR
3995 5.25-inch, single-, double-, quad-, or 8*x*-density rewritable optical disk media.

3995WORM
3995 5.25-inch, single-, double-, quad-, or 8*x*-density WORM optical disk media.

3995-1
Only 3995 5.25-inch, single-density, WORM or rewritable optical disk media.

3995-1RW
Only 3995 5.25-inch, single-density, rewritable optical disk media.

3995-1WO
Only 3995 5.25-inch, single-density, WORM optical disk media.

3995-2
: Only 3995 5.25-inch, double-density, WORM or rewritable optical disk media.

3995-2RW
: Only 3995 5.25-inch, double-density, rewritable optical disk media.

3995-2WO
: Only 3995 5.25-inch, double-density, WORM optical disk media.

3995-4
: Only 3995 5.25-inch, quad-density, rewritable or WORM optical disk media.

3995-4RW
: Only 3995 5.25-inch, quad-density, rewritable optical disk media.

3995-4WO
: Only 3995 5.25-inch, quad-density, WORM optical disk media.

3995-8
: Only 3995 5.25-inch, 8*x*-density, rewritable or WORM optical disk media.

3995-8RW
: Only 3995 5.25-inch, 8*x*-density, rewritable optical disk media.

3995-8WO
: Only 3995 5.25-inch, 8*x*-density, WORM optical disk media.

Note:

1. Double-density, quad-, and 8*x*-density, WORM includes CCW media.
2. CCW is continuous composite WORM media. WORM is write-once-read-many media.
3. The 3995-SW3 drives (used within the C3A, C1*x*, and C3*x* libraries) are capable of only reading single-density WORM or rewritable media. A 3995-SW3 cannot handle write requests to this media. It is capable of reading from and writing to all other 3995 optical media types. The 3995-SW4 drives used within these libraries are capable of only reading from single- or double-density WORM or rewritable media. However, these drives are capable of reading from and writing to quad- or 8*x*-density WORM or rewritable media. Keep this in mind when deciding the appropriate default media type for an optical library.

DEFAULT PSEUDO LIBRARY
: The name of the pseudo library to which the volume is assigned after it is ejected from this real library.

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                             3995 LIBRARY DEFINE               Page 2 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARY1
Library Type :REAL

  Initial Online Status (Yes, No, or Blank):
    SYSTEM1  ===> YES  SYSTEM2  ===> NO   SYSTEM3  ===> NO   SYSTEM4  ===>
    SYSTEM5  ===>      SYSTEM6  ===>      SYSTEM7  ===>      SYSTEM8  ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>


Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 42. 3995 Library Define Panel for LIBRARY1 (Page 2 of 2)

The following is a description of the information required for the 3995 Library Define panel (2 of 2). The information provided on this panel is stored in the OAM configuration database and an entry is made in the current source control data set:

INITIAL ONLINE STATUS

Indicates the library connectivity to specified systems when this SCDS is activated. The library status is set to this value each time this SCDS is activated. The library must be connected to at least one system. The library only can be online to one system at a time. The default is blank.

- Y (YES) for online.
- N (NO) for offline.
- `Blank` for not connected.

Figure 43 on page 401 shows how to define a 3995-C3A controlling library (LIB1C). The following information must be provided:

- CDS Name—`'SCDS.PRIMARY'`
- Library Name—`LIB1C`
- Library Device Number—`3995-C3A`
- Select Option 3 (Define) to continue the definition

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                             3995 LIBRARY DEFINE               Page 1 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIB1C
Library Type :REAL

To Define Library, Specify:
  Description ==>
              ==>

  Model Number . . . . .  C3A
  Base Device Number  . . 0900      (Valid base device number)
  Controlling Library . .           (Library Name)        (if expansion unit)
  Default Media Type  . . 3995
  Default Pseudo Library.           (Pseudo Library name)

Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 43. 3995 Library Define Panel for LIB1C (Page 1 of 2)

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                             3995 LIBRARY DEFINE               Page 2 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIB1C
Library Type :REAL

  Initial Online Status (Yes, No, or Blank):
    SYSTEM1  ===> YES  SYSTEM2  ===> NO   SYSTEM3  ===> NO   SYSTEM4  ===>
    SYSTEM5  ===>      SYSTEM6  ===>      SYSTEM7  ===>      SYSTEM8  ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>


Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 44. 3995 Library Define Panel for LIB1C (Page 2 of 2)

After you complete the library definitions (for LIBRARY1, LIB1C, or both) the Optical Library Application Selection menu (Figure 40 on page 397) is displayed again, and you can define your next library. Enter the name of the library into the library name field, the model number into the model number field, and select option 3. The 3995 Library Define panel is again displayed. Figure 45 on page 402 and Figure 46 on page 403, Figure 47 on page 403 and Figure 48 on page 404, and Figure 49 on page 404 and Figure 50 on page 405 provide sample definitions for LIBRARY2 (3995-113), LIBRARYA (3995-C38), and LIBRARYB (3995-C18).

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                             3995 LIBRARY DEFINE               Page 1 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARY2
Library Type :REAL

To Define Library, Specify:
  Description ==>
              ==>

  Model Number . . . . .  113
  Base Device Number  . . 0948     (Valid device number)
  Controlling Library . . LIBRARY1 (Library Name)        (if expansion unit)
  Default Media Type  . . 3995WORM
  Default Pseudo Library. PSEUDO1  (Pseudo Library name)

Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 45. 3995 Library Define Panel for LIBRARY2 (Page 1 of 2)

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                             3995 LIBRARY DEFINE               Page 2 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARY2
Library Type :REAL

  Initial Online Status (Yes, No, or Blank):
    SYSTEM1  ===> YES  SYSTEM2  ===> NO   SYSTEM3  ===> NO   SYSTEM4  ===>
    SYSTEM5  ===>      SYSTEM6  ===>      SYSTEM7  ===>      SYSTEM8  ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>


Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 46. 3995 Library Define Panel for LIBRARY2 (Page 2 of 2)

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                             3995 LIBRARY DEFINE               Page 1 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARYA
Library Type :REAL

To Define Library, Specify:
  Description ==>
              ==>

  Model Number . . . . .  C38
  Base Device Number  . . 0904     (Valid base device number)
  Controlling Library . . LIB1C    (Library Name)       (if expansion unit)
  Default Media Type  . . 3995
  Default Pseudo Library. PSEUDO1  (Pseudo Library name)

Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 47. 3995 Library Define Panel for LIBRARYA (Page 1 of 2)

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                             3995 LIBRARY DEFINE               Page 2 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARYA
Library Type :REAL

  Initial Online Status (Yes, No, or Blank):
    SYSTEM1  ===> YES  SYSTEM2  ===> NO   SYSTEM3  ===> NO   SYSTEM4  ===>
    SYSTEM5  ===>      SYSTEM6  ===>      SYSTEM7  ===>      SYSTEM8  ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>


Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 48. 3995 Library Define Panel for LIBRARYA (Page 2 of 2)

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                             3995 LIBRARY DEFINE               Page 1 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARYB
Library Type :REAL

To Define Library, Specify:
  Description ==>
              ==>

  Model Number . . . . .  C18
  Base Device Number  . . 090C     (Valid base device number)
  Controlling Library . . LIB1C    (Library Name)       (if expansion unit)
  Default Media Type  . . 3995
  Default Pseudo Library. PSEUDO1  (Pseudo Library name)

Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 49. 3995 Library Define Panel for LIBRARYB (Page 1 of 2)

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                             3995 LIBRARY DEFINE               Page 2 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARYB
Library Type :REAL

  Initial Online Status (Yes, No, or Blank):
    SYSTEM1  ===> YES  SYSTEM2  ===> NO   SYSTEM3  ===> NO   SYSTEM4  ===>
    SYSTEM5  ===>      SYSTEM6  ===>      SYSTEM7  ===>      SYSTEM8  ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>


Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 50. 3995 Library Define Panel for LIBRARYB (Page 2 of 2)

Defining pseudo libraries

After you complete the definition of the last real library, display the Optical Library Application Selection menu (Figure 40 on page 397) again, and you can then define your pseudo library. Enter the SCDS name, the Library Name, the Library Device Type (if it is an old default pseudo library), and the pseudo library type into the Library Type field, and then select option 3 (Define). Figure 51 on page 405 is displayed.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                             PSEUDO LIBRARY DEFINE
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :PSEUDO1
Library Type :PSEUDO

To Define Library, Specify:
  Description ==> Department HRA functional testing data
              ==>







Use ENTER to Perform Verification;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 51. Pseudo Library Define Panel for PSEUDO1

Hit ENTER to perform the pseudo library definition. This definition is added to the SCDS.

Defining additional optical libraries

You can copy existing optical library definitions and modify them to create new ones by using the COPY line operator from the Optical Library List panel.

For more information on how to copy these existing optical library definitions, see “Copying optical library and drive definitions” on page 416.

Defining optical drives

After completing the previous library definitions, you have defined real libraries (LIBRARY1, LIBRARY2, LIBRARYA, LIBRARYB, and LIB1C) and a pseudo library (PSEUDO1). Define drives for each of these libraries. For this sample configuration, define one drive for each library (library LIB1C is not illustrated). Define two operator-accessible drives for the pseudo library. Define all of these libraries within the same SCDS named SCDS.PRIMARY.

The ISMF screens that follow provide examples for defining 3995-11*x*, 3995-13*x*, and 3995-SW3 (the drive used with all the 3995-C*xx* models) drives. Where applicable, the ISMF screens are duplicated with examples for the different drive definitions.

Table 49 on page 406 shows the names of the libraries and their associated drive names and drive device types as defined in the sample configuration:

Table 49. Optical libraries and their associated drive names and drive device types

Library	Drive name	Drive device type
LIBRARY1	LIB1D1	3995-133
LIBRARY2	LIB2D1	3995-113
LIBRARYA	LIBAD1	3995-SW3
LIBRARYB	LIBBD1	3995-SW3
PSEUDO1	OPA1	3995-133
PSEUDO1	OPDRV1	3995-SW3

From the Library Management Selection menu, (Figure 39 on page 396), select option 2 (Optical Drive Configuration), to display the Optical Drive Application Selection menu, as shown in Figure 52 on page 406. For each of the drives to be defined, enter the drive name into the drive name field, the drive device type into the drive device type field, and choose option 3 to continue with the definition.

Note: The optical drives inherit their online and system connectivity for the configuration from the library to which they belong, or in the case of operator-accessible drives, from their controlling library.

```
   Panel  Utilities  Help
 ------------------------------------------------------------------------------
                      OPTICAL DRIVE APPLICATION SELECTION
 Command ===>_

 To Perform Drive operations, Specify:
   CDS Name  . . . . . . 'SCDS.PRIMARY'
                                 (1 to 44 Character Data Set Name or 'ACTIVE')
   Drive Name  . . . . . LIB1D1     (For Optical Drive List, fully or
                                     Partially Specified or * for all)
   Drive Device Type . . 3995-133   (For Optical Drive List, fully or
                                     Partially Specified or * for all)

Select One of the following options:
  3  1. List     - Generate a list of Drives
     2. Display - Display a Drive
     3. Define  - Define a Drive
     4. Alter   - Alter a Drive

If List Option is Chosen,
   Enter "/" to select option    _ Respecify View Criteria
                                 _ Respecify Sort Criteria
 Use ENTER to Perform Selection;
 Use HELP Command for Help; Use END Command to Exit.
```

Figure 52. Optical Drive Application Selection Panel

Note: After the first drive definition, ISMF primes CDS NAME, DRIVE NAME, and DRIVE DEVICE TYPE with the last used reference values on the Optical Drive Application Selection panel. To define a drive, you must specify the name of an SCDS in CDS NAME, provide a name in the DRIVE NAME field, and a device type in the DRIVE DEVICE TYPE field. (The default CDS NAME is the single-quoted word 'ACTIVE', which represents the currently active configuration.) The 'ACTIVE' SCDS cannot be modified.

Choose option 3 to display the 3995 Drive Define panel with all input fields blanked out.

Note: You can leave the Drive Define panel at any time without saving optical disk drive attributes by issuing the CANCEL command.

Figure 53 on page 407 shows the drive being defined for LIBRARY1.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                            3995 DRIVE DEFINE
Command ===>_

SCDS Name . . :SCDS.PRIMARY
Drive Name  . :LIB1D1
Model Number  :133

To Define Drive, Specify:

  Description ===>
              ===>

  Library Name  . . . . LIBRARY1             (1 to 8 characters)
  Drive Number  . . . . 1
  Operator Accessible Drive . . Y            (Y or N)

The Following Field is for Operator Accessible Drive Type Only:
  Controlling Library Name . .               (1 to 8 characters)

Use ENTER to Perform Verification;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 53. 3995 Drive Define Panel for LIB1D1

SCDS NAME, DRIVE NAME, and MODEL NUMBER are output fields that contain the SCDS name, drive name, and model number of the drive device type you specified in the Optical Drive Application Selection panel.

Specify the following information in the OAM configuration database and make an entry in the current SCDS:

DESCRIPTION
A 120-byte field that allows you to enter a description of the drive definition. There are no restrictions on its content.

LIBRARY NAME
A 1- to 8-character library name to which the drive is assigned. For operator-accessible drives, this field is the name of a pseudo optical library.

DRIVE NUMBER
A 3-character field representing the position the drive occupies in the library. This is a required field for 3995 models and should have the following values:

- 1 to 4 for library-resident drives; 5 for operator accessible 3995-1*xx* models
- 1 to 999 for all other 3995 models

OPERATOR ACCESSIBLE DRIVE
Specify if this drive is an operator-accessible drive that should be connected to a controlling library.

CONTROLLING LIBRARY NAME
Specify the name of the 3995-13*x* or -C3A model optical library to which the operator-accessible disk drive is connected.

Figure 54 on page 408 shows the drive definition for library LIBRARY2.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                            3995 DRIVE DEFINE
Command ===>_

SCDS Name . . :SCDS.PRIMARY
Drive Name  . :LIB2D1
Model Number  :113

To Define Drive, Specify:

  Description ===>
              ===>

  Library Name  . . . . LIBRARY2             (1 to 8 characters)
  Drive Number  . . . . 1
  Operator Accessible Drive . . N            (Y or N)

The Following Field is for Operator Accessible Drive Type Only:
  Controlling Library Name . .               (1 to 8 characters)

Use ENTER to Perform Verification;
Use HELP command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 54. 3995 Drive Define Panel for LIB2D1

Figure 55 on page 408 shows the drive definition for library LIBRARYA.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                            3995 DRIVE DEFINE
Command ===>_

SCDS Name . . :SCDS.PRIMARY
Drive Name  . :LIBAD1
Model Number  :SW3

To Define Drive, Specify:

  Description ===>
              ===>

  Library Name  . . . . LIBRARYA            (1 to 8 characters)
  Drive Number  . . . . 1
  Operator Accessible Drive . . N           (Y or N)

The Following Field is for Operator Accessible Drive Type Only:
  Controlling Library Name . .              (1 to 8 characters)

Use ENTER to Perform Verification;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 55. 3995 Drive Define Panel for LIBAD1

Figure 56 on page 409 shows the drive definition for library LIBRARYB.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                            3995 DRIVE DEFINE
Command ===>_

SCDS Name . . :SCDS.PRIMARY
Drive Name  . :LIBBD1
Model Number  :SW3

To Define Drive, Specify:

  Description ===>
              ===>

  Library Name  . . . . LIBRARYB             (1 to 8 characters)
  Drive Number  . . . . 1
  Operator Accessible Drive . . N            (Y or N)

The Following Field is for Operator Accessible Drive Type Only:
  Controlling Library Name . .               (1 to 8 characters)

Use ENTER to Perform Verification;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 56. 3995 Drive Define Panel for LIBBD1

Figure 57 on page 409 shows the drive definition for the pseudo library, PSEUDO1.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                            3995 DRIVE DEFINE
Command ===>_

SCDS Name . . :SCDS.PRIMARY
Drive Name  . :OPA1
Model Number  :133

To Define Drive, Specify:

  Description ===>
              ===>

  Library Name  . . . . PSEUDO1             (1 to 8 characters)
  Drive Number  . . . . 5
  Operator Accessible Drive . . Y           (Y or N)

The Following Field is for Operator Accessible Drive Type Only:
  Controlling Library Name . . LIBRARY1     (1 to 8 characters)

Use ENTER to Perform Verification;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 57. 3995 Drive Define Panel for OPA1

Figure 58 on page 410 shows the drive definition for the pseudo library, PSEUDO1.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                            3995 DRIVE DEFINE
Command ===>_

SCDS Name . . :SCDS.PRIMARY
Drive Name  . :OPDRV1
Model Number  :SW3

To Define Drive, Specify:

  Description ===>
              ===>

  Library Name  . . . . PSEUDO1            (1 to 8 characters)
  Drive Number  . . . . 1
  Operator Accessible Drive . .Y           (Y or N)

 The Following Field is for Operator Accessible Drive Type Only:
  Controlling Library Name . . LIB1C       (1 to 8 characters)

Use ENTER to Perform Verification;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 58. 3995 Drive Define Panel for OPDRV1

Defining additional optical disk drives

You can copy existing optical disk drive definitions and modify them to create new ones by using the COPY line operator from the Optical Drive List panel.

For more information on how to copy these existing optical disk drive definitions, see "Copying optical library and drive definitions" on page 416.

Maintaining and modifying optical libraries and optical drives

After defining your optical libraries and optical disk drives to the SCDS and the OAM configuration database, you might find that you need to change some of the definitions originally assigned to them. The following information provides options on how you can alter, copy, change, and delete definitions for optical libraries and optical disk drives.

Altering a 3995 optical library

You can use the optical library alter option to alter the attributes of an existing optical library. Altering a library results in updating the library definition within the specified SCDS and the attributes stored in the OAM configuration database. The alter option is available only when the OAM address space is NOT active.

You can modify an optical library to change its definition in the OAM configuration database by using the 3995 LIBRARY ALTER panel, Figure 59 on page 411 and Figure 60 on page 413. To modify an optical library, start from the Library Application Selection panel, shown in Figure 40 on page 397, and specify the name of the SCDS containing the optical library you want to change. Specify the optical library name and select option 4, ALTER. ISMF displays the appropriate library alter panel.

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                             3995 LIBRARY ALTER               Page 1 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARY1
Library Type :REAL

To ALTER Library, Specify:
  Description ==>
              ==>

  Model Number  . . . . .  133
  Base Device Number  . . 0940     (Valid base device number)
  Controlling Library . .          (Library Name)        (if expansion unit)
  Default Media Type  . . 3995
  Default Pseudo Library.          (Pseudo Library name)

Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 59. 3995 Library Alter Panel (Page 1 of 2)

The following fields can be modified on the 3995 Library Alter panel (Page 1 of 2) to alter the 3995 library definition in the specified SCDS and in the OAM configuration database for LIBRARY1.

MODEL NUMBER

Indicates the model number of the optical library. You can only change this field for 3995-1*xx* models. You cannot change the model number if the following conditions exist:

- The requested library is a PSEUDO library.
- The original library is one of the following model numbers: C3A, C1*x*, or C3*x*.

  Note: For simplicity, this publication refers to the following 3995 library models as C1*x* and C3*x* respectively: C12, C16, C18, C32, C34, C36, C38.
- The new model number specified is one of the following model numbers: C3A, SW3, SW4, C3*x*.

If you change the model number of a 3995-13*x* to that of another -13*x* model (for example, changing a model number from 131 to 133), you must perform the following steps:

1. Delete any existing operator-accessible drive from the SCDS **before** you change the model number if one had been previously defined as part of the 3995-13*x* optical library model.
2. Change the model number.
3. Redefine the operator-accessible drive by specifying the pseudo optical library name.
4. Repeat the above steps for other SCDSs that have the operator-accessible drive defined in them. These other SCDSs must have the same new model number so that they can be validated.

   Note: If the model 3995-13*x* optical library has a model 3995-11*x* optical library expansion unit connected to it before the MODEL NUMBER field is changed, it continues to be connected after the MODEL NUMBER field is changed. The model number of the connected optical library expansion unit is also changed accordingly. For example, changing the model number from 131 to 132 also changes the model number of all attached drives in the library and expansion units.

If you change the MODEL NUMBER field from a -1*xx* to any other -1*xx* model (for example, changing the model number from a 111 to a 112), you must perform the following steps:

1. Change the optical library name in the CONTROLLING LIBRARY field to the appropriate corresponding -13*x* controlling library name.

   The system recalculates the base device number of the optical library and all the attached optical disk drives based on the base device number of the new controlling library specified.
2. Change the model number.

If you change the MODEL NUMBER field from a -13*x* model number to a -11*x* model number (for example, changing the model number from a 131 to a 111), you must perform the following steps:

1. Delete any existing operator-accessible drive from the SCDS that was previously defined as part of the 3995-13*x* optical library model.
2. Use the DELETE line operator to delete any connected -11*x* optical library **before** you change the model number.
3. Fill in the CONTROLLING LIBRARY field with the new controlling library name (3995-13*x*) to validate the controlling library.

   The system recalculates the base device number of the optical libraries and all of the attached optical disk drives based on the base device number of the new controlling library.

If you change the MODEL NUMBER field from a -11*x* model number to a -13*x* model number (for example, changing from a model number 113 to a 133), you must perform the following steps:

1. Fill in the BASE DEVICE NUMBER field with a valid 3995 device number.
2. Blank out the CONTROLLING LIBRARY NAME field.

   Note: If you are changing the model number to 131 or 132 from model 113, then the DEFAULT MEDIA TYPE field is blanked out. If you are changing the model number to 133 from model 111 or 112, then the DEFAULT MEDIA TYPE field is displayed as "3995".

BASE DEVICE NUMBER

Changes to the base device number for a 3995 optical controlling library has the following results:

- All the base device numbers for all connected optical disk drives are recalculated based on the new base device number specified.
- All the base device numbers for any connected 3995 optical library expansion units are recalculated based on the new base device number specified.
- All the base device numbers for all the optical disk drives connected to the attached optical library expansion units are recalculated based on the new base device number specified.

Note: You cannot manually change the base device number for a 3995 optical library expansion unit model because this number is derived by the system based on the controlling library to which it is attached.

CONTROLLING LIBRARY

Changes to the CONTROLLING LIBRARY field for a connected 3995 library expansion unit model, has the following result:

- The new controlling library is verified. The library base device number of the connected 3995 optical library expansion unit and all of the attached optical disk drives are recalculated based on the base device number of the new 3995 controlling library model whose name is specified in the CONTROLLING LIBRARY field.

DEFAULT MEDIA TYPE

Changes to the DEFAULT MEDIA TYPE field restrict the type of optical media that can be used for the 3995 optical libraries with multifunction optical disk drives.

DEFAULT PSEUDO LIBRARY

Changes to the DEFAULT PSEUDO LIBRARY field might result in the volume being associated with a different pseudo library when it is ejected from this real library.

```
  Panel  Utilities  Scroll   Help
------------------------------------------------------------------------------
                             3995 LIBRARY ALTER                Page 2 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARY1
Library Type :REAL

  Initial Online Status (Yes, No, or Blank):
    SYSTEM1  ===> YES  SYSTEM2  ===> NO   SYSTEM3  ===> NO   SYSTEM4  ===>
    SYSTEM5  ===>      SYSTEM6  ===>      SYSTEM7  ===>      SYSTEM8  ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>


Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 60. 3995 Library Alter Panel (Page 2 of 2)

The following field can be modified on the 3995 Library Alter panel (Page 2 of 2) to alter the 3995 library system connection definition in the specified SCDS and in the OAM configuration database for LIBRARY1.

INITIAL ONLINE STATUS

To change the system connectivity, or the online or offline status for the library, or both, alter the values on page 2 of 2 of the 3995 Library Alter panel (see Figure 60 on page 413). A library can be defined as connected to any system that is defined in the SCDS, whether it was originally defined to that system or not. The library status is set to this value each time this SCDS is activated. The library must be connected to at least one system. Only one library can be online and connected (YES) to any system at a time. The default is blank. The following values can be used to specify initial online status:

- **YES** — online and connected to the system
- **NO** — offline and connected to the system
- **blank** — offline and not connected to the system.

Changing the 3995 library connectivity

The 3995 LIBRARY ALTER panels also allow the storage administrator to alter the library system connectivity, online or offline status, or both. This allows the library to be connected to a different system (different from the currently logged on system or different from the system on which the library was originally defined).

This change in system connectivity for the library allows customers to test OAM software and 3995 hardware on a test system without interrupting the production system; however, once a 3995 library is connected to a test system, it is not available to the production system until it is reconnected and the SCDS is reactivated. If the system connectivity of a 3995 optical library is changed, all the storage groups associated with that library must also have their system enablement status changed so they are enabled to the same system as the library. The test system then can use the same SMS configuration as the production system.

Note: DB2 databases belonging to OAM must be defined to the DB2 subsystem on the test system, because these databases cannot be shared between systems, and the OAM configuration database used on the test system must be a copy of the one used on the production system.

To change the library's system connectivity and online status, assume a scenario where systems PRODSYS1-3 are part of an OAMplex on a production sysplex. Also assume that systems TESTSYS5-8 are being used as test systems and are not part of a separate test OAMplex on the same sysplex or a different sysplex.

Changing connectivity is done by changing the INITIAL ONLINE STATUS for system name PRODSYS1 to blank and the INITIAL ONLINE STATUS for system name TESTSYS5 to YES (see Figure 62 on page 415). LIBRARY1 is now online and connected to the TESTSYS5 system and can be used for testing. LIBRARY1 is now offline and not connected to any of the production systems (PRODSYS1-3) and is connected but offline to all the other test systems (TESTSYS6-8).

Changing the INITIAL ONLINE STATUS of a 3995 optical library dataserver controlling library model also updates the library definition in the specified SCDS to indicate the library is connected to the new system. This also changes the system connectivity for any optical library expansion unit connected to these controlling libraries. In addition, all of the optical drive definitions in the specified SCDS for all the optical disk drives associated with this library are updated to indicate the optical drives are connected to the new system.

Note: Changing the system connectivity of an optical library *does not* automatically change the system enablement status for the storage groups associated with that library. Therefore, the storage group definitions might need to be updated to provide the correct storage group enablement status. This allows the system connectivity of the optical library's associated storage groups' also to be changed from the OAMplex systems to the test system.

Attention: To reconnect the library to the original system (PRODSYS1), simply change the INITIAL ONLINE STATUS for system name PRODSYS1 back to Y (YES) and the system name TESTSYS1 back to N (NO) or BLANK. Also change the system enablement status of the storage groups which were changed from TESTSYS1 to PRODSYS1.

Indicate the library (LIBRARY1) for which the system connectivity and initial online status will change on the 3995 LIBRARY ALTER (Page 1 of 2) panel (Figure 61 on page 414).

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                             3995 LIBRARY ALTER               Page 1 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARY1
Library Type :REAL

To ALTER Library, Specify:
  Description ==>
              ==>

  Model Number . . . . .  133
  Base Device Number  . . 0940     (Valid base device number)
  Controlling Library . .          (Library Name)        (if expansion unit)
  Default Media Type  . . 3995
  Default Pseudo Library.          (Pseudo Library name)

Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 61. 3995 Library Alter Panel (Page 1 of 2)

The 3995 LIBRARY ALTER (Page 2 of 2) panel (Figure 62 on page 415) is displayed next. This panel is used to indicate the initial online status and system connectivity of the library to various systems in the installation.

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                             3995 LIBRARY ALTER               Page 2 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARY1
Library Type :REAL

  Initial Online Status (Yes, No, or Blank):
    PRODSYS1  ===>     PRODSYS2  ===>     PRODSYS3  ===>     PRODSYS4  ===>
    TESTSYS5  ===> YES TESTSYS6  ===> NO  TESTSYS7  ===> NO  TESTSYS8  ===> NO
              ===>               ===>               ===>               ===>
              ===>               ===>               ===>               ===>
              ===>               ===>               ===>               ===>
              ===>               ===>               ===>               ===>
              ===>               ===>               ===>               ===>
              ===>               ===>               ===>               ===>


Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 62. 3995 Library Alter Panel (Page 2 of 2)

Confirming a system mode conversion

The Conversion Confirmation panel (see Figure 63 on page 415) is displayed when there is an attempt to change the system mode of an SCDS from SYSTEMS(8) to SYSTEMS(32), or if the SCDS is in conflict with the current system mode for the active configuration. This panel occurs when the CDS being altered is in 8 system mode, and the SMS complex is in 32-system mode.

For more information regarding system modes, see *z/OS DFSMSdfp Storage Administration*.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                         CONVERSION CONFIRMATION PANEL
Command ===>

  To Confirm Conversion on the following CDS:

     CDS Name . . :SCDS.EIGHT.SYSTEM.MODE


  Specify the following:

      Perform Conversion:. . Y         (Y or N)



  This CDS is in 8-Name Mode, and the SMS complex is in 32-Name Mode.
  You must convert the CDS to 32-Name Mode prior to being able to
  access the CDS for update.  The conversion is PERMANENT.  A 32-Name
  Mode CDS cannot be converted to 8-Name Mode, be used by pre-DFSMS 1.3.0
  systems or be used by DFSMS 1.3.0 and above systems running in 8-Name Mode.

Use ENTER to Perform Operation;
Use Help Command for Help; Use END Command to Exit.
```

Figure 63. Conversion Confirmation Panel

The following is a description of the fields on the Conversion Confirmation Panel:

CDS NAME
: Specifies the name of the CDS identified on the ISMF Optical Library Application Selection panel. This CDS system mode is in conflict with the system mode of the active configuration.

PERFORM CONVERSION
: Specifies a conversion should be done for the CDS to convert it to 32-system mode.

  Note: A CDS in 32-system mode cannot be converted to 8-system mode. Only the reverse is possible.

Altering an optical disk drive

You can use the drive alter option to alter the attributes of an existing drive. Altering a drive definition results in updating the database drive row for that drive. The alter option is available only when OAM is **not** running.

To alter a drive definition in the OAM configuration database and the SCDS, start from the Drive Application Selection panel, shown in Figure 40 on page 397, and specify the name of the SCDS containing the drive you want to change. Specify the drive name and select option 4, ALTER. For the 3995 optical drive, ISMF displays the 3995 Drive Alter panel shown in Figure 64 on page 416.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                              3995 DRIVE ALTER
Command ===>

SCDS Name   . . . . . :'SCDS.TEMP.PRIMARY'
Drive Name  . . . . . :P1D0
Drive Type  . . . . . :LIBRARY
Model Number  . . . . :132
Controlling Lib Name  :LIBRARY1

To ALTER Drive, Specify:

  Description ===> Line 1
              ===> Line 2
  Drive Number . . . 1


The Following Field may be Changed by Operator Accessible Drives Only:
  Library Name . . .                      (1 to 8 characters)

Use ENTER to Perform Verification;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 64. 3995 Drive Alter Panel

Note:

1. You cannot change the drive number (always 5) for an operator-accessible drive (model 111, 112, or 113, the 3995 C-series models are exceptions).
2. For 3995 models 111, 112, 113, 131, 132, or 133, the drive number must be within the range of 1 to 4.
3. For all other 3995 models, the drive number must be within the range of 1 to 999.
4. You cannot change the library name for a drive unless it is an operator-accessible drive.

Copying optical library and drive definitions

You can copy existing optical drive and optical library definitions and modify them to create new ones using two methods. One method uses the attributes from the last optical disk drive or library definition. These values from the last definition are primed on the 3995 LIBRARY DEFINE or the 3995 DRIVE DEFINE panels, saving you from having to re-input similar data. It is simple to modify the attributes to define a new optical disk drive or optical library. After the attributes have been modified and you hit enter to finish the definition, the new optical disk drive or optical library is added to the SCDS and the OAM configuration database. A second method of copying existing optical disk drives or optical library definitions to create new optical disk drives or optical libraries uses the COPY line operator from the OPTICAL LIBRARY LIST panel (for optical libraries) or the OPTICAL DRIVE LIST panel (for optical disk drives). To do so, enter the COPY line operator in the LINE OPERATOR column next to the optical disk drive or optical library you wish to copy. Press ENTER to copy the existing optical disk drive or optical library, and the COPY ENTRY panel is displayed (see Figure 65 on page 417).

Note: The copy function requires that an controlling library be specified if you are copying an optical library expansion unit definition into a SCDS. The corresponding optical controlling library the SCDS is attached to must already be defined.

```
  Panel  Utilities  Help
 ------------------------------------------------------------------------------
                           COPY ENTRY PANEL
 Command ===>

   Definition will be copied from:

     Data Set Name . :'SCDS.TEMP.PRIMARY'
     Definition Name :LIB1
     Definition Type :OPTICAL LIBRARY

   Specify "Copy To" Definition:

     Data Set Name . .   'SCDS.TEMP.PRIMARY'
                                     (1 to 46 characters)

     Definition Name . .             (1 to 8 characters, fully specified)

     Enter "/" to select option       _  Perform Alter


  Use ENTER to Perform COPY;
  Use HELP Command for HELP; Use END command to Exit.
```

Figure 65. Copy Entry Panel

The *from* DATA SET NAME field identifies the source for the copy. It is primed with the value you specified on the Optical Drive (or Library) Application Selection panel. The *from* DEFINITION NAME field identifies the name of the optical disk drive or optical library to be copied. This field is primed with the value from the DRIVE NAME or LIBRARY NAME field of the Optical Drive (or Library) List panel.

The *to* DATA SET NAME field identifies the target SCDS of the copy. It must be a name of an SCDS. It is primed with the value of the *from* DATA SET NAME if the *from* DATA SET NAME contains an SCDS name. It is primed with blanks if the *from* DATA SET NAME is 'ACTIVE'. The *to* DEFINITION NAME field identifies the name of the optical disk drive or optical library. It is primed with blanks.

In the PERFORM ALTER field, indicate if you want to change some of the attributes of the source copy. If you specify Y (YES), the appropriate Alter panel is displayed. If you specify N (NO), you remain on the Copy Entry panel, where you can perform another copy or return to the original List panel.

When copying an optical disk drive or optical library definition from one SCDS into another SCDS, you do not need to select the PERFORM ALTER option. In the case where an optical disk drive or optical library definition is copied within the same SCDS, you must choose the PERFORM ALTER option because optical disk drives and optical libraries must differ from one definition to another. (In particular, those fields that are used in addressing an optical disk drive or optical library, such as CTC device number and SCSI address, must be unique.)

When you have specified the values, press ENTER to perform the copy.

Deleting an optical library

Note: Optical libraries are no longer supported in a Multiple OAM environment. If you need to delete an optical library, you must do this in a Classic OAM environment.

Note: When deleting Optical libraries, delete the Optical drives first and remove any references to the optical library from any of your Object Storage Group definitions.

You can delete an optical library definition within the specified SCDS. Before the optical library definition is deleted, all optical disk drives defined for that optical library are deleted and all storage groups constructs that reference the optical library are updated to not reference that optical library. This is done automatically as part of the optical library deletion process.

From the Optical Library List panel, enter DELETE in the LINE OPERATOR column next to the optical library you wish to delete. When you press ENTER, the Confirm Delete Request panel, shown in Figure 66 on page 418, is displayed.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                            CONFIRM DELETE REQUEST
Command ===>

  To Confirm Deletion on the following Optical Library:

     Optical Library Name :LIB1
     Residing in SCDS . . :'SCDS.TEMP.PRIMARY'

  Specify the following:
      Enter "/" to select option   _ Perform deletion

  Note:If Deletion is Performed, All Drive Definitions associated with the
        Library will be Deleted and all Storage Group Constructs that
        reference the Library will be Updated.  In addition, if the Library is
        a 3995 Model 131, 132, 133 or C3A, then any Library connected to it
        (such as a 3995 Model 111, 112, 113, C12, C16, C18, C32, C34, C36
        or C38) will also be deleted in the same manner.  Definitions
        will not be removed from the OAM Configuration Database
        DB2 tables.  DB2 (SPUFI) can be used for this purpose.

Use ENTER to Perform Operation;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 66. Deleting an Optical Library

Confirm that the displayed library is the one that you want to delete. If it is, enter Y for YES and press ENTER. The Optical Library List should appear with '*DELETE' in the LINE OPERATOR column next to the deleted library.

Note: If an optical controlling library is deleted, any attached optical library expansion unit and optical disk drives are also deleted. To delete an optical library from the OAM configuration database, use the DB2I or SPUFI tools.

Deleting an optical disk drive

Note: Optical libraries are no longer supported in a Multiple OAM environment. If you need to delete an optical drive, you must do this in a Classic OAM environment.

You can delete an optical disk drive definition within the specified SCDS. From the Optical Drive List panel, enter DELETE in the LINE OPERATOR column next to the optical disk drive you wish to delete. When you press ENTER, the Confirm Delete Request panel, shown in Figure 67 on page 419, is displayed. To delete an optical disk drive from the OAM configuration database, use the DB2I or SPUFI tools.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                            CONFIRM DELETE REQUEST
Command ===>

  To Confirm Deletion on the Following Optical Drive:

    Optical Drive Name :LIB0
    Residing in SCDS . :'SCDS.TEMP.PRIMARY'



  Specify the Following:
      Enter "/" to select option   _ Perform Deletion



  Note: If deletion is performed, the drive definition will not be removed
         from the OAM Configuration Database DB2 tables.



Use ENTER to Perform Operation;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 67. Deleting an Optical Disk Drive

Confirm that the displayed drive is the one that you want to delete. If it is, enter Y for YES and press ENTER. The Optical Drive List appears with *DELETE in the LINE OPERATOR column next to the deleted drive.

Deleting a drive has no effect on the OAM configuration database. When you delete a drive, the definition for that drive is removed from the specified SCDS. When the SCDS is reactivated, the deleted drive is unavailable to the system. If you are removing this DRIVE permanently and are planning to delete the drive from the OAM configuration database as well, ensure that you also delete the drive from any other SCDSs that references the drive before you remove the row for the drive from the OAM configuration database; otherwise, OAM initialization will fail. To delete a drive from the OAM configuration database, use the DB2I or SPUFI tools.

Using DELETE FORCE to delete an optical library or optical drive

Under certain circumstances, the ISMF Library Management DELETE line operator will not work. To get around this problem the FORCE parameter is used. To cause deletion of an optical library or optical disk drive when the standard DELETE line operator does not work, enter DELETE FORCE.

Remember, deleting an optical library or optical drive has no effect on the OAM configuration database. Instead, when you delete an optical library or an optical disk drive, the definition for that deleted device is unavailable to the system when the SCDS is reactivated. Therefore, once the record in the SCDS is removed, you can redefine the record using the values found in the OAM configuration database. Make sure you save the names of the optical libraries, optical disk drives, or both so they can be redefined.

Attention: DELETE FORCE will *not* delete any other entries in the SCDS. For example, if a library is deleted using the normal DELETE line operator, any optical disk drives associated with the optical library will also be deleted. However, if the optical library is deleted using DELETE FORCE, the optical disk drives associated with the optical library will not be deleted.

Appendix B. Sample library members

This appendix provides examples of some of the SAMPLIB members that enable you to install and use OAM. The SAMPLIB members provided include the following examples:

OAM processing depends on the DB2 tables that are created by the sample jobs. It is crucial that the tables are created, and migrated where applicable, exactly as specified in the sample jobs without change. Changes to columns or other characteristics of these tables might cause errors initializing or using OAM. These tables are not intended as an interface. Though accessible through system administration authority, access should be restricted from end users.

Changing system libraries

SAMPLIB members CBRAPROC and CBRIPROC are supplied to create the OAM and OTIS procedures in PROCLIB.

CBRAPROC

SAMPLIB member CBRAPROC, as shown here, creates member OAM in PROCLIB.

```
//CBRAPROC JOB  MSGLEVEL=(1,1),MSGCLASS=A                               00050000
//********************************************************************  00058300
//*                                                                     00066600
//*  $SEG(CBRAPROC) COMP(OSMC) PROD(OAM):                               00067400
//*                                                                     00068200
//*  OAM Update PROCLIB Job (for OAM procedure).                        00069000
//*                                                                     00069001
//*  Licensed Materials - Property of IBM                               00069002
//*  5650-ZOS                                                           00069003
//*  COPYRIGHT IBM CORP. 1989, 2017                                     00069004
//*                                                                     00069800
//*  This job will create a procedure in PROCLIB that can be used       00070600
//*  to start OAM.                                                      00071400
//*                                                                     00072200
//*  NOTE: If the DB2 load module library containing DSNALI is          00073000
//*        not in the LNKLST concatenation, either include              00073800
//*        the DB2 load module library in the SYS1.LINKLIB              00074600
//*        concatenation (LNKLSTxx) or add a STEPLIB DD to              00075400
//*        this PROCEDURE.                                              00076200
//*                                                                     00076300
//*        If a STEPLIB is used, then that concatenation must be        00076400
//*        APF-authorized.                                              00076500
//*                                                                     00076600
//*    Note:                                                            00076700
//*      If you want to have access to SETOAM, SETOPT, SETOSMC,         00076800
//*      SETDISK, SETTLIB, and OAMXCF statements in the                 00076900
//*      PARMLIB member (required for many functions, such as           00077000
//*      writing to tape volumes, using an OAMplex, multiple            00077100
//*      backups, etc), you must update this job step to               00077200
//*      include 'OAM=&OAM , and you must supply the default            00077300
//*      OAM=xx (where xx is the low order suffix of your CBROAMxx      00077400
//*      PARMLIB member) specification on the PROC statement.           00077500
//*      Refer to OAM Planning, Installation, and Storage               00077600
//*      Administration Guide for Object Support for more info.         00077700
//*                                                                     00077800
//*      If you are using a multiple OAM configuration, you must        00077801
//*      update this job step to include 'D=&DB2ID' in the PARM= string 00077802
//*      and you must supply the default DB2ID=xxxx specification on    00077803
//*      the PROC statement. D= is required in a multiple OAM           00077804
//*      configuration. It specifies the SSID or Group Attachment Name  00077805
//*      of the DB2 subsystem associated with the OAM address space     00077806
//*      being started with this procedure if it is for Object          00077807
//*      processing or NONE if the OAM address space is for Tape        00077808
//*      Library processing.                                            00077809
//*                                                                     00077810
//*      Refer to OAM Planning, Installation, and Storage               00077811
//*      Administration Guide for Object Support for more info.         00077812
//*                                                                     00077813
//*  CHANGE ACTIVITY:                                                   00077900
//*    $L0=JDP3227 320 890601 TUCJRL: Initial Release                   00078600
//*    $P1=KBI0238 331 900904 TUCKHB: Added the UNLOAD keyword and      00079400
//*                                   made OSMC, MAXS, and UNLOAD       00080200
//*                                   procedure variables.              00081000
//*    $L1=HDZ11C0 130 940818 TUCGRD: Added the EJECT keyword           00082100
//*    $01=OW22202 1C0 960809 TUCLJT: Added the RESTART keyword         00082600
//*    $P2=K190347 R19 060921 TUCBLC: RESTART is reserved word so       00082800
//*                                   change to REST                    00083000
//*    $L2=OAMR23M R23 160330 TUCAED: Added the D= keyword for multi    00083001
//*                                   OAM support                       00083002
//*                                                                     00083200
//********************************************************************  00091500
//         EXEC PGM=IEBUPDTE,PARM=NEW                                   00100000
//SYSPRINT DD   SYSOUT=A                                                00150000
//SYSUT2   DD   DSNAME=SYS1.PROCLIB,DISP=SHR                            00200000
//SYSIN    DD   DATA                                                    00250000
./         ADD   NAME=OAM,LEVEL=01,SOURCE=0,LIST=ALL                    00300000
./      NUMBER   NEW1=10,INCR=10                                        00350000
//OAM PROC OSMC=YES,MAXS=2,UNLOAD=9999,EJECT=LRW,REST=YES               00400000
//IEFPROC EXEC PGM=CBROAM,REGION=0M,                                    00450000
// PARM=('OSMC=&OSMC,APLAN=CBROAM,MAXS=&MAXS,UNLOAD=&UNLOAD',           00500000
//    'EJECT=&EJECT','RESTART=&REST')                                   00550000
//SYSABEND DD SYSOUT=A                                                  00600000
./      ENDUP                                                           00650000
/*                                                                      00700000
```

CBRIPROC

SAMPLIB member CBRIPROC, as shown here, creates member OTIS in PROCLIB.

```
//CBRIPROC JOB  MSGLEVEL=(1,1),MSGCLASS=A                                00050000
//********************************************************************  00100000
//*                                                                     00150000
//*  $SEG(CBRIPROC) COMP(OSR) PROD(OAM):                                00200000
//*                                                                     00250000
//*  OAM Update PROCLIB Job (for OTIS procedure).                       00300000
//*                                                                     00300001
//*  Licensed Materials - Property of IBM                               00300002
//*  5650-ZOS                                                           00300003
//*  COPYRIGHT IBM CORP. 1991, 1993                                     00300004
//*                                                                     00350000
//*  This job will create a procedure in PROCLIB that can be used       00400000
//*  to start OTIS.                                                     00450000
//*                                                                     00500000
//*  NOTE: If the DB2 load module library containing DSNALI is          00550000
//*        not in the LNKLST concatenation, either include              00600000
//*        the DB2 load module library in the SYS1.LINKLIB              00650000
//*        concatenation (LNKLSTxx) or add a STEPLIB DD to              00700000
//*        this PROCEDURE.                                              00750000
//*                                                                     00762500
//*        If a STEPLIB is used, then that concatenation must be        00775000
//*        APF-authorized.                                              00787500
//*                                                                     00800000
//*                                                                     00850000
//*  CHANGE ACTIVITY:                                                   00900000
//*    $L0=JDP3347 331 910614 TUCJRL: OTIS Support                      00950000
//*    $P1=KBN0003 331 910930 TUCTNN: Change instructions in comment    01000000
//*                                   block OTIS PROC from SYSLIB to    01050000
//*                                   STEPLIB                           01100000
//*    $P2=KBP0114 410 920625 TUCEMB: to be consistent with CBRAPROC    01150000
//*                                   added note above and deleted      01200000
//*                                   STEPLIB DD statement              01250000
//*                                                                     01300000
//********************************************************************  01350000
//STEP1     EXEC PGM=IEBUPDTE,PARM=NEW                                  01400000
//SYSPRINT DD   SYSOUT=A                                                01450000
//SYSUT2   DD   DSNAME=SYS1.PROCLIB,DISP=SHR                            01500000
//SYSIN    DD   DATA                                                    01550000
./         ADD   NAME=OTIS,LEVEL=01,SOURCE=0,LIST=ALL                   01600000
./      NUMBER   NEW1=10,INCR=10                                        01650000
//OTIS     PROC                                                         01700000
//IEFPROC EXEC PGM=CBRIIAS,REGION=0M                                    01750000
//SYSABEND DD SYSOUT=A                                                  01800000
./      ENDUP                                                           01850000
/*                                                                      01900000
```

CBRCTI00

SAMPLIB member CBRCTI00 is no longer needed, as the CTICBR00 member is now shipped in PARMLIB.

Creating object databases

To create the object databases for OAM, several jobs are supplied as members in SAMPLIB. Three members contain the data set allocation jobs and three members contain the DB2 database definition jobs. The CBRIALC0 job allocates the VSAM data sets for the DB2 object storage databases, and the CBRIALCX and CBRIALCY jobs allocate the VSAM data sets for the DB2 object administration database. Similarly, the CBRISQL0 database definition job defines the object storage databases, and the CBRISQLX and CBRISQLY jobs define the object administration database. CBRILOB is the OAM DB2 Data Set Allocation and Database Definition job for LOB Storage Structures (LOB table spaces, Base tables, Base table views, Auxiliary tables and Auxiliary index).

For each database used, the corresponding allocation and database definition job steps must be run successfully. You must modify the jobs for your installation before you run the jobs. See the job prologs for the required modifications and related information.

CBRIALC0

SAMPLIB member CBRIALC0, as shown here, provides data set allocation for the OAM object tables and directories. You must modify and run the job successfully before you use OAM.

```
//CBRIALC0 JOB  MSGLEVEL=(1,1),MSGCLASS=A                               00050000
//************************************************************          00100000
//*                                                                     00150000
//*  $SEG(CBRIALC0) COMP(OSR) PROD(OAM):                                00200000
//*                                                                     00250000
//*  OAM DB2 Data Set Allocation Job (for Object Tables                 00300000
//*  and Directories).                                                  00350000
//*                                                                     00350001
//*  Licensed Materials - Property of IBM                               00350002
//*  5650-ZOS                                                           00350003
//*  COPYRIGHT IBM CORP. 1989, 2009                                     00350004
//*                                                                     00400000
//*  Run CBRIALC0 to define a VSAM LDS that will be                     00450000
//*  used by DB2 for an OSR object database.                            00500000
//*                                                                     00550000
//*  Prior to executing this job you need to make the                   00600000
//*  following modifications:                                           00650000
//*                                                                     00700000
//*  1. Change "vol_ser" to the volume serials that your                00750000
//*     target database should reside on.                              00800000
//*  2. Change "pri_alloc" and "sec_alloc" to the desired               00850000
//*     number of cylinders for each particular VSAM LDS                00900000
//*     being defined. For example, CYLINDER(pri_alloc                  00950000
//*     sec_alloc) may be CYLINDER(200 10).                             01000000
//*  3. Change "cat_name" to the name of the catalog you                01050000
//*     will be using under DB2.                                        01100000
//*  4. If you intend on using the DSN1COPY utility to copy             01112500
//*     these data bases, then you must include the REUSE keyword       01125000
//*     in the DEFINE CLUSTER command for each data base.               01137500
//*  5. Change "osg_hlq" to the high level qualifier to be used         01138700
//*     for the object storage group definition and tables.             01139900
//*     This is the qualifier used on the object storage group          01141100
//*     define through ISMF and used by OAM and OSR for all access      01142300
//*     to the object storage group's directories and data tables.      01143500
//*  6. Add additional job steps, repeating all statements in the       01144700
//*     first STEP01, for each object storage group defined in your     01145900
//*     configuration.  In each repeated step, change the qualifier     01147100
//*     to match the qualifier for each object storage group.           01148300
//*                                                                     01150000
//*  Following data set allocations, run CBRISQL0 (provided             01200000
//*  in SAMPLIB for your modification) to define                        01250000
//*  DB2 databases, table spaces, indexes, views, etc.                  01300000
//*  using the data sets allocated by this job.                         01350000
//*                                                                     01400000
//*  If you have run this job and want to start over                    01450000
//*  again, just issue a DROP for each database that was                01500000
//*  previously defined in DB2.                                         01550000
//*                                                                     01600000
//*  CHANGE ACTIVITY:                                                   01650000
//*    $L0=JDP3227  320 890601 TUCJRL: Initial Release                  01700000
//*                                                                     01750000
//*    $L1=HDZ11E0  150 970812 TUCLJT: GROUP00-GROUP99 qualifier        01758300
//*                                    restriction removed.  Single     01766600
//*                                    set of JCL provided and user     01774900
//*                                    to customize to installation.    01783200
//*    $L2=OAMR1B   R1B 080807 TUCBLC: Prolog cleanup (Bug Sug #82)     01787300
//*                                                                     01791500
//************************************************************          01800000
//STEP01  EXEC PGM=IDCAMS                                               01850000
//SYSPRINT DD SYSOUT=*                                                  01900000
//SYSUDUMP DD SYSOUT=*                                                  01950000
//SYSIN    DD *                                                         02000000
 DELETE                                                  -              02050000
 cat_name.DSNDBC.osg_hlq.OSMDTS.I0001.A001               -              02100000
     CLUSTER                                             -              02150000
     PURGE                                                              02200000
 DELETE                                                  -              02250000
 cat_name.DSNDBC.osg_hlq.OSMOTS04.I0001.A001             -              02300000
     CLUSTER                                             -              02350000
     PURGE                                                              02400000
 DELETE                                                  -              02450000
 cat_name.DSNDBC.osg_hlq.OSMOTS32.I0001.A001             -              02500000
     CLUSTER                                             -              02550000
     PURGE                                                              02600000
 SET LASTCC=0                                                           02650000
```

```
SET MAXCC=0                                                            02700000
DEFINE CLUSTER                                        -                02750000
(NAME(cat_name.DSNDBC.osg_hlq.OSMDTS.I0001.A001)      -                02800000
        LINEAR                                        -                02850000
        SHAREOPTIONS(3 3)                             -                02900000
        VOLUMES(vol_ser)                              -                02950000
        CYLINDERS(pri_alloc sec_alloc)                -                03000000
        UNIQUE   )                                    -                03050000
      DATA                                            -                03100000
(NAME(cat_name.DSNDBD.osg_hlq.OSMDTS.I0001.A001))                      03150000
DEFINE CLUSTER                                        -                03200000
(NAME(cat_name.DSNDBC.osg_hlq.OSMOTS04.I0001.A001)    -                03250000
        LINEAR                                        -                03300000
        SHAREOPTIONS(3 3)                             -                03350000
        VOLUMES(vol_ser)                              -                03400000
        CYLINDERS(pri_alloc sec_alloc)                -                03450000
        UNIQUE   )                                    -                03500000
      DATA                                            -                03550000
(NAME(cat_name.DSNDBD.osg_hlq.OSMOTS04.I0001.A001))                    03600000
DEFINE CLUSTER                                        -                03650000
(NAME(cat_name.DSNDBC.osg_hlq.OSMOTS32.I0001.A001)    -                03700000
        LINEAR                                        -                03750000
        SHAREOPTIONS(3 3)                             -                03800000
        VOLUMES(vol_ser)                              -                03850000
        CYLINDERS(pri_alloc sec_alloc)                -                03900000
        UNIQUE   )                                    -                03950000
      DATA                                            -                04000000
(NAME(cat_name.DSNDBD.osg_hlq.OSMOTS32.I0001.A001))                    04050000
DELETE                                                -                04100000
cat_name.DSNDBC.osg_hlq.OBJDIRX1.I0001.A001           -                04150000
    CLUSTER                                           -                04200000
    PURGE                                                              04250000
DELETE                                                -                04300000
cat_name.DSNDBC.osg_hlq.OBJDIRX2.I0001.A001           -                04350000
    CLUSTER                                           -                04400000
    PURGE                                                              04450000
DELETE                                                -                04500000
cat_name.DSNDBC.osg_hlq.OBJDIRX3.I0001.A001           -                04550000
    CLUSTER                                           -                04600000
    PURGE                                                              04650000
DELETE                                                -                04700000
cat_name.DSNDBC.osg_hlq.OBJT04X1.I0001.A001           -                04750000
    CLUSTER                                           -                04800000
    PURGE                                                              04850000
DELETE                                                -                04900000
cat_name.DSNDBC.osg_hlq.OBJT32X1.I0001.A001           -                04950000
    CLUSTER                                           -                05000000
    PURGE                                                              05050000
SET LASTCC=0                                                           05100000
SET MAXCC=0                                                            05150000
DEFINE CLUSTER                                        -                05200000
(NAME(cat_name.DSNDBC.osg_hlq.OBJDIRX1.I0001.A001)    -                05250000
        LINEAR                                        -                05300000
        SHAREOPTIONS(3 3)                             -                05350000
        VOLUMES(vol_ser)                              -                05400000
        CYLINDERS(pri_alloc sec_alloc)                -                05450000
        UNIQUE   )                                    -                05500000
      DATA                                            -                05550000
(NAME(cat_name.DSNDBD.osg_hlq.OBJDIRX1.I0001.A001))                    05600000
DEFINE CLUSTER                                        -                05650000
(NAME(cat_name.DSNDBC.osg_hlq.OBJDIRX2.I0001.A001)    -                05700000
        LINEAR                                        -                05750000
        SHAREOPTIONS(3 3)                             -                05800000
        VOLUMES(vol_ser)                              -                05850000
        CYLINDERS(pri_alloc sec_alloc)                -                05900000
        UNIQUE   )                                    -                05950000
      DATA                                            -                06000000
(NAME(cat_name.DSNDBD.osg_hlq.OBJDIRX2.I0001.A001))                    06050000
DEFINE CLUSTER                                        -                06100000
(NAME(cat_name.DSNDBC.osg_hlq.OBJDIRX3.I0001.A001)    -                06150000
        LINEAR                                        -                06200000
        SHAREOPTIONS(3 3)                             -                06250000
        VOLUMES(vol_ser)                              -                06300000
        CYLINDERS(pri_alloc sec_alloc)                -                06350000
        UNIQUE   )                                    -                06400000
      DATA                                            -                06450000
(NAME(cat_name.DSNDBD.osg_hlq.OBJDIRX3.I0001.A001))                    06500000
DEFINE CLUSTER                                        -                06550000
(NAME(cat_name.DSNDBC.osg_hlq.OBJT04X1.I0001.A001)    -                06600000
        LINEAR                                        -                06650000
        SHAREOPTIONS(3 3)                             -                06700000
        VOLUMES(vol_ser)                              -                06750000
```

```
         CYLINDERS(pri_alloc sec_alloc)              -                    06800000
         UNIQUE   )                                  -                    06850000
       DATA                                          -                    06900000
 (NAME(cat_name.DSNDBD.osg_hlq.OBJT04X1.I0001.A001))                      06950000
 DEFINE CLUSTER                                      -                    07000000
 (NAME(cat_name.DSNDBC.osg_hlq.OBJT32X1.I0001.A001)  -                    07050000
         LINEAR                                      -                    07100000
         SHAREOPTIONS(3 3)                           -                    07150000
         VOLUMES(vol_ser)                            -                    07200000
         CYLINDERS(pri_alloc sec_alloc)              -                    07250000
         UNIQUE   )                                  -                    07300000
       DATA                                          -                    07350000
 (NAME(cat_name.DSNDBD.osg_hlq.OBJT32X1.I0001.A001))                      07400000
```

CBRIALCX

SAMPLIB member CBRIALCX, as shown here, provides data set allocation for part of the OAM administration tables. You must modify and run the job successfully before you use OAM.

```
//CBRIALCX JOB  MSGLEVEL=(1,1),MSGCLASS=A                                 00050000
//*************************************************************           00100000
//*                                                                       00150000
//*  $SEG(CBRIALCX) COMP(OSR) PROD(OAM):                                  00200000
//*                                                                       00250000
//*  OAM DB2 Data Set Allocation Job (for Administration                  00300000
//*  Databases).                                                          00350000
//*                                                                       00350001
//*  Licensed Materials - Property of IBM                                 00350002
//*  5650-ZOS                                                             00350003
//*  COPYRIGHT IBM CORP. 1989, 2009                                       00350004
//*                                                                       00400000
//*  Run CBRIALCX to define the VSAM LDSs that                            00450000
//*  will be used by DB2 for the OAMADMIN databases                       00500000
//*  required by OAM.                                                     00550000
//*                                                                       00600000
//*  Prior to executing this job you need to make the                     00650000
//*  following modifications:                                             00700000
//*                                                                       00750000
//*  1. Change "vol_ser" to the volume serials that your                  00800000
//*     target database should reside on.                                 00850000
//*  2. Change "pri_alloc" and "sec_alloc" to the desired                 00900000
//*     number of cylinders for each particular VSAM LDS                  00950000
//*     being defined. For example, CYLINDER(pri_alloc                    01000000
//*     sec_alloc) may be CYLINDER(200 10).                               01050000
//*  3. Change "cat_name" to the name of the catalog you                  01100000
//*     will be using under DB2.                                          01150000
//*  4. If you intend on using the DSN1COPY utility to copy               01162500
//*     these data bases, then you must include the REUSE keyword         01175000
//*     in the DEFINE CLUSTER command for each data base.                 01187500
//*                                                                       01200000
//*  Following data set allocations, run CBRISQLX (provided               01250000
//*  in SAMPLIB for your modification) to define the                      01300000
//*  OAM Administration databases using the data sets                     01350000
//*  allocated by this job.                                               01400000
//*                                                                       01450000
//*  If you have run this job and want to start over                      01500000
//*  again, just issue a DROP for each OSR database                       01550000
//*  in DB2 that was previously defined by this job.                      01600000
//*                                                                       01650000
//*  CHANGE ACTIVITY:                                                     01700000
//*    $L0=JDP3227 320 890601 TUCJRL: Initial Release                     01750000
//*    $L1=OAMR1B  R1B 080807 TUCBLC: Prolog (Bug Sug #82)                01775000
//*                                                                       01800000
//*************************************************************           01850000
//STEP0X  EXEC PGM=IDCAMS                                                 01900000
//SYSPRINT DD SYSOUT=*                                                    01950000
//SYSUDUMP DD SYSOUT=*                                                    02000000
//SYSIN    DD *                                                           02050000
 DELETE                                              -                    02100000
 cat_name.DSNDBC.OAMADMIN.MCIND.I0001.A001           -                    02150000
     CLUSTER                                         -                    02200000
     PURGE                                                                02250000
 DELETE                                              -                    02300000
 cat_name.DSNDBC.OAMADMIN.SCIND.I0001.A001           -                    02350000
     CLUSTER                                         -                    02400000
     PURGE                                                                02450000
 DELETE                                              -                    02500000
 cat_name.DSNDBC.OAMADMIN.COLIND.I0001.A001          -                    02550000
     CLUSTER                                         -                    02600000
```

```
    PURGE                                                               02650000
  SET LASTCC=0                                                          02700000
  SET MAXCC=0                                                           02750000
  DEFINE CLUSTER                                       -                02800000
  (NAME(cat_name.DSNDBC.OAMADMIN.MCIND.I0001.A001)     -                02850000
          LINEAR                                       -                02900000
          SHAREOPTIONS(3 3)                            -                02950000
          VOLUMES(vol_ser)                             -                03000000
          CYLINDERS(pri_alloc sec_alloc)               -                03050000
          UNIQUE   )                                   -                03100000
        DATA                                           -                03150000
  (NAME(cat_name.DSNDBD.OAMADMIN.MCIND.I0001.A001))                     03200000
  DEFINE CLUSTER                                       -                03250000
  (NAME(cat_name.DSNDBC.OAMADMIN.SCIND.I0001.A001)     -                03300000
          LINEAR                                       -                03350000
          SHAREOPTIONS(3 3)                            -                03400000
          VOLUMES(vol_ser)                             -                03450000
          CYLINDERS(pri_alloc sec_alloc)               -                03500000
          UNIQUE   )                                   -                03550000
        DATA                                           -                03600000
  (NAME(cat_name.DSNDBD.OAMADMIN.SCIND.I0001.A001))                     03650000
  DEFINE CLUSTER                                       -                03700000
  (NAME(cat_name.DSNDBC.OAMADMIN.COLIND.I0001.A001)    -                03750000
          LINEAR                                       -                03800000
          SHAREOPTIONS(3 3)                            -                03850000
          VOLUMES(vol_ser)                             -                03900000
          CYLINDERS(pri_alloc sec_alloc)               -                03950000
          UNIQUE   )                                   -                04000000
        DATA                                           -                04050000
  (NAME(cat_name.DSNDBD.OAMADMIN.COLIND.I0001.A001))                    04100000
  DELETE                                               -                04150000
  cat_name.DSNDBC.OAMADMIN.CBRMGTX.I0001.A001          -                04200000
      CLUSTER                                          -                04250000
      PURGE                                                             04300000
  DELETE                                               -                04350000
  cat_name.DSNDBC.OAMADMIN.CBRSTOX.I0001.A001          -                04400000
      CLUSTER                                          -                04450000
      PURGE                                                             04500000
  DELETE                                               -                04550000
  cat_name.DSNDBC.OAMADMIN.CBRCLTX1.I0001.A001         -                04600000
      CLUSTER                                          -                04650000
      PURGE                                                             04700000
  DELETE                                               -                04750000
  cat_name.DSNDBC.OAMADMIN.CBRCLTX2.I0001.A001         -                04800000
      CLUSTER                                          -                04850000
      PURGE                                                             04900000
  DELETE                                               -                04950000
  cat_name.DSNDBC.OAMADMIN.CBRCLTX3.I0001.A001         -                05000000
      CLUSTER                                          -                05050000
      PURGE                                                             05100000
  SET LASTCC=0                                                          05150000
  SET MAXCC=0                                                           05200000
  DEFINE CLUSTER                                       -                05250000
  (NAME(cat_name.DSNDBC.OAMADMIN.CBRMGTX.I0001.A001)   -                05300000
          LINEAR                                       -                05350000
          SHAREOPTIONS(3 3)                            -                05400000
          VOLUMES(vol_ser)                             -                05450000
          CYLINDERS(pri_alloc sec_alloc)               -                05500000
          UNIQUE   )                                   -                05550000
        DATA                                           -                05600000
  (NAME(cat_name.DSNDBD.OAMADMIN.CBRMGTX.I0001.A001))                   05650000
  DEFINE CLUSTER                                       -                05700000
  (NAME(cat_name.DSNDBC.OAMADMIN.CBRSTOX.I0001.A001)   -                05750000
          LINEAR                                       -                05800000
          SHAREOPTIONS(3 3)                            -                05850000
          VOLUMES(vol_ser)                             -                05900000
          CYLINDERS(pri_alloc sec_alloc)               -                05950000
          UNIQUE   )                                   -                06000000
        DATA                                           -                06050000
  (NAME(cat_name.DSNDBD.OAMADMIN.CBRSTOX.I0001.A001))                   06100000
  DEFINE CLUSTER                                       -                06150000
  (NAME(cat_name.DSNDBC.OAMADMIN.CBRCLTX1.I0001.A001)  -                06200000
          LINEAR                                       -                06250000
          SHAREOPTIONS(3 3)                            -                06300000
          VOLUMES(vol_ser)                             -                06350000
          CYLINDERS(pri_alloc sec_alloc)               -                06400000
          UNIQUE   )                                   -                06450000
        DATA                                           -                06500000
  (NAME(cat_name.DSNDBD.OAMADMIN.CBRCLTX1.I0001.A001))                  06550000
  DEFINE CLUSTER                                       -                06600000
  (NAME(cat_name.DSNDBC.OAMADMIN.CBRCLTX2.I0001.A001)  -                06650000
          LINEAR                                       -                06700000
```

```
        SHAREOPTIONS(3 3)                         -                00006750000
        VOLUMES(vol_ser)                          -                06800000
        CYLINDERS(pri_alloc sec_alloc)            -                06850000
        UNIQUE   )                                -                06900000
      DATA                                        -                06950000
 (NAME(cat_name.DSNDBD.OAMADMIN.CBRCLTX2.I0001.A001))              07000000
 DEFINE CLUSTER                                   -                07050000
 (NAME(cat_name.DSNDBC.OAMADMIN.CBRCLTX3.I0001.A001)  -            07100000
        LINEAR                                    -                07150000
        SHAREOPTIONS(3 3)                         -                07200000
        VOLUMES(vol_ser)                          -                07250000
        CYLINDERS(pri_alloc sec_alloc)            -                07300000
        UNIQUE   )                                -                07350000
      DATA                                        -                07400000
 (NAME(cat_name.DSNDBD.OAMADMIN.CBRCLTX3.I0001.A001))              07450000
```

CBRIALCY

SAMPLIB member CBRIALCY, as shown here, provides data set allocation for part of the OAM object tables and directories. You must modify and run the job successfully before you use OAM.

```
//CBRIALCY JOB  MSGLEVEL=(1,1),MSGCLASS=A                          00050000
//************************************************************     00100000
//*                                                                00150000
//*  $SEG(CBRIALCY) COMP(OSR) PROD(OAM):                           00200000
//*                                                                00250000
//*  OAM DB2 Data Set Allocation Job (for Administration           00300000
//*  Databases).                                                   00350000
//*                                                                00350001
//*  Licensed Materials - Property of IBM                          00350002
//*  5650-ZOS                                                      00350003
//*  COPYRIGHT IBM CORP. 1989, 2009                                00350004
//*                                                                00400000
//*  Run CBRIALCY to define the VSAM LDS that                      00450000
//*  will be used by DB2 for the OAMADMIN databases                00500000
//*  required by OAM.                                              00550000
//*                                                                00600000
//*  Prior to executing this job you need to make the              00650000
//*  following modifications:                                      00700000
//*                                                                00750000
//*  1. Change "vol_ser" to the volume serials that your           00800000
//*     target database should reside on.                          00850000
//*  2. Change "pri_alloc" and "sec_alloc" to the desired          00900000
//*     number of cylinders for each particular VSAM LDS           00950000
//*     being defined. For example, CYLINDER(pri_alloc             01000000
//*     sec_alloc) may be CYLINDER(200 10).                        01050000
//*  3. Change "cat_name" to the name of the catalog you           01100000
//*     will be using under DB2.                                   01150000
//*  4. If you intend on using the DSN1COPY utility to copy        01162500
//*     these data bases, then you must include the REUSE keyword  01175000
//*     in the DEFINE CLUSTER command for each data base.          01187500
//*                                                                01200000
//*  Following data set allocations, run CBRISQLY (provided        01250000
//*  in SAMPLIB for your modification) to define the               01300000
//*  OAM Administration databases using the data sets              01350000
//*  allocated by this job.                                        01400000
//*                                                                01450000
//*  CHANGE ACTIVITY:                                              01500000
//*    $00=OY28892 320 890601 TUCJRL: Add unique indexes to        01550000
//*                                   those defined in             01600000
//*                                   CBRIALCX                     01650000
//*    $L1=OAMR1B  R1B 080807 TUCBLC: Prolog (Bug Sug #82)         01675000
//*                                                                01700000
//************************************************************     01750000
//STEP0X  EXEC PGM=IDCAMS                                          01800000
//SYSPRINT DD SYSOUT=*                                             01850000
//SYSUDUMP DD SYSOUT=*                                             01900000
//SYSIN    DD *                                                    01950000
 DELETE                                           -                02000000
 cat_name.DSNDBC.OAMADMIN.CBRMGTY.I0001.A001      -                02050000
     CLUSTER                                      -                02100000
     PURGE                                                         02150000
 DELETE                                           -                02200000
 cat_name.DSNDBC.OAMADMIN.CBRSTOY.I0001.A001      -                02250000
     CLUSTER                                      -                02300000
     PURGE                                                         02350000
 DEFINE CLUSTER                                   -                02400000
 (NAME(cat_name.DSNDBC.OAMADMIN.CBRMGTY.I0001.A001)   -            02450000
        LINEAR                                    -                02500000
```

```
        SHAREOPTIONS(3 3)                    -                 02550000
        VOLUMES(vol_ser)                     -                 02600000
        CYLINDERS(pri_alloc sec_alloc)       -                 02650000
        UNIQUE   )                           -                 02700000
      DATA                                   -                 02750000
 (NAME(cat_name.DSNDBD.OAMADMIN.CBRMGTY.I0001.A001))          02800000
 DEFINE CLUSTER                              -                 02850000
 (NAME(cat_name.DSNDBC.OAMADMIN.CBRSTOY.I0001.A001)   -        02900000
        LINEAR                               -                 02950000
        SHAREOPTIONS(3 3)                    -                 03000000
        VOLUMES(vol_ser)                     -                 03050000
        CYLINDERS(pri_alloc sec_alloc)       -                 03100000
        UNIQUE   )                           -                 03150000
      DATA                                   -                 03200000
 (NAME(cat_name.DSNDBD.OAMADMIN.CBRSTOY.I0001.A001))          03250000
```

CBRILOB

SAMPLIB member CBRILOB, as shown here, creates the LOB storage structure for LOB support. You must modify and run the job successfully before you use OAM.

```
//CBRILOB JOB MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A                         00050000
//*********************************************************************  00100000
//*                                                                     00150000
//*  $SEG(CBRILOB) COMP(DBM) PROD(OAM):                                 00200000
//*                                                                     00250000
//*  OAM DB2 Data Set Allocation and Database Definition job for        00300000
//*  LOB Storage Structures (LOB tablespaces, Base tables, Base         00350000
//*  table views, Auxiliary tables and Auxiliary index.                 00400000
//*                                                                     00400001
//*  Licensed Materials - Property of IBM                               00400002
//*  5650-ZOS                                                           00400003
//*  COPYRIGHT IBM CORP. 2006, 2017                                     00400004
//*  ---------------------------------------------------------------    00450000
//*  ---------------------------------------------------------------    00500000
//*                                                                     00550000
//*  CBRILOB                                                            00600000
//*                                                                     00650000
//*  This job will                                                      00700000
//*  1. DEFINE VSAM LDS THAT WILL BE USED BY DB2 to create              00750000
//*     the LOB Storage Structure.                                      00800000
//*  2. Create the LOB base table, base table view, auxiliary table,    00850000
//*     and index that comprise the LOB Storage Structure within the    00900000
//*     Object Storage Table Hierarchy.                                 00950000
//*                                                                     01000000
//*  Prior to executing this job you need to make the                   01050000
//*  following modifications:                                           01100000
//*                                                                     01150000
//*  1. Change the name in the DSN SYSTEM(db2) statement to        @L2A 01150001
//*     the name of the DB2 Subsystem in your installation.        @L2A 01150003
//*                                                                @L2A 01150004
//*     In a multiple OAM configuration, you will create a         @L2A 01150005
//*     unique job for each Object OAM instance and specify        @L2A 01150006
//*     the DB2 SSID for the DB2 subsystem associated with         @L2A 01150007
//*     the Object OAM instance.                                   @L2A 01150008
//*                                                                @L2A 01150009
//*  2. Change the data set name in the RUN statement              @L2A 01150010
//*     LIB('db2xx.RUNLIB.LOAD') phrase to the data set name used  @L2A 01150011
//*     in your installation for the DB2 RUNLIB.LOAD data set.     @L2A 01150012
//*                                                                @L2A 01150013
//*     In a multiple OAM configuration, you will create a         @L2A 01150014
//*     unique job for each Object OAM instance and specify        @L2A 01150015
//*     the DB2 RUNLIB.LOAD for the DB2 subsystem associated with  @L2A 01150016
//*     the Object OAM instance.                                   @L2A 01150017
//*                                                                @L2A 01150018
//*  3. Change the PLAN name (DSNTIAxx) in the RUN statement to    @L2A 01150019
//*     match your current DB2 version and release level.          @L2A 01150020
//*  4. Change "vol_ser" to the volume serials that your            @L2C01200000
//*     target database should reside on.                               01250000
//*  5. Change "pri_alloc" and "sec_alloc" to the desired          @L2C 01300000
//*     number of cylinders for each particular VSAM LDS                01350000
//*     being defined. For example, CYLINDER(pri_alloc                  01400000
//*     sec_alloc) may be CYLINDER(200 10).                             01450000
//*  6. Change "cat_name" to the name of the catalog you           @L2C 01500000
//*     will be using under DB2.                                        01550000
//*  7. If you intend on using the DSN1COPY utility to copy        @L2C 01600000
//*     these data bases, then you must include the REUSE keyword       01650000
//*     in the DEFINE CLUSTER command for each data base.               01700000
//*  8. Change "osg_hlq" to the high level qualifier to be used    @L2C 01750000
```

```
//*     for the object storage group definition and tables.             01800000
//*     This is the qualifier used on the object storage group          01850000
//*     define through ISMF and used by OAM and OSR for all access      01900000
//*     to the object storage group's directories and data tables.      01950000
//*  9. Change "ds_size" to the maximum size alowed for each data  @L2C 02000000
//*     set.  Please refer to the DB2 for z/OS SQL reference            02050000
//*     manual for limitations.                                         02100000
//* 10. Change "auth_id" to the identifier(s)                      @L2C 02150000
//*     authorized for the respective group.                            02200000
//* 11. Add additional job steps, repeating all statements in           02600000
//*     STEP00 - STEP02, for each object storage group defined in       02650000
//*     your configuration.  In each repeated step, change the          02700000
//*     qualifier to match the qualifier for each object storage        02750000
//*     group.                                                          02800000
//*                                                                     02850000
//*                                                                     02900000
//*  If you have run this job and want to start over                    02950000
//*  again, just issue a DROP for each LOB base table and for each      03000000
//*  base tablespace and auxiliary tablespace that was previously       03050000
//*  defined in DB2 by this job.                                        03100000
//*                                                                     03150000
//*******************************************************************   03200000
//*                                                                     03250000
//*  CHANGE ACTIVITY:                                                   03300000
//*    $L0=HDZ1180  R18 050531 TUCGPW: INITIAL RELEASE                  03350000
//*    $P0=K180710  R18 051214 TUCGPW: Change ROW_ID to OTROWID         03375000
//*    $00=OA16562  R18 060519 TUCBLC: Change OTOBJ to 2G               03387500
//*    $L1=OAMR1B   R1B 080807 TUCBLC: OAMR1B Bug Sug #82               03393700
//*    $L2=OAMR23M  R23 160405 TUCAED: Multi OAM AS Support        @L2A 03393701
//*                                                                     03400000
//********************************************************************* 03450000
//STEP00  EXEC PGM=IDCAMS                                               03900000
//SYSPRINT DD SYSOUT=*                                                  03950000
//SYSUDUMP DD SYSOUT=*                                                  04000000
//SYSIN    DD *                                                         04050000
 DELETE                                               -                 04100000
 cat_name.DSNDBC.osg_hlq.OSMLBTS.I0001.A001           -                 04150000
     CLUSTER                                          -                 04200000
     PURGE                                                              04250000
 DELETE                                               -                 04300000
 cat_name.DSNDBC.osg_hlq.OTLOBX1.I0001.A001           -                 04350000
     CLUSTER                                          -                 04400000
     PURGE                                                              04450000
 DELETE                                               -                 04500000
 cat_name.DSNDBC.osg_hlq.OSMLATS.I0001.A001           -                 04550000
     CLUSTER                                          -                 04600000
     PURGE                                                              04650000
 DELETE                                               -                 04700000
 cat_name.DSNDBC.osg_hlq.OTLOBAX1.I0001.A001          -                 04750000
     CLUSTER                                          -                 04800000
     PURGE                                                              04850000
 SET LASTCC=0                                                           04900000
 SET MAXCC=0                                                            04950000
/*                                                                      05000000
//STEP01  EXEC PGM=IDCAMS                                               05050000
//SYSPRINT DD SYSOUT=*                                                  05200000
//SYSUDUMP DD SYSOUT=*                                                  05250000
//SYSIN    DD *                                                         05300000
 DEFINE CLUSTER                                       -                 05350000
 (NAME(cat_name.DSNDBC.osg_hlq.OSMLBTS.I0001.A001)    -                 05400000
         LINEAR                                       -                 05450000
         SHAREOPTIONS(3 4)                            -                 05500000
         VOLUMES(vol_ser)                             -                 05550000
         CYLINDERS(pri_alloc sec_alloc)               -                 05600000
         UNIQUE   )                                   -                 05650000
       DATA                                           -                 05700000
 (NAME(cat_name.DSNDBD.osg_hlq.OSMLBTS.I0001.A001))                     05750000
 DEFINE CLUSTER                                       -                 05800000
 (NAME(cat_name.DSNDBC.osg_hlq.OTLOBX1.I0001.A001)    -                 05850000
         LINEAR                                       -                 05900000
         SHAREOPTIONS(3 4)                            -                 05950000
         VOLUMES(vol_ser)                             -                 06000000
         CYLINDERS(pri_alloc sec_alloc)               -                 06050000
         UNIQUE   )                                   -                 06100000
       DATA                                           -                 06150000
 (NAME(cat_name.DSNDBD.osg_hlq.OTLOBX1.I0001.A001))                     06200000
 DEFINE CLUSTER                                       -                 06250000
 (NAME(cat_name.DSNDBC.osg_hlq.OSMLATS.I0001.A001)    -                 06300000
         LINEAR                                       -                 06350000
         SHAREOPTIONS(3 4)                            -                 06400000
         VOLUMES(vol_ser)                             -                 06450000
         CYLINDERS(pri_alloc sec_alloc)               -                 06500000
```

```
         UNIQUE   )                                   -
       DATA                                           -
 (NAME(cat_name.DSNDBD.osg_hlq.OSMLATS.I0001.A001))
 DEFINE CLUSTER                                       -
 (NAME(cat_name.DSNDBC.osg_hlq.OTLOBAX1.I0001.A001)   -
         LINEAR                                       -
         SHAREOPTIONS(3 4)                            -
         VOLUMES(vol_ser)                             -
         CYLINDERS(pri_alloc sec_alloc)               -
         UNIQUE   )                                   -
       DATA                                           -
 (NAME(cat_name.DSNDBD.osg_hlq.OTLOBAX1.I0001.A001))
/*
//STEP02  EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 DSN SYSTEM(db2)
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIAxx) -
      LIB('db2xx.RUNLIB.LOAD')
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
CREATE TABLESPACE OSMLBTS
       IN         osg_hlq
       USING      VCAT cat_name
       LOCKSIZE   ANY
       CLOSE      NO
       SEGSIZE    64
       BUFFERPOOL BP32K;
CREATE TABLE osg_hlq.OSM_LOB_BASE_TBL
       (
        OTVER    CHAR(1)      NOT NULL,
        OTSEG    SMALLINT     NOT NULL,
        OTCLID   INTEGER      NOT NULL,
        OTNAME   VARCHAR(44)  NOT NULL,
        OTROWID  ROWID        NOT NULL GENERATED ALWAYS,
        OTOBJ    BLOB(2G)     NOT NULL
       )
       IN osg_hlq.OSMLBTS;
CREATE UNIQUE INDEX osg_hlq.OTLOBX1
       ON osg_hlq.OSM_LOB_BASE_TBL
       (
        OTCLID   ASC,
        OTNAME   ASC
       )
       CLUSTER
       USING      VCAT cat_name
       CLOSE      NO
       BUFFERPOOL BP1
       PCTFREE    10;
COMMIT;
CREATE VIEW
       osg_hlq.V_OSM_LOB_BASE_TBL
     AS SELECT ALL * FROM
       osg_hlq.OSM_LOB_BASE_TBL;
GRANT ALL ON
       osg_hlq.V_OSM_LOB_BASE_TBL
     TO auth_id;
COMMIT;
CREATE LOB TABLESPACE OSMLATS
       IN         osg_hlq
       USING      VCAT cat_name
       LOG        NO
       LOCKSIZE   LOB
       BUFFERPOOL BP32K
       DSSIZE     ds_size
       GBPCACHE   SYSTEM;
CREATE AUXILIARY TABLE osg_hlq.OSM_LOB_AUX_TBL
       IN         osg_hlq.OSMLATS
       STORES     osg_hlq.OSM_LOB_BASE_TBL
       COLUMN     OTOBJ;
CREATE UNIQUE INDEX osg_hlq.OTLOBAX1
       ON osg_hlq.OSM_LOB_AUX_TBL
       USING      VCAT cat_name;
COMMIT;
/*
//
```

CBRISQL0

SAMPLIB member CBRISQL0, as shown here, provides the DB2 definitions for the OAM object tables and directories. You must modify and run the job successfully before you use OAM.

CBRISQL0 SAMPLIB Member

```
//CBRISQL0 JOB  MSGLEVEL=(1,1),MSGCLASS=A                              00002900
//******************************************************************   00005800
//*                                                                    00008700
//*  $SEG(CBRISQL0) COMP(OSR) PROD(OAM):                               00013000
//*                                                                    00017400
//*  OAM DB2 Database Definition Job (for Object Tables                00017600
//*  and Directories).                                                 00017800
//*                                                                    00017801
//*  Licensed Materials - Property of IBM                              00017802
//*  5650-ZOS                                                          00017803
//*  COPYRIGHT IBM CORP. 1989, 2017                                    00017804
//*                                                                    00018000
//*  This job will create an OSR database, table, and                  00018200
//*  index in DB2 for an object storage group.                         00018400
//*                                                                    00018600
//*  Before running this job, you must change the following:           00018800
//*                                                                    00019000
//*  1. Change the name in the DSN SYSTEM(db2) statement to        @L7A 00019001
//*     the name of the DB2 Subsystem in your installation.        @L7A 00019002
//*                                                                @L7A 00019003
//*     In a multiple OAM configuration, you will create a         @L7A 00019004
//*     unique job for each Object OAM instance and specify        @L7A 00019005
//*     the DB2 SSID for the DB2 subsystem associated with         @L7A 00019006
//*     the Object OAM instance.                                   @L7A 00019007
//*                                                                @L7A 00019008
//*  2. Change the data set name in the RUN statement              @L7A 00019009
//*     LIB('db2xx.RUNLIB.LOAD') phrase to the data set name used  @L7A 00019010
//*     in your installation for the DB2 RUNLIB.LOAD data set.     @L7A 00019011
//*                                                                @L7A 00019012
//*     In a multiple OAM configuration, you will create a         @L7A 00019013
//*     unique job for each Object OAM instance and specify        @L7A 00019014
//*     the DB2 RUNLIB.LOAD for the DB2 subsystem associated with  @L7A 00019015
//*     the Object OAM instance.                                   @L7A 00019016
//*                                                                @L7A 00019017
//*  3. Change the PLAN name (DSNTIAxx) in the RUN statement to    @L7A 00019018
//*     match your current DB2 version and release level.          @L7A 00019019
//*                                                                @L7A 00019020
//*  4. Change "cat_name" to the DB2 VCAT name used                @L7C 00019200
//*     for defining the VSAM data sets in CBRIALC0.                    00019400
//*                                                                    00019600
//*  5. Change "auth_id" to the identifier(s)                      @L7C 00019800
//*     authorized for the respective group.                           00020000
//*                                                                    00020200
//*  6. If you plan to use just one collection, reverse the            00022400
//*     order of ODCLID and ODPENDDT in index OBJDIRX2.                00022800
//*                                                                    00022900
//*  7. Change "osg_hlq" to the high level qualifier to be used        00023000
//*     for the object storage group definition and tables.            00023100
//*     This is the qualifier used on the object storage group         00023200
//*     define through ISMF and used by OAM and OSR for all access     00023300
//*     to the object storage group's directories and data tables.     00023400
//*                                                                    00023500
//*  8. Add additional job steps, repeating all statements in          00023600
//*     STEP01, for each object storage group defined in your          00023700
//*     configuration.  In each repeated step, change the qualifier    00023800
//*     to match the qualifier for each object storage group.          00023900
//*                                                                    00024000
//*  CHANGE ACTIVITY:                                                  00024100
//*    $L0=JDP3227 320 890601 TUCJRL: Initial Release                  00024200
//*    $01=OY26530 320 891113 TUCTNN: Removed OTSEG from OBJT04X1      00024300
//*    $02=OY33596 320 901019 TUCHTT: Changed index OBJDIRX1 to        00024400
//*                                   ODCREATS. Changed index          00024500
//*                                   OBJDIRX2 to ODPENDDT, ODCLID,    00024600
//*                                   ODCREATS. Changed index          00024700
//*                                   OBJDIRX3 to ODNAME, ODCLID.      00024800
//*    $L1=JDP3331 331 910614 TUCKSG: Reverse order of ODCLID          00024900
//*                                   and ODPENDDT for index           00025000
//*                                   OBJDIRX2.                        00025100
//*    $L2=HDZ11D0 140 970331 TUCSPP: Specify TYPE 1 INDEX for         00025200
//*                                   DB2 4.1 or above level           00025300
//*    $L3=HDZ11E0 150 970812 TUCLJT: GROUP00-GROUP99 qualifier        00025400
//*                                   restriction removed.  Single     00025500
//*                                   set of JCL provided and user     00025600
```

```
//*                                 to customize to installation.     00025700
//*    $L4=HDZ11G0 R13 001012 TUCLJT: Add ODBK2LOC and ODBK2SEC for     00025900
//*                                 Multiple Object Backup Support    00026100
//*                                 Also:                             00026300
//*                                 - Removed reference to type 1     00026500
//*                                   indexes, which are no longer    00026700
//*                                   supported by DB2                00026900
//*                                 - Removed SUBPAGES from CREATE    00027100
//*                                   statements, since they are      00027300
//*                                   only for type 1 indexes         00027500
//*    $P1=K170872 R13 010913 TUCLJT: Correct misplaced commas        00027600
//*    $P2=K1K0640 R17 041130 TUCVRE: INDEX OBJDIRX1 and OBJDIRX2     00027700
//*                                 do not need to be UNIQUE.         00027800
//*    $L5=HDZ1180 R18 050531 TUCGPW: Add ODLOBFL for LOB support     00027900
//*    $L6=HDZ1B10 R1B 080627 TUCBLC: Add ODSTATF and ODRETDT and     00028000
//*                                 ODINSTID and change               00028100
//*                                 OTOBJ to FOR BIT DATA             00028200
//*    $P3=K1B0208 R1B 080806 TUCBLC: Default ODRETDT to '0001-01-01'   00028300
//*    $L7=OAMR23M R23 160405 TUCAED: Multi OAM AS Support         @L7A 00028301
//*                                                                   00028400
//*******************************************************************      00028500
//STEP00  EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)                   00028600
//SYSTSPRT DD SYSOUT=*                                                00028700
//SYSTSIN  DD *                                                       00029000
 DSN SYSTEM(db2)                                                      00031900
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIAxx) -                               00034800
      LIB('db2xx.RUNLIB.LOAD')                                        00037700
//SYSPRINT DD SYSOUT=*                                                00040600
//SYSUDUMP DD SYSOUT=*                                                00043500
//SYSIN    DD *                                                       00046400
          CREATE DATABASE   osg_hlq;                                  00060700
          COMMIT;                                                     00075000
          CREATE TABLESPACE OSMDTS                                    00100000
                 IN         osg_hlq                                   00150000
                 USING      VCAT cat_name                             00200000
                 LOCKSIZE   ANY                                       00250000
                 CLOSE      NO                                        00300000
                 SEGSIZE    64                                        00325000
                 BUFFERPOOL BP0;                                      00350000
          CREATE TABLESPACE OSMOTS04                                  00400000
                 IN         osg_hlq                                   00450000
                 USING      VCAT cat_name                             00500000
                 LOCKSIZE   ANY                                       00550000
                 CLOSE      NO                                        00600000
                 SEGSIZE    64                                        00625000
                 BUFFERPOOL BP2;                                      00650000
          CREATE TABLESPACE OSMOTS32                                  00700000
                 IN         osg_hlq                                   00750000
                 USING      VCAT cat_name                             00800000
                 LOCKSIZE   ANY                                       00850000
                 CLOSE      NO                                        00900000
                 SEGSIZE    64                                        00925000
                 BUFFERPOOL BP32K;                                    00950000
          COMMIT;                                                     01000000
          CREATE TABLE osg_hlq.OSM_OBJ_DIR                            01048600
                 (                                                    01097300
                  ODVER    CHAR(1)      NOT NULL,                     01144600
                  ODSIZE   INTEGER      NOT NULL,                     01191900
                  ODCREATS TIMESTAMP    NOT NULL,                     01239200
                  ODEXPDT  DATE         NOT NULL,                     01286500
                  ODLREFDT DATE         NOT NULL,                     01333800
                  ODPENDDT DATE         NOT NULL,                     01381100
                  ODMCASDT DATE         NOT NULL,                     01428400
                  ODSCNUM  SMALLINT     NOT NULL,                     01475700
                  ODMCNUM  SMALLINT     NOT NULL,                     01523000
                  ODLOCFL  CHAR(1)      NOT NULL,                     01570300
                  ODLSLOC  CHAR(6)      NOT NULL,                     01617600
                  ODSECLOC INTEGER      NOT NULL,                     01664900
                  ODBKLOC  CHAR(6)      NOT NULL,                     01712200
                  ODBKSEC  INTEGER      NOT NULL,                     01759500
                  ODCLID   INTEGER      NOT NULL,                     01806800
                  ODNAME   VARCHAR(44)  NOT NULL,                     01830400
                  ODBK2LOC CHAR(6)      NOT NULL WITH DEFAULT,        01848100
                  ODBK2SEC INTEGER      NOT NULL WITH DEFAULT,        01865800
                  ODLOBFL  CHAR(1)      NOT NULL WITH DEFAULT,        01874700
                  ODSTATF  SMALLINT     NOT NULL WITH DEFAULT,        01883600
                  ODRETDT  DATE   NOT NULL WITH DEFAULT '0001-01-01',   01889500
                  ODINSTID INTEGER      NOT NULL WITH DEFAULT         01895400
                 )                                                    01901400
                 IN osg_hlq.OSMDTS;                                   01951000
          CREATE INDEX osg_hlq.OBJDIRX1                               02000600
                 ON osg_hlq.OSM_OBJ_DIR                               02050200
```

```
                (                                                                02100000
                 ODCREATS ASC                                                    02150000
                )                                                                02200000
                CLUSTER                                                          02233300
                USING      VCAT cat_name                                         02266600
                CLOSE      NO                                                    02300000
                BUFFERPOOL BP1                                                   02400000
                PCTFREE    10;                                                   02450000
         CREATE INDEX osg_hlq.OBJDIRX2                                           02500000
                ON osg_hlq.OSM_OBJ_DIR                                           02550000
                (                                                                02600000
                 ODCLID   ASC,                                                   02637500
                 ODPENDDT ASC,                                                   02675000
                 ODCREATS ASC                                                    02712500
                )                                                                02750000
                USING      VCAT cat_name                                         02800000
                CLOSE      NO                                                    02850000
                BUFFERPOOL BP1                                                   02950000
                PCTFREE    10;                                                   03000000
         CREATE UNIQUE INDEX osg_hlq.OBJDIRX3                                    03050000
                ON osg_hlq.OSM_OBJ_DIR                                           03100000
                (                                                                03150000
                 ODNAME   ASC,                                                   03200000
                 ODCLID   ASC                                                    03250000
                )                                                                03300000
                USING      VCAT cat_name                                         03350000
                CLOSE      NO                                                    04550000
                BUFFERPOOL BP1                                                   04650000
                PCTFREE    10;                                                   04700000
         COMMIT;                                                                 04750000
         CREATE TABLE osg_hlq.OSM_04K_OBJ_TBL                                    04800000
                (                                                                04850000
                 OTVER    CHAR(1)      NOT NULL,                                 04900000
                 OTSEG    SMALLINT     NOT NULL,                                 04950000
                 OTCLID   INTEGER      NOT NULL,                                 04975000
                 OTNAME   VARCHAR(44)  NOT NULL,                                 05000000
                 OTOBJ    LONG VARCHAR FOR BIT DATA NOT NULL                     05050000
                )                                                                05100000
                IN osg_hlq.OSMOTS04;                                             05150000
         CREATE UNIQUE INDEX osg_hlq.OBJT04X1                                    05200000
                ON osg_hlq.OSM_04K_OBJ_TBL                                       05250000
                (                                                                05300000
                 OTCLID   ASC,                                                   05350000
                 OTNAME   ASC                                                    05400000
                )                                                                05450000
                CLUSTER                                                          05500000
                USING      VCAT cat_name                                         05550000
                CLOSE      NO                                                    05600000
                BUFFERPOOL BP1                                                   05700000
                PCTFREE    10;                                                   05750000
         CREATE TABLE osg_hlq.OSM_32K_OBJ_TBL                                    05800000
                (                                                                05850000
                 OTVER    CHAR(1)      NOT NULL,                                 05900000
                 OTSEG    SMALLINT     NOT NULL,                                 05950000
                 OTCLID   INTEGER      NOT NULL,                                 05975000
                 OTNAME   VARCHAR(44)  NOT NULL,                                 06000000
                 OTOBJ    LONG VARCHAR FOR BIT DATA NOT NULL                     06050000
                )                                                                06100000
                IN osg_hlq.OSMOTS32;                                             06150000
         CREATE UNIQUE INDEX osg_hlq.OBJT32X1                                    06200000
                ON osg_hlq.OSM_32K_OBJ_TBL                                       06250000
                (                                                                06300000
                 OTCLID   ASC,                                                   06325000
                 OTNAME   ASC,                                                   06350000
                 OTSEG    ASC                                                    06400000
                )                                                                06450000
                CLUSTER                                                          06500000
                USING      VCAT cat_name                                         06550000
                CLOSE      NO                                                    06600000
                BUFFERPOOL BP1                                                   06700000
                PCTFREE    10;                                                   06750000
         COMMIT;                                                                 06800000
         CREATE VIEW                                                             06850000
                osg_hlq.V_OSM_OBJ_DIR                                            06900000
              AS SELECT ALL * FROM                                               06950000
                osg_hlq.OSM_OBJ_DIR;                                             07000000
         CREATE VIEW                                                             07050000
                osg_hlq.V_OSM_04K_OBJ_TBL                                        07100000
              AS SELECT ALL * FROM                                               07150000
                osg_hlq.OSM_04K_OBJ_TBL;                                         07200000
         CREATE VIEW                                                             07250000
                osg_hlq.V_OSM_32K_OBJ_TBL                                        07300000
```

```
                AS SELECT ALL * FROM                                            07350000
                  osg_hlq.OSM_32K_OBJ_TBL;                                      07400000
           GRANT ALL ON                                                         07450000
                  osg_hlq.V_OSM_OBJ_DIR                                         07500000
                TO auth_id;                                                     07550000
           GRANT ALL ON                                                         07600000
                  osg_hlq.V_OSM_04K_OBJ_TBL                                     07650000
                TO auth_id;                                                     07700000
           GRANT ALL ON                                                         07750000
                  osg_hlq.V_OSM_32K_OBJ_TBL                                     07800000
                TO auth_id;                                                     07850000
           COMMIT;                                                              07900000
```

CBRISQLX

SAMPLIB member CBRISQLX, as shown here, provides DB2 definitions for part of the OAM administration database. You must modify and run the job successfully before you use OAM.

```
//CBRISQLX JOB  MSGLEVEL=(1,1),MSGCLASS=A                               00050000
//*******************************************************************   00100000
//*                                                                     00150000
//*  $SEG(CBRISQLX) COMP(OSR) PROD(OAM):                                00200000
//*                                                                     00250000
//*  OAM DB2 Database Definition Job (for Administration                00300000
//*  Databases).                                                        00350000
//*                                                                     00350001
//*  Licensed Materials - Property of IBM                               00350002
//*  5650-ZOS                                                           00350003
//*  COPYRIGHT IBM CORP. 1989, 2017                                     00350004
//*                                                                     00400000
//*  This job will create the OAM Administration databases,             00450000
//*  tables, and indexes in DB2.                                        00500000
//*                                                                     00550000
//*  Before running this job, you must change the following:            00600000
//*                                                                     00650000
//*  1. Change the name in the DSN SYSTEM(db2) statement to        @L3A 00650001
//*     the name of the DB2 Subsystem in your installation.        @L3A 00650002
//*                                                                @L3A 00650003
//*     In a multiple OAM configuration, you will create a         @L3A 00650004
//*     unique job for each Object OAM instance and specify        @L3A 00650005
//*     the DB2 SSID for the DB2 subsystem associated with         @L3A 00650006
//*     the Object OAM instance.                                   @L3A 00650007
//*                                                                @L3A 00650008
//*  2. Change the data set name in the RUN statement              @L3A 00650009
//*     LIB('db2xx.RUNLIB.LOAD') phrase to the data set name used  @L3A 00650010
//*     in your installation for the DB2 RUNLIB.LOAD data set.     @L3A 00650011
//*                                                                @L3A 00650012
//*     In a multiple OAM configuration, you will create a         @L3A 00650013
//*     unique job for each Object OAM instance and specify        @L3A 00650014
//*     the DB2 RUNLIB.LOAD for the DB2 subsystem associated with  @L3A 00650015
//*     the Object OAM instance.                                   @L3A 00650016
//*                                                                @L3A 00650019
//*  3. Change the PLAN name (DSNTIAxx) in the RUN statement to    @L3A 00650020
//*     match your current DB2 version and release level.          @L3A 00650021
//*                                                                @L3A 00650022
//*  4. Change "cat_name" to the DB2 VCAT name used                @L3C 00700000
//*     for defining the VSAM data sets in CBRIALCX.                    00750000
//*                                                                     00800000
//*  5. Change "auth_id" to the identifier(s)                      @L3C 00850000
//*     authorized for the respective group.                            00900000
//*                                                                     01450000
//*                                                                     01492600
//*  CHANGE ACTIVITY:                                                   01500000
//*    $L0=JDP3227 320 890601 TUCJRL: Initial Release                   01550000
//*    $L1=HDZ11D0 140 970331 TUCSPP: Specify TYPE 1 INDEX for          01566600
//*                                   DB2 4.1 or above level            01583200
//*    $L2=HDZ11G0 R13 001016 TUCLJT: Removed reference to type 1       01586500
//*                                   indexes, no longer supported      01589800
//*                                   by DB2 and removed SUBPAGES       01593100
//*                                   from CREATE statements            01596400
//*    $L3=OAMR23M R23 160405 TUCAED: Multi OAM AS Support         @L3A 01596401
//*                                                                     01600000
//*******************************************************************   01650000
//CREATE EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)                      01700000
//SYSTSPRT DD SYSOUT=*                                                  01750000
//SYSTSIN  DD *                                                         01800000
 DSN SYSTEM(db2)                                                        01850000
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIAxx) -                                 01900000
      LIB('db2xx.RUNLIB.LOAD')                                          01950000
```

```
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
          CREATE DATABASE   OAMADMIN;
          COMMIT;
          CREATE TABLESPACE MCIND
                 IN         OAMADMIN
                 USING      VCAT cat_name
                 LOCKSIZE   ANY
                 CLOSE      NO
                 BUFFERPOOL BP0;
          CREATE TABLESPACE SCIND
                 IN         OAMADMIN
                 USING      VCAT cat_name
                 LOCKSIZE   ANY
                 CLOSE      NO
                 BUFFERPOOL BP0;
          CREATE TABLESPACE COLIND
                 IN         OAMADMIN
                 USING      VCAT cat_name
                 LOCKSIZE   ANY
                 CLOSE      NO
                 BUFFERPOOL BP0;
          COMMIT;
          CREATE TABLE OAMADMIN.CBR_MGT_CLASS_TBL
                 (
                  ODMCNUM  SMALLINT    NOT NULL,
                  ODMCNAME VARCHAR(30) NOT NULL
                 )
                 IN OAMADMIN.MCIND;
          CREATE UNIQUE INDEX OAMADMIN.CBRMGTX
                 ON OAMADMIN.CBR_MGT_CLASS_TBL
                 (
                  ODMCNUM  ASC
                 )
                 USING      VCAT cat_name
                 CLOSE      NO
                 BUFFERPOOL BP1
                 PCTFREE    10;
          COMMIT;
          CREATE TABLE OAMADMIN.CBR_STO_CLASS_TBL
                 (
                  ODSCNUM  SMALLINT    NOT NULL,
                  ODSCNAME VARCHAR(30) NOT NULL
                 )
                 IN OAMADMIN.SCIND;
          CREATE UNIQUE INDEX OAMADMIN.CBRSTOX
                 ON OAMADMIN.CBR_STO_CLASS_TBL
                 (
                  ODSCNUM  ASC
                 )
                 USING      VCAT cat_name
                 CLOSE      NO
                 BUFFERPOOL BP1
                 PCTFREE    10;
          COMMIT;
          CREATE TABLE OAMADMIN.CBR_COLLECTION_TBL
                 (
                  ODCLSCNM VARCHAR(30) NOT NULL,
                  ODCLMCNM VARCHAR(30) NOT NULL,
                  ODCLSGNM VARCHAR(30) NOT NULL,
                  ODCLID   INTEGER     NOT NULL,
                  ODCLNAME VARCHAR(44) NOT NULL
                 )
                 IN OAMADMIN.COLIND;
          CREATE UNIQUE INDEX OAMADMIN.CBRCLTX1
                 ON OAMADMIN.CBR_COLLECTION_TBL
                 (
                  ODCLID   ASC
                 )
                 USING      VCAT cat_name
                 CLOSE      NO
                 BUFFERPOOL BP1
                 PCTFREE    10;
          CREATE UNIQUE INDEX OAMADMIN.CBRCLTX2
                 ON OAMADMIN.CBR_COLLECTION_TBL
                 (
                  ODCLNAME ASC
                 )
                 USING      VCAT cat_name
                 CLOSE      NO
                 BUFFERPOOL BP1
```

```
              PCTFREE    10;                                                    06300000
       CREATE INDEX OAMADMIN.CBRCLTX3                                           06350000
              ON OAMADMIN.CBR_COLLECTION_TBL                                    06400000
              (                                                                 06450000
               ODCLSGNM ASC                                                     06500000
              )                                                                 06550000
              USING      VCAT cat_name                                          06600000
              CLOSE      NO                                                     06650000
              BUFFERPOOL BP1                                                    06750000
              PCTFREE    10;                                                    06800000
       COMMIT;                                                                  06850000
       GRANT ALL ON                                                             06900000
              OAMADMIN.CBR_MGT_CLASS_TBL                                        06950000
            TO auth_id;                                                         07000000
       GRANT ALL ON                                                             07050000
              OAMADMIN.CBR_STO_CLASS_TBL                                        07100000
            TO auth_id;                                                         07150000
       GRANT ALL ON                                                             07200000
              OAMADMIN.CBR_COLLECTION_TBL                                       07250000
            TO auth_id;                                                         07300000
       COMMIT;                                                                  07350000
```

CBRISQLY

SAMPLIB member CBRISQLY, as shown here, provides the DB2 definitions for part of the OAM administration database. You must modify and run the job successfully before you use OAM.

```
//CBRISQLY JOB  MSGLEVEL=(1,1),MSGCLASS=A                               00050000
//*************************************************************         00100000
//*                                                                     00150000
//*  $SEG(CBRISQLY) COMP(OSR) PROD(OAM):                                00200000
//*                                                                     00250000
//*  OAM DB2 Database Definition Job (for Administration                00300000
//*  Databases).                                                        00350000
//*                                                                     00350001
//*  Licensed Materials - Property of IBM                               00350002
//*  5650-ZOS                                                           00350003
//*  COPYRIGHT IBM CORP. 1989, 2017                                     00350004
//*                                                                     00400000
//*  This job will create additional unique indexes for                 00450000
//*  the OAM Administration Databases in DB2.                           00500000
//*                                                                     00550000
//*  Prior to executing this job you need to make the                   00600000
//*  following modifications:                                           00650000
//*                                                                     00700000
//*  1. Change the name in the DSN SYSTEM(db2) statement to        @L3A 00700001
//*     the name of the DB2 Subsystem in your installation.        @L3A 00700002
//*                                                                @L3A 00700003
//*     In a multiple OAM configuration, you will create a         @L3A 00700004
//*     unique job for each Object OAM instance and specify        @L3A 00700005
//*     the DB2 SSID for the DB2 subsystem associated with         @L3A 00700006
//*     the Object OAM instance.                                   @L3A 00700007
//*                                                                @L3A 00700008
//*  2. Change the data set name in the RUN statement              @L3A 00700009
//*     LIB('db2xx.RUNLIB.LOAD') phrase to the data set name used  @L3A 00700010
//*     in your installation for the DB2 RUNLIB.LOAD data set.     @L3A 00700012
//*                                                                @L3A 00700013
//*     In a multiple OAM configuration, you will create a         @L3A 00700014
//*     unique job for each Object OAM instance and specify        @L3A 00700015
//*     the DB2 RUNLIB.LOAD for the DB2 subsystem associated with  @L3A 00700016
//*     the Object OAM instance.                                   @L3A 00700017
//*                                                                @L3A 00700018
//*  3. Change the PLAN name (DSNTIAxx) in the RUN statement to    @L3A 00700019
//*     match your current DB2 version and release level.          @L3A 00700021
//*                                                                @L3A 00700022
//*  4. Change "cat_name" to the DB2 VCAT name used                @L3C 00750000
//*     for defining the VSAM data sets in CBRIALCY.                    00800000
//*                                                                     00850000
//*  CHANGE ACTIVITY:                                                   01400000
//*    $00=OY28892 320 890601 TUCJRL: Add unique indexes to             01450000
//*                                   those defined in                  01500000
//*                                   CBRISQLX                          01550000
//*    $L1=HDZ11D0 140 970331 TUCSPP: Specify TYPE 1 INDEX for          01566600
//*                                   DB2 4.1 or above level            01583200
//*    $L2=HDZ11G0 R13 001016 TUCLJT: Removed reference to type 1       01586500
//*                                   indexes, no longer supported      01589800
//*                                   by DB2 and removed SUBPAGES       01593100
//*                                   from CREATE statements            01596400
//*    $L3=OAMR23M R23 160405 TUCAED: Multi OAM AS Support         @L3A 01596401
```

```
//*                                                                     01600000
//*****************************************************************     01650000
//CREATE EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)                      01700000
//SYSTSPRT DD SYSOUT=*                                                  01750000
//SYSTSIN  DD *                                                         01800000
 DSN SYSTEM(db2)                                                        01850000
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIAxx) -                                 01900000
      LIB('db2xx.RUNLIB.LOAD')                                          01950000
//SYSPRINT DD SYSOUT=*                                                  02000000
//SYSUDUMP DD SYSOUT=*                                                  02050000
//SYSIN    DD *                                                         02100000
          CREATE UNIQUE INDEX OAMADMIN.CBRMGTY                          02150000
                 ON OAMADMIN.CBR_MGT_CLASS_TBL                          02200000
                 (                                                      02250000
                  ODMCNAME ASC                                          02300000
                 )                                                      02350000
                 USING      VCAT cat_name                               02400000
                 CLOSE      NO                                          02450000
                 BUFFERPOOL BP1                                         02550000
                 PCTFREE    10;                                         02600000
          CREATE UNIQUE INDEX OAMADMIN.CBRSTOY                          02650000
                 ON OAMADMIN.CBR_STO_CLASS_TBL                          02700000
                 (                                                      02750000
                  ODSCNAME ASC                                          02800000
                 )                                                      02850000
                 USING      VCAT cat_name                               02900000
                 CLOSE      NO                                          02950000
                 BUFFERPOOL BP1                                         03050000
                 PCTFREE    10;                                         03100000
          COMMIT;                                                       03150000
```

OAM configuration database

CBRSAMPL installs the OAM configuration database in a new installation. For an example of this SAMPLIB member, see “CBRSAMPL” on page 438.

Sample migration jobs

These SAMPLIB members help you migrate to the current release of OAM:

- **CBRSMERG** merges two OAM configuration databases, while **CBRSG100** performs a database merge of two OAM object storage and administration databases. Both of these jobs are executed to allow DB2 data sharing in an OAMplex. For examples of these SAMPLIB members, see “CBRSMERG” on page 448 and “CBRSG100” on page 453.
- **CBRSMR1B** adds three new columns, ODSTATF, ODRETDT and ODINSTID to the object directory table. For an example of this SAMPLIB member, see “CBRSMR1B” on page 445.
- **CBRSMR1D** adds the File System Delete Table to the OAM Configuration Database. For an example of this SAMPLIB member, see “CBRSMR1D” on page 446.

CBRSAMPL

SAMPLIB member CBRSAMPL, shown here, creates the OAM configuration database.

```
//CBRSAMPL JOB MSGLEVEL=(1,1),MSGCLASS=A                                00050000
//********************************************************************** 00100000
//*                                                                     00150000
//*  $SEG(CBRSAMPL) COMP(DBM) PROD(OAM):                                00151900
//*                                                                     00153800
//*  OAM DB2 Database Definition Job (for OAM Configuration             00155700
//*  Database).                                                         00157600
//*                                                                     00157601
//*  Licensed Materials - Property of IBM                               00157602
//*  5650-ZOS                                                           00157603
//*  COPYRIGHT IBM CORP. 1989, 2017                                     00157604
//*                                                                     00159500
//*  This job will create the LCS databases, tables, and indexes        00161400
//*  in DB2 for the OAM Configuration Database.                         00163300
//*                                                                     00165200
//*  Before running this job, you must change the following:            00167100
//*                                                                     00169000
//*  1. Change the name in the DSN SYSTEM(db2) statement to        @LGA 00169001
```

```
//*     the name of the DB2 Subsystem in your installation.        @LGA 00169002
//*                                                                @LGA 00169003
//*     In a multiple OAM configuration, you will create a         @LGA 00169004
//*     unique job for each Object OAM instance and specify        @LGA 00169005
//*     the DB2 SSID for the DB2 subsystem associated with         @LGA 00169006
//*     the Object OAM instance.                                   @LGA 00169007
//*                                                                @LGA 00169008
//*  2. Change the data set name in the RUN statement              @LGA 00169009
//*     LIB('db2xx.RUNLIB.LOAD') phrase to the data set name used  @P2C 00169010
//*     in your installation for the DB2 RUNLIB.LOAD data set.     @LGA 00169011
//*                                                                @LGA 00169012
//*     In a multiple OAM configuration, you will create a         @LGA 00169013
//*     unique job for each Object OAM instance and specify        @LGA 00169014
//*     the DB2 RUNLIB.LOAD for the DB2 subsystem associated with  @LGA 00169015
//*     the Object OAM instance.                                   @LGA 00169016
//*                                                                @LGA 00169017
//*  3. Change the PLAN name (DSNTIAxx) in the RUN statement to    @LGA 00169018
//*     match your current DB2 version and release level.          @LGA 00169019
//*                                                                @LGA 00169020
//*  4. Change "dbpack" to the volume serials that your            @LGC 00170900
//*     target database should reside on.                               00172800
//*                                                                     00174700
//*  5. Change "cat_name" to the name of the catalog you           @LGC 00176600
//*     will be using under DB2.                                        00178500
//*                                                                     00180400
//*  6. Change "pass_word" to the name of the catalog              @LGC 00182300
//*     password.                                                       00184200
//*                                                                     00205100
//*  CHANGE ACTIVITY:                                                   00207000
//*    $L0=JDP3227 320 890601 TUCJRL: Initial Release                   00208900
//*    $L1=JDP3331 331 900815 TUCJRL: Added the deleted objects         00210800
//*                                   table. Changed NUMSLOTS and       00212700
//*                                   NUMESLOT to INTEGER in            00214600
//*                                   OLIBRARY table. Added DEVTYPE     00216500
//*                                   and LIBRDES to OLIBRARY table.    00218400
//*                                   Added DEVTYPE and DRIVDES to      00220300
//*                                   DRIVE table. Added FRESPACE,      00222200
//*                                   DELSPACE, DELCOUNT, FRAGIDX,      00224100
//*                                   MEDIATYP, CREDATE, ERRSTAT,       00226000
//*                                   VOLEMPTY, and RECOUNT to          00227900
//*                                   VOLUME table.                     00229800
//*    $P1=KBI0238 331 900904 TUCKHB: CHANGED LABEL FOR VOLUMSET        00231700
//*                                   TO STORAGE_GROUP.                 00233600
//*                                                                     00235500
//*    $L2=CAPELLA2 120 921203 TUCGRD: ADDED CLIBRARY, MEDIATYP AND     00237400
//*                                   LIBINDEX TO OLIBRARY TABLE.       00239300
//*                                   ADDED CAPACITY TO VOLUME TABLE    00241200
//*                                                                     00243100
//*    $L3=FRBTAPE  332 920220 TUCKHB: ADDED TAPEVOL TABLE FOR          00245000
//*                                   FRB TAPE SUPPORT.                 00246900
//*                                                                     00247400
//*    $L4=OBJTAPE  120 930523 TUCKMF: UPDATED TAPEVOL STEPS TO ADD     00247900
//*                                   NEW 120 COLUMNS FOR LOGICAL       00248400
//*                                   KB DELETED AND TAPE COMPACTION    00248900
//*                                   INDICATOR.                        00249400
//*    $L5=OPTSPE   130 950823 TUCSMC: Added new DRIVENUM to the        00249600
//*                                    DRIVE table.                     00249800
//*    $L6=HDZ11D0  140 970331 TUCSPP: Specify TYPE 1 INDEX for         00262300
//*                                    DB2 4.1 or above level           00274800
//*    $L7=HDZ11E0  150 970812 TUCLJT: ADDED MEMBER TO OLIBRARY TABLE,  00276800
//*                                    DRIVE TABLE, VOLUME TABLE, AND   00278800
//*                                    TAPEVOL TABLE                    00280800
//*                                    ADDED PLIBRARY TO OLIBRARY TABLE 00282800
//*                                    AND VOLUME TABLE                 00284800
//*    $L8=OW38975  1E0 990609 TUCSPP: ADDED NEW EPI FIELD TO THE       00285600
//*                                    TAPE VOLUME TABLE                00286400
//*    $L9=HDZ11G0  R13 000915 TUCLJT: ADDED BKTYPE TO VOLUME TABLE AND 00286700
//*                                    TAPEVOL TABLE                    00287000
//*    $LA=HDZ11H0  R15 011207 TUCBLC: ADDED OUNITNAM AND DATACLAS TO   00287100
//*                                    TAPEVOL TABLE                    00287200
//*    $LB=WRM3592  1J0 042904 TUCLJS: ADDED DSNFMT TO TAPEVOL TABLE    00290400
//*                                     (OA07105)                       00293600
//*    $LC=OAMR19   R19 060310 TUCBJF: ADDED SUBLEVEL TO TAPEVOL TABLE  00295200
//*    $LD=35923E   1K0 070914 TUCBJF: Adding KB overflow columns to    00295500
//*                                    TAPEVOL table:                   00295800
//*                                    (CAPACITYO, FRESPACEO,           00296100
//*                                    NUMLKBWO, NUMPKBWO, NUMLKBDEO)   00296400
//*    $LE=HYDRA16  190 043009 TUCPSS: ADDED VOLATTRF TO TAPEVOL TABLE  00296600
//*    $LF=OAMR1D   R1D 100223 TUCDEW: Added FSDELETE Table             00296700
//*    $LG=OAMR23M  R23 160405 TUCAED: Multi OAM AS Support        @LGA 00296701
//*    $P2=130050   R23 160817 TUCAED: Typo in db2xx.runlib        @P2A 00296702
//*                                                                     00296800
```

```
//********************************************************************* 00300000
//OCDBTABS EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)                    00350000
//SYSTSPRT DD SYSOUT=*                                                  00400000
//SYSTSIN  DD *                                                         00450000
 DSN SYSTEM(db2)                                                        00500000
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIAxx) -                                 00550000
      LIB('db2xx.RUNLIB.LOAD')                                          00600000
//SYSPRINT DD SYSOUT=*                                                  00650000
//SYSUDUMP DD SYSOUT=*                                                  00700000
//SYSIN    DD *                                                         00750000
  CREATE STOGROUP CBROAM                                                00800000
      VOLUMES (dbpack)                                                  00850000
      VCAT cat_name                                                     00900000
      PASSWORD pass_word;                                               00950000
                                                                        01000000
  CREATE DATABASE CBROAM                                                01050000
      STOGROUP CBROAM                                                   01100000
      BUFFERPOOL BP0;                                                   01150000
                                                                        01200000
  CREATE TABLESPACE OCLIBTSP                                            01250000
     IN CBROAM                                                          01300000
     USING STOGROUP CBROAM                                              01350000
     BUFFERPOOL BP0                                                     01400000
     CLOSE NO;                                                          01450000
                                                                        01500000
  CREATE TABLESPACE OCDRVTSP                                            01550000
     IN CBROAM                                                          01600000
     USING STOGROUP CBROAM                                              01650000
     BUFFERPOOL BP0                                                     01700000
     CLOSE NO;                                                          01750000
                                                                        01800000
  CREATE TABLESPACE OCSLTTSP                                            01850000
     IN CBROAM                                                          01900000
     USING STOGROUP CBROAM                                              01950000
     BUFFERPOOL BP0                                                     02000000
     CLOSE NO;                                                          02050000
                                                                        02100000
  CREATE TABLESPACE OCVOLTSP                                            02150000
     IN CBROAM                                                          02200000
     USING STOGROUP CBROAM                                              02250000
     BUFFERPOOL BP0                                                     02300000
     CLOSE NO;                                                          02350000
                                                                        02400000
  CREATE TABLESPACE OCDELTSP                                            02450000
     IN CBROAM                                                          02500000
     USING STOGROUP CBROAM                                              02550000
     BUFFERPOOL BP0                                                     02600000
     CLOSE NO;                                                          02650000
                                                                        02700000
  CREATE TABLESPACE OCTVLTSP                                            02707100
     IN CBROAM                                                          02708100
     USING STOGROUP CBROAM                                              02709100
     BUFFERPOOL BP0                                                     02710100
     CLOSE NO;                                                          02711100
                                                                        02712100
  CREATE TABLESPACE OCFSDTSP                                            02713100
     IN CBROAM                                                          02714200
     USING STOGROUP CBROAM                                              02721300
     BUFFERPOOL BP0                                                     02728400
     CLOSE NO;                                                          02735500
                                                                        02742600
  CREATE TABLE OLIBRARY                                                 02750000
    (NAME      CHAR(8)    NOT NULL,                                     02800000
     ONLINE    CHAR(1)    NOT NULL,                                     02850000
     OPERATNL  CHAR(1)    NOT NULL,                                     02900000
     PATHSTAT  CHAR(1)    NOT NULL,                                     02950000
     COMMAND   CHAR(5)    NOT NULL WITH DEFAULT,                        03000000
     PRIMCTC   CHAR(4)    NOT NULL WITH DEFAULT,                        03050000
     PRIMPORT  CHAR(1)    NOT NULL WITH DEFAULT,                        03100000
     ALTCTC    CHAR(4)    NOT NULL WITH DEFAULT,                        03150000
     ALTPORT   CHAR(1)    NOT NULL WITH DEFAULT,                        03200000
     FAULT     CHAR(3)    NOT NULL WITH DEFAULT,                        03250000
     OLIBTYPE  CHAR(1)    NOT NULL,                                     03300000
     NUMSLOTS  INTEGER    NOT NULL WITH DEFAULT,                        03350000
     NUMESLOT  INTEGER    NOT NULL WITH DEFAULT,                        03400000
     NUMDRVS   SMALLINT   NOT NULL,                                     03450000
     RCOMMAND  CHAR(5)    NOT NULL WITH DEFAULT,                        03464200
     DEVTYPE   CHAR(8)    NOT NULL WITH DEFAULT,                        03478400
     LIBRDES   CHAR(120)  NOT NULL WITH DEFAULT,                        03492600
     CLIBRARY  CHAR(8)    NOT NULL WITH DEFAULT,                        03506800
     MEDIATYP  CHAR(8)    NOT NULL WITH DEFAULT,                        03521000
     LIBINDEX  SMALLINT   NOT NULL WITH DEFAULT,                        03528200
```

```
     MEMBER    CHAR(16)   NOT NULL WITH DEFAULT,                            03535400
     PLIBRARY  CHAR(8)    NOT NULL WITH DEFAULT)                            03542600
    IN CBROAM.OCLIBTSP;                                                     03550000
                                                                            03600000
  CREATE TABLE DRIVE                                                        03650000
    (NAME      CHAR(8)    NOT NULL,                                         03700000
     OLIBRARY  CHAR(8)    NOT NULL,                                         03750000
     CTC       CHAR(4)    NOT NULL,                                         03800000
     SCSI      CHAR(1)    NOT NULL,                                         03850000
     LUN       CHAR(1)    NOT NULL,                                         03900000
     ONLINE    CHAR(1)    NOT NULL,                                         03950000
     OPERATNL  CHAR(1)    NOT NULL,                                         04000000
     LDRIVENO  CHAR(1)    NOT NULL WITH DEFAULT,                            04050000
     DRIVTYPE  CHAR(1)    NOT NULL,                                         04075000
     DEVTYPE   CHAR(8)    NOT NULL WITH DEFAULT,                            04100000
     DRIVDES   CHAR(120)  NOT NULL WITH DEFAULT,                            04116600
     DRIVENUM  SMALLINT   NOT NULL WITH DEFAULT,                            04127700
     MEMBER    CHAR(16)   NOT NULL WITH DEFAULT)                            04138800
    IN CBROAM.OCDRVTSP;                                                     04150000
                                                                            04200000
  CREATE TABLE SLOT                                                         04250000
    (NAME      CHAR(3)    NOT NULL,                                         04300000
     OLIBRARY  CHAR(8)    NOT NULL,                                         04350000
     OCCUPIED  CHAR(1)    NOT NULL,                                         04400000
     OPERATNL  CHAR(1)    NOT NULL,                                         04450000
     VOLSER0   CHAR(6)    NOT NULL WITH DEFAULT,                            04500000
     VOLSER1   CHAR(6)    NOT NULL WITH DEFAULT)                            04550000
    IN CBROAM.OCSLTTSP;                                                     04600000
                                                                            05250000
  CREATE TABLE VOLUME                                                       05300000
    (VOLSER    CHAR(6)    NOT NULL,                                         05350000
     OVOLSER   CHAR(6)    NOT NULL,                                         05400000
     LOCATION  CHAR(1)    NOT NULL,                                         05450000
     SLOT      CHAR(3)    NOT NULL,                                         05500000
     OLIBRARY  CHAR(8)    NOT NULL,                                         05550000
     SHELFLOC  CHAR(32)   NOT NULL WITH DEFAULT,                            05600000
     MNTDATE   DATE       NOT NULL WITH DEFAULT,                            05650000
     WRTDATE   DATE       NOT NULL WITH DEFAULT,                            05700000
     EXPDATE   DATE       NOT NULL WITH DEFAULT,                            05750000
     EJECTDAT  DATE       NOT NULL WITH DEFAULT,                            05800000
     LASTDATA  INTEGER    NOT NULL,                                         05850000
     LASTVTCL  INTEGER    NOT NULL,                                         05900000
     LASTVTCP  INTEGER    NOT NULL,                                         05950000
     VOLUMSET  CHAR(8)    NOT NULL,                                         06000000
     TYPE      CHAR(1)    NOT NULL,                                         06050000
     ORIENT    CHAR(1)    NOT NULL,                                         06100000
     FULL      CHAR(1)    NOT NULL,                                         06150000
     READABLE  CHAR(1)    NOT NULL,                                         06200000
     WRITABLE  CHAR(1)    NOT NULL,                                         06250000
     WRTPROT   CHAR(1)    NOT NULL,                                         06257100
     OWNERP    CHAR(1)    NOT NULL WITH DEFAULT,                            06264200
     OWNER     CHAR(32)   NOT NULL WITH DEFAULT,                            06271300
     FRESPACE  INTEGER    NOT NULL WITH DEFAULT,                            06278400
     DELSPACE  INTEGER    NOT NULL WITH DEFAULT,                            06285500
     DELCOUNT  INTEGER    NOT NULL WITH DEFAULT,                            06292600
     FRAGIDX   SMALLINT   NOT NULL WITH DEFAULT,                            06299700
     MEDIATYP  CHAR(2)    NOT NULL WITH DEFAULT,                            06306800
     CREDATE   DATE       NOT NULL WITH DEFAULT,                            06313900
     ERRSTAT   SMALLINT   NOT NULL WITH DEFAULT,                            06321000
     VOLEMPTY  CHAR(1)    NOT NULL WITH DEFAULT,                            06328100
     RECOUNT   SMALLINT   NOT NULL WITH DEFAULT,                            06335200
     CAPACITY  INTEGER    NOT NULL WITH DEFAULT,                            06338900
     MEMBER    CHAR(16)   NOT NULL WITH DEFAULT,                            06340700
     PLIBRARY  CHAR(8)    NOT NULL WITH DEFAULT,                            06342600
     BKTYPE    CHAR(1)    NOT NULL WITH DEFAULT)                            06346300
    IN CBROAM.OCVOLTSP;                                                     06350000
                                                                            06351400
  CREATE TABLE DELOBJT                                                      06352800
    (COLNAME   CHAR(44)   NOT NULL,                                         06354200
     OBJNAME   CHAR(44)   NOT NULL,                                         06355600
     VOLSER    CHAR(6)    NOT NULL,                                         06357000
     VTOCTOKN  INTEGER    NOT NULL,                                         06358400
     OBJSIZE   INTEGER    NOT NULL)                                         06359800
    IN CBROAM.OCDELTSP;                                                     06361200
                                                                            06362600
  CREATE TABLE TAPEVOL                                                      06364000
    (VOLSER    CHAR(6)    NOT NULL,                                         06365400
     UNITNAME  CHAR(8)    NOT NULL,                                         06366800
     MEDIATYP  CHAR(2)    NOT NULL,                                         06368200
     STORGRP   CHAR(8)    NOT NULL,                                         06369600
     TYPE      CHAR(1)    NOT NULL,                                         06371000
     CREDATE   DATE       NOT NULL,                                         06372400
```

```
      MNTDATE   DATE       NOT NULL,                                     06373800
      WRTDATE   DATE       NOT NULL,                                     06375200
      EXPDATE   DATE       NOT NULL,                                     06376600
      CAPACITY  INTEGER    NOT NULL,                                     06378000
      FRESPACE  INTEGER    NOT NULL,                                     06379400
      LSTBLKID  INTEGER    NOT NULL,                                     06380800
      PFULL     SMALLINT   NOT NULL,                                     06382200
      NUMLBLKS  INTEGER    NOT NULL,                                     06383600
      NUMLKBW   INTEGER    NOT NULL,                                     06385000
      NUMPKBW   INTEGER    NOT NULL,                                     06386400
      NUMLKBDE  INTEGER    NOT NULL,                                     06387100
      FULL      CHAR(1)    NOT NULL,                                     06387800
      READABLE  CHAR(1)    NOT NULL,                                     06389200
      WRITABLE  CHAR(1)    NOT NULL,                                     06390600
      INUSE     CHAR(1)    NOT NULL,                                     06392000
      COPIED    CHAR(1)    NOT NULL,                                     06393400
      AVOLSER   CHAR(6)    NOT NULL,                                     06394300
      COMPACT   CHAR(1)    NOT NULL,                                     06394900
      EPI       SMALLINT   NOT NULL WITH DEFAULT,                        06395200
      MEMBER    CHAR(16)   NOT NULL WITH DEFAULT,                        06395500
      BKTYPE    CHAR(1)    NOT NULL WITH DEFAULT,                        06395600
      OUNITNAM  CHAR(8)    NOT NULL WITH DEFAULT,                        06395700
      DATACLAS  CHAR(8)    NOT NULL WITH DEFAULT,                        06395800
      DSNFMT    CHAR(1)    NOT NULL WITH DEFAULT,                        06395900
      SUBLEVEL  CHAR(1)    NOT NULL WITH DEFAULT,                        06396400
      CAPACITYO INTEGER    NOT NULL WITH DEFAULT,                        06396900
      FRESPACEO INTEGER    NOT NULL WITH DEFAULT,                        06397400
      NUMLKBWO  INTEGER    NOT NULL WITH DEFAULT,                        06397900
      NUMPKBWO  INTEGER    NOT NULL WITH DEFAULT,                        06398400
      NUMLKBDEO INTEGER    NOT NULL WITH DEFAULT,                        06398700
      VOLATTRF  SMALLINT   NOT NULL WITH DEFAULT)                        06399000
     IN CBROAM.OCTVLTSP;                                                 06399400
                                                                         06403600
  CREATE TABLE FSDELETE                                                  06407800
    (ID       INTEGER PRIMARY KEY GENERATED ALWAYS AS IDENTITY           06409900
                      (START WITH 1, INCREMENT BY 1, CACHE 50, CYCLE),   06412000
     DELTIME  INTEGER      NOT NULL WITH DEFAULT,                        06414100
     DELREAS  CHAR(1)      NOT NULL WITH DEFAULT,                        06416200
     SYSNAME  CHAR(8)      NOT NULL WITH DEFAULT,                        06420400
     SGNAME   CHAR(8)      NOT NULL WITH DEFAULT,                        06424600
     COLNAME  CHAR(44)     NOT NULL WITH DEFAULT,                        06428800
     OBJNAME  CHAR(44)     NOT NULL WITH DEFAULT,                        06433000
     INSTID   INTEGER      NOT NULL WITH DEFAULT)                        06437200
    IN CBROAM.OCFSDTSP;                                                  06441400
                                                                         06445600
  CREATE UNIQUE INDEX LNAMINDX                                           06450000
    ON OLIBRARY                                                          06500000
    (NAME ASC)                                                           06550000
    USING STOGROUP CBROAM                                                06600000
    CLUSTER                                                              06650000
    BUFFERPOOL BP0                                                       06700000
    CLOSE NO;                                                            06750000
                                                                         06800000
  CREATE UNIQUE INDEX DNAMINDX                                           06850000
    ON DRIVE                                                             06900000
    (NAME ASC)                                                           06950000
    USING STOGROUP CBROAM                                                07000000
    BUFFERPOOL BP0                                                       07050000
    CLOSE NO;                                                            07100000
                                                                         07150000
  CREATE UNIQUE INDEX DRIDINDX                                           07200000
    ON DRIVE                                                             07250000
    (CTC, SCSI, LUN ASC)                                                 07300000
    USING STOGROUP CBROAM                                                07350000
    BUFFERPOOL BP0                                                       07400000
    CLUSTER                                                              07450000
    CLOSE NO;                                                            07500000
                                                                         07550000
  CREATE UNIQUE INDEX SLIBINDX                                           07600000
    ON SLOT                                                              07650000
    (NAME, OLIBRARY ASC)                                                 07700000
    USING STOGROUP CBROAM                                                07750000
    BUFFERPOOL BP0                                                       07800000
    CLUSTER                                                              07850000
    CLOSE NO;                                                            07900000
                                                                         07950000
  CREATE UNIQUE INDEX VSERINDX                                           07960500
    ON VOLUME                                                            07971000
    (VOLSER ASC)                                                         07981500
    USING STOGROUP CBROAM                                                07992000
    BUFFERPOOL BP0                                                       08002500
    CLUSTER                                                              08013000
```

```
  CLOSE NO;

CREATE INDEX DVOLINDX
  ON DELOBJT
  (VOLSER ASC)
  USING STOGROUP CBROAM
  BUFFERPOOL BP0
  CLOSE NO;

CREATE UNIQUE INDEX DELOINDX
  ON DELOBJT
  (COLNAME, OBJNAME, VOLSER, VTOCTOKN ASC)
  USING STOGROUP CBROAM
  BUFFERPOOL BP0
  CLUSTER
  CLOSE NO;

CREATE UNIQUE INDEX TVOLINDX
  ON TAPEVOL
  (VOLSER ASC)
  USING STOGROUP CBROAM
  BUFFERPOOL BP0
  CLUSTER
  CLOSE NO;

CREATE UNIQUE INDEX FSDTINDX
  ON FSDELETE
  (ID ASC)
  USING STOGROUP CBROAM
  BUFFERPOOL BP0
  CLUSTER
  CLOSE NO;

LABEL ON OLIBRARY
  (NAME IS 'NAME',
   ONLINE IS 'ONLINE',
   OPERATNL IS 'OPERATIONAL',
   PATHSTAT IS 'CURRENT_PATH',
   COMMAND IS 'CURRENT_COMMAND',
   PRIMCTC IS 'PRIMARY_CTC',
   PRIMPORT IS 'PRIMARY_PORT',
   ALTCTC IS 'ALTERNATE_CTC',
   ALTPORT IS 'ALTERNATE_PORT',
   FAULT IS 'FAULT_CODE',
   OLIBTYPE IS 'LIBRARY_TYPE',
   NUMSLOTS IS 'SLOTS',
   NUMESLOT IS 'EMPTY_SLOTS',
   NUMDRVS IS 'DRIVES',
   RCOMMAND IS 'RECOVERY_COMMAND',
   DEVTYPE IS 'DEVICE_TYPE',
   LIBRDES IS 'LIBRARY_DESCRIPTION',
   CLIBRARY IS 'CONTROLLING_LIBRARY',
   MEDIATYP IS 'DEFAULT_MEDIA_TYPE',
   LIBINDEX IS 'LIBRARY_INDEX',
   MEMBER IS 'OAM_XCF_MEMBER',
   PLIBRARY IS 'DEFAULT_PSEUDO_LIBRARY');

LABEL ON DRIVE
  (NAME IS 'NAME',
   OLIBRARY IS 'OLIBRARY',
   CTC IS 'CTC',
   SCSI IS 'SCSI',
   LUN IS 'LUN',
   ONLINE IS 'ONLINE',
   OPERATNL IS 'OPERATIONAL',
   LDRIVENO IS 'DRIVE_NUMBER',
   DRIVTYPE IS 'DRIVE_TYPE',
   DEVTYPE IS 'DEVICE_TYPE',
   DRIVDES IS 'DRIVE_DESCRIPTION',
   DRIVENUM IS 'PHYS_DRIVE_NUMBER',
   MEMBER IS 'OAM_XCF_MEMBER');

LABEL ON SLOT
  (NAME IS 'NAME',
   OLIBRARY IS 'OLIBRARY',
   OCCUPIED IS 'OCCUPIED',
   OPERATNL IS 'OPERATIONAL',
   VOLSER0 IS 'VOLSER0',
   VOLSER1 IS 'VOLSER1');

LABEL ON VOLUME
  (VOLSER IS 'VOLSER',
```

```
     OVOLSER IS 'OTHER_VOLSER',                                          11250000
     LOCATION IS 'LOCATION',                                             11300000
     SLOT IS 'SLOT',                                                     11350000
     OLIBRARY IS 'OLIBRARY',                                             11400000
     SHELFLOC IS 'SHELF_LOCATION',                                       11450000
     MNTDATE IS 'DATE_LAST_MOUNTED',                                     11500000
     WRTDATE IS 'DATE_LAST_WRITTEN',                                     11550000
     EXPDATE IS 'EXPIRATION_DATE',                                       11600000
     EJECTDAT IS 'EJECT/ENTER_DATE',                                     11650000
     LASTDATA IS 'LAST_DATA_SECTOR',                                     11700000
     LASTVTCL IS 'LAST_LOGICAL_VTOC_SECTOR',                             11750000
     LASTVTCP IS 'LAST_PHYSICAL_VTOC_SECTOR',                            11800000
     VOLUMSET IS 'STORAGE_GROUP',                                        11850000
     TYPE IS 'TYPE',                                                     11900000
     ORIENT IS 'ORIENTATION',                                            11950000
     FULL IS 'FULL',                                                     12000000
     READABLE IS 'VOLUME_READABLE_STATUS',                               12004200
     WRITABLE IS 'VOLUME_WRITABLE_STATUS',                               12008400
     WRTPROT IS 'WRITE_PROTECTED',                                       12012600
     OWNERP IS 'OWNER_INFORMATION_POSITION',                             12016800
     OWNER IS 'OWNER_INFORMATION',                                       12021000
     FRESPACE IS 'FREE_SPACE',                                           12025200
     DELSPACE IS 'DELETED_SPACE',                                        12029400
     DELCOUNT IS 'DELETED_OBJECTS',                                      12033600
     FRAGIDX IS 'FRAGMENTATION_INDEX',                                   12037800
     MEDIATYP IS 'MEDIA_TYPE',                                           12042000
     CREDATE IS 'CREATE_DATE',                                           12046200
     ERRSTAT IS 'VOLUME_ERROR_STATUS',                                   12050400
     VOLEMPTY IS 'VOLUME_EMPTY',                                         12054600
     RECOUNT IS 'DELETED_OBJECTS_RECOUNT',                               12058800
     CAPACITY IS 'CAPACITY',                                             12060900
     MEMBER IS 'OAM_XCF_MEMBER',                                         12061900
     PLIBRARY IS 'PSEUDO_LIBRARY_FOR_VOLUME',                            12063000
     BKTYPE IS 'BACKUP_TYPE');                                           12065100
                                                                         12067200
  LABEL ON DELOBJT                                                       12071400
    (COLNAME IS 'COLLECTION_NAME',                                       12075600
     OBJNAME IS 'OBJECT_NAME',                                           12079800
     VOLSER IS 'VOLSER',                                                 12084000
     VTOCTOKN IS 'VTOC_TOKEN',                                           12088200
     OBJSIZE IS 'OBJECT_SIZE');                                          12092400
                                                                         12096600
  LABEL ON TAPEVOL                                                       12100800
    (VOLSER IS 'VOLSER',                                                 12105000
     UNITNAME IS 'UNIT_NAME',                                            12109200
     MEDIATYP IS 'MEDIA_TYPE',                                           12113400
     STORGRP IS 'STORAGE_GROUP',                                         12117600
     TYPE IS 'TYPE',                                                     12121800
     CREDATE IS 'CREATION_DATE',                                         12126000
     MNTDATE IS 'DATE_LAST_MOUNTED',                                     12130200
     WRTDATE IS 'DATE_LAST_WRITTEN',                                     12134400
     EXPDATE IS 'EXPIRATION_DATE',                                       12138600
     CAPACITY IS 'CAPACITY',                                             12142800
     FRESPACE IS 'FREE_SPACE',                                           12147000
     LSTBLKID IS 'LAST_BLOCKID',                                         12151200
     PFULL IS 'PERCENT_FULL',                                            12155400
     NUMLBLKS IS 'LOGICAL_BLOCKS_WRITTEN',                               12159600
     NUMLKBW IS 'LOGICAL_KILOBYTES_WRITTEN',                             12163800
     NUMPKBW IS 'PHYSICAL_KILOBYTES_WRITTEN',                            12168000
     NUMLKBDE IS 'LOGICAL_KILOBYTES_DELETED',                            12170100
     FULL IS 'FULL',                                                     12172200
     READABLE IS 'VOLUME_READABLE_STATUS',                               12176400
     WRITABLE IS 'VOLUME_WRITABLE_STATUS',                               12180600
     INUSE IS 'IN_USE',                                                  12184800
     COPIED IS 'COPIED',                                                 12189000
     AVOLSER IS 'ALTERNATE_VOLUME',                                      12192600
     COMPACT IS 'TAPE_COMPACTION_INDICATOR',                             12195000
     EPI      IS 'EPI',                                                  12196200
     MEMBER IS 'OAM_XCF_MEMBER',                                         12197400
     BKTYPE IS 'BACKUP_TYPE',                                            12198000
     OUNITNAM IS 'ORIGINAL_UNIT_NAME',                                   12198600
     DATACLAS IS 'DATACLASS',                                            12199000
     DSNFMT IS 'DATASET_NAME_FORMAT',                                    12199300
     SUBLEVEL IS 'TAPE_SUBLEVEL',                                        12199400
     CAPACITYO IS 'CAPACITY_OVERFLOW',                                   12199500
     FRESPACEO IS 'FREE_SPACE_OVERFLOW',                                 12199600
     NUMLKBWO  IS 'LOGICAL_KBS_WRITTEN_OVERFLOW',                        12199700
     NUMPKBWO  IS 'PHYSICAL_KBS_WRITTEN_OVERFLOW',                       12199800
     NUMLKBDEO IS 'LOGICAL_KBS_DELETED_OVERFLOW',                        12249800
     VOLATTRF  IS 'VOLUME_ATTRIBUTE_FLAGS');                             12299800
                                                                         12349800
  LABEL ON FSDELETE                                                      12399800
```

```
   (ID      IS 'ID',                                                    12433100
    DELTIME IS 'DELETE_TIME',                                           12466400
    DELREAS IS 'REASON',                                                12499800
    SYSNAME IS 'SYS_NAME',                                              12549800
    SGNAME  IS 'SG_NAME',                                               12599800
    OBJNAME IS 'OBJECT_NAME',                                           12649800
    COLNAME IS 'COLLECTION_NAME',                                       12699800
    INSTID  IS 'INSTANCE_ID');                                          12749800
                                                                        17500000
 COMMIT;                                                                17550000
/*                                                                      17600000
```

CBRSMR1B

The CBRSMR1B job, as shown here, adds three new columns, ODSTATF, ODRETDT and ODINSTID to the DB2 table.

```
//CBRSMR1B  JOB MSGLEVEL=(1,1),MSGCLASS=A                               00050000
//********************************************************************* 00100000
//*                                                                     00150000
//*  $SEG(SMR1B) COMP(DBM) PROD(OAM):                                   00200000
//*                                                                     00250000
//*  OAM DB2 Database Migration Job (for the Object Storage             00300000
//*  Databases).                                                        00350000
//*                                                                     00350001
//*  Licensed Materials - Property of IBM                               00350002
//*  5650-ZOS                                                           00350003
//*  COPYRIGHT IBM CORP. 2009                                           00350004
//*                                                                     00400000
//*  ---------------------------------------------------------------    00450000
//*  ---------------------------------------------------------------    00500000
//*                                                                     00550000
//*  SMR1B                                                              00600000
//*                                                                     00650000
//*  This job will perform the migration from the z/OS V1 R10           00700000
//*  version of the Object Storage Database to the z/OS V1 R11          00750000
//*  version which supports DB2 large objects and Archive               00783300
//*  Retention Enhancements.                                            00816600
//*                                                                     00850000
//*  This job will:                                                     00900000
//*   1. Add new columns ODSTATF, ODRETDT and ODINSTID to               00950000
//*      existing object directory table.                               01000000
//*                                                                     01050000
//*  It is recommended that you create a DB2 image copy of the          01100000
//*  existing hlq.OSM_OBJ_DIR table prior to executing                  01150000
//*  this migration job for recovery purposes.                          01200000
//*                                                                     01250000
//*  Before running this job, you must change the following:            01300000
//*                                                                     01350000
//*    1.Change the name in the DSN SYSTEM(DB2) statement to            01400000
//*      the name of the DB2 Subsystem in your installation.            01450000
//*                                                                     01500000
//*    2.Change the PLAN name (DSNTIA91) in the RUN statement to        01550000
//*      match your current DB2 version and release level.              01600000
//*                                                                     01650000
//*    3.Change the data set name in the RUN statement                  01700000
//*      LIB('DB2MINI.V9R1M0.RUNLIB.LOAD') phrase to the data set name  01750000
//*      used in your installation for the DB2 RUNLIB.LOAD data set.    01800000
//*                                                                     01850000
//*    4.Change the high level qualifier (hlq) in the ALTER             01900000
//*      statements of the table 'hlq.OSM_OBJ_DIR' to                   01950000
//*      match the database name the table lives in.                    02000000
//*                                                                     02050000
//*    5.Change the high level qualifier (hlq) in the DROP and CREATE   02100000
//*      VIEW statements for 'hlq.V_OSM_OBJ_DIR' and                    02137500
//*      'hlq.OSM_OBJ_DIR' to match the database                        02175000
//*      name the table lives in.                                       02212500
//*                                                                     02250000
//*    6. Add a new ALTER, DROP and CREATE statement for every          02300000
//*       object storage database.                                      02350000
//*                                                                     02400000
//*                                                                     02450000
//*  CHANGE ACTIVITY:                                                   02500000
//*    $L0=HDZ1B10  R1B 080627 TUCBLC: Initial Release                  02533300
//*    $P0=K1B0208  R1B 080806 TUCBLC: Default date to '0001-01-01'     02566600
//*    $P1=K1B0452  R1B 081002 TUCBLC: Prolog changes                   02583300
//*                                                                     02600000
//********************************************************************* 02650000
//********************************************************************* 02700000
```

```
//*  Alter the hlq.OSM_OBJ_DIR table to add the                         02750000
//*  ODSTATF, ODRETDT and ODINSTID column definitions.                  02800000
//******************************************************************** 02850000
//ALTERTAB EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)                    02900000
//SYSTSPRT DD SYSOUT=*                                                  02950000
//SYSTSIN  DD *                                                         03000000
 DSN SYSTEM(DB2)                                                        03050000
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIA91) -                                 03100000
      LIB('DB2MINI.V9R1M0.RUNLIB.LOAD')                                 03150000
//SYSPRINT DD SYSOUT=*                                                  03200000
//SYSUDUMP DD SYSOUT=*                                                  03250000
//SYSIN    DD *                                                         03300000
                                                                        03350000
 ALTER TABLE hlq.OSM_OBJ_DIR ADD ODSTATF                                03400000
             SMALLINT NOT NULL WITH DEFAULT;                            03450000
 ALTER TABLE hlq.OSM_OBJ_DIR ADD ODRETDT                                03500000
             DATE NOT NULL WITH DEFAULT '0001-01-01';                   03533300
 ALTER TABLE hlq.OSM_OBJ_DIR ADD ODINSTID                               03566600
             INTEGER NOT NULL WITH DEFAULT;                             03583200
                                                                        03600000
 COMMIT;                                                                03650000
/*                                                                      03700000
//******************************************************************** 03750000
//*  DROP and CREATE the hlq.V_OSM_OBJ_DIR table view                   03800000
//******************************************************************** 03850000
//REFRESHV EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)                    03900000
//SYSTSPRT DD SYSOUT=*                                                  03950000
//SYSTSIN  DD *                                                         04000000
 DSN SYSTEM(DB2)                                                        04050000
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIA91) -                                 04100000
      LIB('DB2MINI.V9R1M0.RUNLIB.LOAD')                                 04150000
//SYSPRINT DD SYSOUT=*                                                  04200000
//SYSUDUMP DD SYSOUT=*                                                  04250000
//SYSIN    DD *                                                         04300000
  DROP VIEW hlq.V_OSM_OBJ_DIR;                                          04350000
                                                                        04400000
  CREATE VIEW                                                           04450000
         hlq.V_OSM_OBJ_DIR                                              04500000
       AS SELECT ALL * FROM                                             04550000
         hlq.OSM_OBJ_DIR;                                               04600000
                                                                        04650000
 COMMIT;                                                                04700000
/*                                                                      04750000
```

CBRSMR1D

This job performs the migration from the z/OS V1R12 version of the OAM Configuration Database to the z/OS V1R13 version, which supports primary objects in the file system sublevel and introduces a new File System Delete Table.

```
//CBRSMR1D  JOB MSGLEVEL=(1,1),MSGCLASS=A                               00050000
//******************************************************************** 00100000
//*                                                                     00150000
//*  $SEG(SMR1D) COMP(DBM) PROD(OAM):                                   00200000
//*                                                                     00250000
//*  OAM DB2 Database Migration Job (for the Object Configuration       00300000
//*  Database).                                                         00350000
//*                                                                     00350001
//*  Licensed Materials - Property of IBM                               00350002
//*  5650-ZOS                                                           00350003
//*  COPYRIGHT IBM CORP. 2010, 2017                                     00350004
//*                                                                     00400000
//*  This job will perform the migration from the z/OS V1 R12           00700000
//*  version of the Object Configuration Database to the z/OS V1 R13    00750000
//*  version which supports storing objects on a file system.           00800000
//*                                                                     00850000
//*  This job will:                                                     00900000
//*    1. Add new tablespace OCFSDTSP                                   00950000
//*    2. Add new table FSDELETE in the new tablespace                  01000000
//*    3. Define labels for the columns in the new table                01050000
//*    4. Define an index for the new table                             01083300
//*    5. Grant all authority for the new table to CBROAM.              01116600
//*                                                                     01150000
//*  Before running this job, you must change the following:            01200000
//*                                                                     01250000
//*    1. Change the name in the DSN SYSTEM(db2) statement to      @L1C 01300000
//*       the name of the DB2 Subsystem in your installation.           01350000
//*                                                                     01400000
//*     In a multiple OAM configuration, you will create a         @L1A 01400001
```

```
//*     unique job for each Object OAM instance and specify        @L1A 01400002
//*     the DB2 SSID for the DB2 subsystem associated with         @L1A 01400003
//*     the Object OAM instance.                                   @L1A 01400004
//*                                                                @L1A 01400005
//*    2. Change the PLAN name (DSNTIAxx) in the RUN statement to  @L1C 01450000
//*       match your current DB2 version and release level.             01500000
//*                                                                     01550000
//*    3. Change the data set name in the RUN statement                 01600000
//*       LIB('db2xx.RUNLIB.LOAD') phrase to the data set name     @L1C 01650000
//*       used in your installation for the DB2 RUNLIB.LOAD data set.   01700000
//*                                                                     01750000
//*     In a multiple OAM configuration, you will create a         @L1A 01750001
//*     unique job for each Object OAM instance and specify        @L1A 01750002
//*     the DB2.RUNLIB.LOAD for the DB2 subsystem associated with  @L1A 01750003
//*     the Object OAM instance.                                   @L1A 01750004
//*                                                                @L1A 01750005
//*  CHANGE ACTIVITY:                                                   01800000
//*    $L0=HDZ1D10  R1D 100223 TUCDEW: Initial Release                  01850000
//*    $L1=OAMR23M  R23 160405 TUCAED: Multi OAM AS Support        @L1A 01850001
//*                                                                     01900000
//******************************************************************** 01950000
//CREATE   EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)                    02000000
//SYSTSPRT DD SYSOUT=*                                                  02050000
//SYSTSIN  DD *                                                         02100000
 DSN SYSTEM(db2)                                                        02150000
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIAxx) -                                 02200000
      LIB('db2xx.RUNLIB.LOAD')                                          02250000
//SYSPRINT DD SYSOUT=*                                                  02300000
//SYSUDUMP DD SYSOUT=*                                                  02350000
//SYSIN    DD *                                                         02400000
                                                                        02450000
 CREATE TABLESPACE OCFSDTSP                                             02500000
    IN CBROAM                                                           02550000
    USING STOGROUP CBROAM                                               02600000
    BUFFERPOOL BP0                                                      02650000
    CLOSE NO;                                                           02700000
                                                                        02750000
 CREATE TABLE FSDELETE                                                  02800000
   (ID       INTEGER PRIMARY KEY GENERATED ALWAYS AS IDENTITY           02825000
                     (START WITH 1, INCREMENT BY 1, CACHE 50, CYCLE),   02850000
    DELTIME  INTEGER      NOT NULL WITH DEFAULT,                        02875000
    DELREAS  CHAR(1)      NOT NULL WITH DEFAULT,                        02900000
    SYSNAME  CHAR(8)      NOT NULL WITH DEFAULT,                        02950000
    SGNAME   CHAR(8)      NOT NULL WITH DEFAULT,                        03000000
    COLNAME  CHAR(44)     NOT NULL WITH DEFAULT,                        03050000
    OBJNAME  CHAR(44)     NOT NULL WITH DEFAULT,                        03100000
    INSTID   INTEGER      NOT NULL WITH DEFAULT)                        03150000
   IN CBROAM.OCFSDTSP;                                                  03200000
                                                                        03250000
 CREATE UNIQUE INDEX FSDTINDX                                           03255500
   ON FSDELETE                                                          03261000
   (ID ASC)                                                             03266500
   USING STOGROUP CBROAM                                                03272000
   BUFFERPOOL BP0                                                       03277500
   CLUSTER                                                              03283000
   CLOSE NO;                                                            03288500
                                                                        03294000
 LABEL ON FSDELETE                                                      03300000
   (ID      IS 'ID',                                                    03333300
    DELTIME IS 'DELETE_TIME',                                           03366600
    DELREAS IS 'REASON',                                                03400000
    SYSNAME IS 'SYS_NAME',                                              03450000
    SGNAME  IS 'SG_NAME',                                               03500000
    OBJNAME IS 'OBJECT_NAME',                                           03550000
    COLNAME IS 'COLLECTION_NAME',                                       03600000
    INSTID  IS 'INSTANCE_ID');                                          03650000
                                                                        03700000
 COMMIT;                                                                03750000
/*                                                                      03800000
//******************************************************************** 03850000
//*  GRANT all authority for the new table to CBROAM                    03900000
//******************************************************************** 03950000
//GRANT    EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)                    04000000
//SYSTSPRT DD SYSOUT=*                                                  04050000
//SYSTSIN  DD *                                                         04100000
 DSN SYSTEM(db2)                                                        04150000
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIAxx) -                                 04200000
      LIB('db2xx.RUNLIB.LOAD')                                          04250000
//SYSPRINT DD SYSOUT=*                                                  04300000
//SYSUDUMP DD SYSOUT=*                                                  04350000
//SYSIN    DD *                                                         04400000
                                                                        04450000
```

```
 GRANT ALL ON                                                           04500000
      FSDELETE                                                          04550000
    TO CBROAM;                                                          04600000
                                                                        04650000
 COMMIT;                                                                04700000
/*                                                                      04750000
```

CBRSMERG

SAMPLIB member CBRSMERG, as shown here, assists you in merging the OAM configuration databases (OCDBs) for use with DB2 data sharing in an OAMplex. Running CBRSMERG might not be the only way of performing this task. Use a method that best suits the requirements for your environment. All warnings, prerequisites, or recommendations apply regardless of the method that is used to perform this merge. If you choose to use this sample job, you must modify and run the job successfully before you use OAM.

```
//CBRSMERG JOB MSGLEVEL=(1,1),MSGCLASS=A                                00050000
//********************************************************************* 00100000
//*                                                                     00150000
//*  $SEG(CBRSMERG) COMP(DBM) PROD(OAM):                                00200000
//*                                                                     00250000
//*  OAM DB2 Database Merge Job (combining optical configuration        00300000
//*  databases from multiple systems)                                   00350000
//*                                                                     00350001
//*  Licensed Materials - Property of IBM                               00350002
//*  5650-ZOS                                                           00350003
//*  COPYRIGHT IBM CORP. 1999, 2017                                     00350004
//*                                                                     00700000
//*  This job is intended to assist in the task of merging OAM          00700001
//*  configuration databases for use with DB2 data sharing in an        00750000
//*  OAMPLEX environment.                                               00800000
//*  DFSMS/MVS 1.5.0 provides OAM support in a parallel sysplex         00850000
//*  environment using XCF communications and DB2 data sharing.         00900000
//*                                                                     00950000
//*                                                                     01000000
//*  This job will:                                                     01050000
//*   1. Load data from other system (DB2 data from other               01100000
//*      system's DB2 tables) into this system's OAM's tables.          01150000
//*                                                                     01200000
//*******************************************************************   01250000
//* WARNING: You MUST obtain the following before proceeding:      @L1C 01350000
//*                                                                     01350001
//*   1. Load the information from the OAM configuration tables    @L1C 01400000
//*      from the other OAM/system where data is to be copied from      01450000
//*      onto a dataset on this system.                            @L1C 01500000
//*      The simplest way to do this is to do an SQL SELECT * from      01550000
//*      the OAM configuration database tables on the other systems     01600000
//*      and editing the column headers our of the resulting output     01650000
//*      so that just the data from the rows remains.                   01700000
//*                                                                @L1A 01750000
//*   2. Note the beginning and ending columns where the data      @L1C 01750001
//*      resides for each column in the table rows.                     01800000
//*                                                                     01850000
//*******************************************************************   01900000
//*                                                                     01950000
//*  It is recommended that you create a DB2 image copy of the          02000000
//*  existing tables:                                                   02050000
//*    OAM configuration database tables:                               02100000
//*       OLIBRARY                                                      02150000
//*       DRIVE                                                         02200000
//*       VOLUME                                                        02250000
//*       TAPEVOL                                                       02300000
//*       DELOBJT                                                       02350000
//*       SLOT                                                          02400000
//*  These tables will be modified directly by this job, so a           02450000
//*  backup copy is necessary in case recovery is needed.               02500000
//*                                                                     02550000
//*******************************************************************   02600000
//*                                                                     02650000
//*  Before running this job, you must change the following:            02700000
//*                                                                     02750000
//*    1.Change the PARM='db2' in the JOB STEP statements to       @L1C 02800000
//*      the name of the DB2 Subsystem in your installation.            02850000
//*                                                                     02900000
//*     In a multiple OAM configuration, you will create a         @L1A 02900001
//*     unique job for each Object OAM instance and specify        @L1A 02900002
//*     the DB2 SSID for the DB2 subsystem associated with         @L1A 02900003
//*     the Object OAM instance.                                   @L1A 02900004
//*                                                                @L1A 02900005
```

```
//*    2.Change the data set name SYS1.db2xx.VxRxM0.SDSNLOAD            02937500
//*      in the STEPLIB statements to the data set name used            02975000
//*      for the DB2 SDSNLOAD dataset, if necessary.                    03012500
//*                                                                     03050000
//*     In a multiple OAM configuration, you will create a         @L1A 03050001
//*     unique job for each Object OAM instance and specify        @L1A 03050002
//*     the SYS1.db2xx.VXRXM0.SDSNLOAD for the DB2 subsystem       @L1A 03050003
//*     associated with the Object OAM instance.                   @L1A 03050004
//*                                                                @L1A 03050005
//*    3. Change "dbpack" to the volume serials that your          @L1A 03050006
//*     target database should reside on.                          @L1A 03050007
//*                                                                @L1A 03050008
//*    4.Change RESUME YES to RESUME NO if you are loading         @L1C 03100000
//*      into empty tables.                                             03150000
//*                                                                     03250000
//*    5.Change the data set names in the CBRSMERx steps to        @L1C 03300000
//*      the appropriate data set names for loading the DB2             03350000
//*      tables from other systems (this sample job is set up           03360000
//*      as though datasets are pre-allocated):                         03370000
//*                                                                     03380000
//*      smerge.map      = map dataset for DB2 in the job (reused       03390000
//*                        in this job, or can use separate data        03400000
//*                        sets for each job step if desired            03410000
//*      smerge.err      = error dataset for DB2 in the job (reused     03420000
//*                        in this job, or can use separate data        03430000
//*                        sets for each job step if desired            03440000
//*                                                                     03450000
//*      input.libtable  = DSN with the OLIBRARY table row values       03500000
//*                        from the system to be merged                 03550000
//*      workdsn.forlib  = work dataset for DB2 in job step             03600000
//*      sortdsn.forlib  = sort dataset for DB2 in job step             03650000
//*      discard.forlib  = DSN for the output of rows that could        03700000
//*                        not be merged from the other system          03750000
//*                                                                     03800000
//*      input.slttable  = DSN with the SLOT table row values           03850000
//*                        from the system to be merged                 03900000
//*      workdsn.forslot = work dataset for DB2 in job step             03950000
//*      sortdsn.forslot = sort dataset for DB2 in job step             04000000
//*      discard.forslot = DSN for the output of rows that could        04050000
//*                        not be merged from the other system          04100000
//*                                                                     04150000
//*      input.drvtable  = DSN with the DRIVE table row values          04200000
//*                        from the system to be merged                 04250000
//*      workdsn.fordrv  = work dataset for DB2 in job step             04300000
//*      sortdsn.fordrv  = sort dataset for DB2 in job step             04350000
//*      discard.fordrv  = DSN for the output of rows that could        04400000
//*                        not be merged from the other system          04450000
//*                                                                     04500000
//*      input.deltable  = DSN with the DELOBJT table row values        04550000
//*                        from the system to be merged                 04600000
//*      workdsn.fordelo = work dataset for DB2 in job step             04650000
//*      sortdsn.fordelo = sort dataset for DB2 in job step             04700000
//*      discard.fordelo = DSN for the output of rows that could        04750000
//*                        not be merged from the other system          04800000
//*                                                                     04850000
//*      input.voltable  = DSN with the VOLUME table row values         04900000
//*                        from the system to be merged                 04950000
//*      workdsn.forvol  = work dataset for DB2 in job step             05000000
//*      sortdsn.forvol  = sort dataset for DB2 in job step             05050000
//*      discard.forvol  = DSN for the output of rows that could        05100000
//*                        not be merged from the other system          05150000
//*                                                                     05200000
//*      input.tvoltable = DSN with the TAPEVOL table row values        05250000
//*                        from the system to be merged                 05300000
//*      workdsn.fortvol = work dataset for DB2 in job step             05350000
//*      sortdsn.fortvol = sort dataset for DB2 in job step             05400000
//*      discard.fortvol = DSN for the output of rows that could        05450000
//*                        not be merged from the other system          05500000
//*                                                                     05550000
//*      **NOTE: For these datasets, use size calculations that         05600000
//*              would be needed for your installation, using the       05650000
//*              DB2 guidelines in the DB2 Command and Utility          05700000
//*              Reference for the LOAD utility.                        05750000
//*                                                                     05800000
//*    6.Change the POSITION(xx:yy) in the CBRSMER* job steps to   @L1C 05850000
//*      correlate to the actual beginning, (and ending if needed),     05900000
//*      columns where the data for each column resides in the input    05950000
//*      datasets (the SYSREC DD statement dataset).                    06000000
//*                                                                     06008300
//*    7.The integer fields are set up as EXTERNAL(zz) in the job  @L1C 06016600
//*      steps, be sure that any integer values in the columns that     06024900
//*      are not the full length are padded with preceding zeroes       06033200
```

```
//*      in the input dataset (the SYSREC DD statement dataset).        06041500
//*                                                                     06050000
//*******************************************************************   06100000
//*                                                                     06150000
//*  After running this job, do the following:                          06200000
//*                                                                     06250000
//*    1.Check the return codes from the job to verify success          06300000
//*      or failure of the data merge.                                  06350000
//*                                                                     06400000
//*    2.Check the data sets below for any rows that could not          06450000
//*      be merged into the configuration database.  The most           06500000
//*      likely cause of failure would be duplicate rows.               06550000
//*                                                                     06600000
//*      discard.forlib  = DSN for the output of rows that could        06650000
//*                        not be merged from the other system          06700000
//*                                                                     06750000
//*      discard.forslot = DSN for the output of rows that could        06800000
//*                        not be merged from the other system          06850000
//*                                                                     06900000
//*      discard.fordrv  = DSN for the output of rows that could        06950000
//*                        not be merged from the other system          07000000
//*                                                                     07050000
//*      discard.fordelo = DSN for the output of rows that could        07100000
//*                        not be merged from the other system          07150000
//*                                                                     07200000
//*      discard.forvol  = DSN for the output of rows that could        07250000
//*                        not be merged from the other system          07300000
//*                                                                     07350000
//*      discard.fortvol = DSN for the output of rows that could        07400000
//*                        not be merged from the other system          07450000
//*                                                                     07500000
//*******************************************************************   07550000
//*                                                                     07600000
//*  CHANGE ACTIVITY:                                                   07650000
//*    $L0=HDZ11E0  150 970812 TUCLJT: Initial Release                  07700000
//*    $P1=K1B0989  R1B 090306 TUCBLC: Added new columns (EPI,     @P1A 07707100
//*                                    BKTYPE,OUNITNAME, DATACLAS, @P1A 07714200
//*                                    DSNFMT,SUBLEVEL, CAPACITYO, @P1A 07721300
//*                                    FREESPACEO, NUMLKBWO,       @P1A 07728400
//*                                    NUMPKBWO, NUMLKBDEO)        @P1A 07735500
//*    $L1=OAMR23M  R23 160405 TUCAED: Multi OAM AS Support        @L1A 07742600
//*                                                                     07750000
//********************************************************************* 07800000
//********************************************************************* 07850000
//*  Load configuration tables from different DB2 database              07900000
//********************************************************************* 07950000
//CBRSMER1 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='db2',        07975000
//         REGION=4096K                                                 08000000
//STEPLIB  DD DSN=SYS1.db2xx.VxRxM0.SDSNLOAD,DISP=SHR                   08025000
//SYSTSPRT DD SYSOUT=*                                                  08050000
//SYSREC   DD DSN=input.libtable,DISP=(OLD,KEEP)                        08300000
//SYSUT1   DD DSN=workdsn.forlib,DISP=(MOD,KEEP),UNIT=3390,             08318100
//         VOL=SER=dbpack                                               08336200
//SORTOUT  DD DSN=sortdsn.forlib,DISP=(MOD,KEEP),UNIT=3390,             08354300
//         VOL=SER=dbpack                                               08372400
//SYSDISC  DD DSN=discard.forlib,DISP=(MOD,KEEP),UNIT=3390,             08390500
//         VOL=SER=dbpack                                               08408600
//SYSMAP   DD DSN=smerge.map,DISP=(MOD,KEEP),UNIT=3390,                 08426700
//         VOL=SER=dbpack                                               08444800
//SYSERR   DD DSN=smerge.err,DISP=(MOD,KEEP),UNIT=3390,                 08462900
//         VOL=SER=dbpack                                               08481000
//SYSPRINT DD SYSOUT=*                                                  08500000
//UTPRINT  DD SYSOUT=*                                                  08525000
//SYSUDUMP DD SYSOUT=*                                                  08550000
//SYSIN    DD *                                                         08600000
                                                                        08650000
 LOAD DATA INDDN(SYSREC)                                                08700000
      RESUME YES                                                        08750000
      INTO TABLE OLIBRARY                                               08800000
    (NAME       POSITION(xx)  CHAR(8),                                  08850000
     ONLINE     POSITION(xx)  CHAR(1),                                  08900000
     OPERATNL   POSITION(xx)  CHAR(1),                                  08950000
     PATHSTAT   POSITION(xx)  CHAR(1),                                  09000000
     COMMAND    POSITION(xx)  CHAR(5) DEFAULTIF(COMMAND=''),            09050000
     PRIMCTC    POSITION(xx)  CHAR(4) DEFAULTIF(PRIMCTC=''),            09100000
     PRIMPORT   POSITION(xx)  CHAR(1) DEFAULTIF(PRIMPORT=''),           09150000
     ALTCTC     POSITION(xx)  CHAR(4) DEFAULTIF(ALTCTC=''),             09200000
     ALTPORT    POSITION(xx)  CHAR(1) DEFAULTIF(ALTPORT=''),            09250000
     FAULT      POSITION(xx)  CHAR(3) DEFAULTIF(FAULT=''),              09300000
     OLIBTYPE   POSITION(xx)  CHAR(1),                                  09350000
     NUMSLOTS   POSITION(xx)  INTEGER EXTERNAL(3),                      09400000
     NUMESLOT   POSITION(xx)  INTEGER EXTERNAL(3),                      09450000
```

```
     NUMDRVS    POSITION(xx)  INTEGER EXTERNAL(3),                      09500000
     RCOMMAND   POSITION(xx)  CHAR(5) DEFAULTIF(RCOMMAND=''),           09550000
     DEVTYPE    POSITION(xx)  CHAR(8) DEFAULTIF(RCOMMAND=''),           09600000
     LIBRDES    POSITION(xx:yy)  CHAR DEFAULTIF(LIBRDES=''),            09650000
     CLIBRARY   POSITION(xx)  CHAR(8) DEFAULTIF(CLIBRARY=''),           09700000
     MEDIATYP   POSITION(xx)  CHAR(8) DEFAULTIF(MEDIATYP=''),           09750000
     LIBINDEX   POSITION(xx)  INTEGER EXTERNAL(1),                      09800000
     PLIBRARY   POSITION(xx)  CHAR(16) DEFAULTIF(MEMBER=''),            09850000
     MEMBER     POSITION(xx)  CHAR(8) DEFAULTIF(PLIBRARY=''))           09900000
/*                                                                      10100000
//********************************************************************* 10150000
//CBRSMER2 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='db2',        10175000
//         REGION=4096K                                                 10200000
//STEPLIB  DD DSN=SYS1.db2xx.VxRxM0.SDSNLOAD,DISP=SHR                   10225000
//SYSTSPRT DD SYSOUT=*                                                  10250000
//SYSREC   DD DSN=input.slttable,DISP=(OLD,KEEP)                        10500000
//SYSUT1   DD DSN=workdsn.forslot,DISP=(MOD,KEEP),UNIT=3390,            10518100
//         VOL=SER=dbpack                                               10536200
//SORTOUT  DD DSN=sortdsn.forslot,DISP=(MOD,KEEP),UNIT=3390,            10554300
//         VOL=SER=dbpack                                               10572400
//SYSDISC  DD DSN=discard.forslot,DISP=(MOD,KEEP),UNIT=3390,            10590500
//         VOL=SER=dbpack                                               10608600
//SYSMAP   DD DSN=smerge.map,DISP=(MOD,KEEP),UNIT=3390,                 10626700
//         VOL=SER=dbpack                                               10644800
//SYSERR   DD DSN=smerge.err,DISP=(MOD,KEEP),UNIT=3390,                 10662900
//         VOL=SER=dbpack                                               10681000
//SYSPRINT DD SYSOUT=*                                                  10700000
//UTPRINT  DD SYSOUT=*                                                  10725000
//SYSUDUMP DD SYSOUT=*                                                  10750000
//SYSIN    DD *                                                         10800000
                                                                        10850000
 LOAD DATA INDDN(SYSREC)                                                10900000
      RESUME YES                                                        10950000
      INTO TABLE SLOT                                                   11000000
    (NAME       POSITION(xx)  CHAR(3),                                  11050000
     OLIBRARY   POSITION(xx)  CHAR(8),                                  11100000
     OCCUPIED   POSITION(xx)  CHAR(1),                                  11150000
     OPERATNL   POSITION(xx)  CHAR(1),                                  11200000
     VOLSER0    POSITION(xx)  CHAR(6) DEFAULTIF(VOLSER0=''),            11250000
     VOLSER1    POSITION(xx)  CHAR(6) DEFAULTIF(VOLSER1=''))            11300000
 /*                                                                     11500000
//********************************************************************* 11550000
//CBRSMER3 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='db2',        11575000
//         REGION=4096K                                                 11600000
//STEPLIB  DD DSN=SYS1.db2xx.VxRxM0.SDSNLOAD,DISP=SHR                   11625000
//SYSTSPRT DD SYSOUT=*                                                  11650000
//SYSREC   DD DSN=input.drvtable,DISP=(OLD,KEEP)                        11900000
//SYSUT1   DD DSN=workdsn.fordrv,DISP=(MOD,KEEP),UNIT=3390,             11918100
//         VOL=SER=dbpack                                               11936200
//SORTOUT  DD DSN=sortdsn.fordrv,DISP=(MOD,KEEP),UNIT=3390,             11954300
//         VOL=SER=dbpack                                               11972400
//SYSDISC  DD DSN=discard.fordrv,DISP=(MOD,KEEP),UNIT=3390,             11990500
//         VOL=SER=dbpack                                               12008600
//SYSMAP   DD DSN=smerge.map,DISP=(MOD,KEEP),UNIT=3390,                 12026700
//         VOL=SER=dbpack                                               12044800
//SYSERR   DD DSN=smerge.err,DISP=(MOD,KEEP),UNIT=3390,                 12062900
//         VOL=SER=dbpack                                               12081000
//SYSPRINT DD SYSOUT=*                                                  12100000
//UTPRINT  DD SYSOUT=*                                                  12125000
//SYSUDUMP DD SYSOUT=*                                                  12150000
//SYSIN    DD *                                                         12200000
                                                                        12250000
 LOAD DATA INDDN(SYSREC)                                                12300000
      RESUME YES                                                        12350000
      INTO TABLE DRIVE                                                  12400000
    (NAME       POSITION(xx)    CHAR(8),                                12450000
     OLIBRARY   POSITION(xx)    CHAR(8),                                12500000
     CTC        POSITION(xx)    CHAR(4),                                12550000
     SCSI       POSITION(xx)    CHAR(1),                                12600000
     LUN        POSITION(xx)    CHAR(1),                                12650000
     ONLINE     POSITION(xx)    CHAR(1),                                12700000
     OPERATNL   POSITION(xx)    CHAR(1),                                12750000
     LDRIVENO   POSITION(xx)    CHAR(1) DEFAULTIF(LDRIVENO=''),         12800000
     DRIVTYPE   POSITION(xx)    CHAR(1),                                12850000
     DEVTYPE    POSITION(xx)    CHAR(8) DEFAULTIF(DEVTYPE=''),          12900000
     DRIVEDES   POSITION(xx:yy) CHAR DEFAULTIF(DRIVDES=''),             12950000
     DRIVENUM   POSITION(xx)    INTEGER EXTERNAL(1),                    13000000
     MEMBER     POSITION(xx)    CHAR(16) DEFAULTIF(MEMBER=''))          13050000
 /*                                                                     13200000
//********************************************************************* 13250000
//CBRSMER4 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='db2',        13275000
//         REGION=4096K                                                 13300000
```

```
//STEPLIB  DD DSN=SYS1.db2xx.VxRxM0.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSREC   DD DSN=input.voltable,DISP=(OLD,KEEP)
//SYSUT1   DD DSN=workdsn.forvol,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SORTOUT  DD DSN=sortdsn.forvol,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SYSDISC  DD DSN=discard.forvol,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SYSMAP   DD DSN=smerge.map,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SYSERR   DD DSN=smerge.err,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 LOAD DATA INDDN(SYSREC)
      RESUME YES
      INTO TABLE VOLUME
    (VOLSER     POSITION(xx)  CHAR(6),
     OVOLSER    POSITION(xx)  CHAR(6),
     LOCATION   POSITION(xx)  CHAR(1),
     SLOT       POSITION(xx)  CHAR(3),
     OLIBRARY   POSITION(xx)  CHAR(8),
     SHELFLOC   POSITION(xx)  CHAR(32),
     MNTDATE    POSITION(xx)  DATE EXTERNAL(10),
     WRTDATE    POSITION(xx)  DATE EXTERNAL(10),
     EXPDATE    POSITION(xx)  DATE EXTERNAL(10),
     EJECTDAT   POSITION(xx)  DATE EXTERNAL(10),
     LASTDATA   POSITION(xx)  INTEGER EXTERNAL(1),
     LASTVTCL   POSITION(xx)  INTEGER EXTERNAL(1),
     LASTVTCP   POSITION(xx)  INTEGER EXTERNAL(1),
     VOLUMESET  POSITION(xx)  CHAR(8),
     TYPE       POSITION(xx)  CHAR(1),
     ORIENT     POSITION(xx)  CHAR(1),
     FULL       POSITION(xx)  CHAR(1),
     READABLE   POSITION(xx)  CHAR(1),
     WRITABLE   POSITION(xx)  CHAR(1),
     WRTPROT    POSITION(xx)  CHAR(1),
     OWNERP     POSITION(xx)  CHAR(1),
     OWNER      POSITION(xx)  CHAR(32),
     FRESPACE   POSITION(xx)  INTEGER EXTERNAL(7),
     DELSPACE   POSITION(xx)  INTEGER EXTERNAL(1),
     DELCOUNT   POSITION(xx)  INTEGER EXTERNAL(1),
     FRAGIDX    POSITION(xx)  INTEGER EXTERNAL(1),
     MEDIATYP   POSITION(xx)  CHAR(2),
     CREDATE    POSITION(xx)  DATE EXTERNAL(10),
     ERRSTAT    POSITION(xx)  INTEGER EXTERNAL(1),
     VOLEMPTY   POSITION(xx)  CHAR(1),
     RECOUNT    POSITION(xx)  INTEGER EXTERNAL(1),
     CAPACITY   POSITION(xx)  INTEGER EXTERNAL(7),
     PLIBRARY   POSITION(xx)  CHAR(8),
     MEMBER     POSITION(xx)  CHAR(16))
 /*
//**********************************************************************
//CBRSMER5 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='db2',
//         REGION=4096K
//STEPLIB  DD DSN=SYS1.db2xx.VxRxM0.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSREC   DD DSN=input.deltable,DISP=(OLD,KEEP)
//SYSUT1   DD DSN=workdsn.fordelo,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SORTOUT  DD DSN=sortdsn.fordelo,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SYSDISC  DD DSN=discard.fordelo,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SYSMAP   DD DSN=smerge.map,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SYSERR   DD DSN=smerge.err,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 LOAD DATA INDDN(SYSREC)
      RESUME YES
      INTO TABLE DELOBJT
    (COLNAME    POSITION(xx)  CHAR(44),
     OBJNAME    POSITION(xx)  CHAR(44),
```

```
     VOLSER     POSITION(xx)  CHAR(6),                                  17000000
     VTOCTOKN   POSITION(xx)  INTEGER EXTERNAL(1),                      17050000
     OBJSIZE    POSITION(xx)  INTEGER EXTERNAL(6))                      17100000
 /*                                                                     17250000
//******************************************************************** 17300000
//CBRSMER6 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='db2',        17325000
//         REGION=4096K                                                 17350000
//STEPLIB  DD DSN=SYS1.db2xx.VxRxM0.SDSNLOAD,DISP=SHR                   17375000
//SYSTSPRT DD SYSOUT=*                                                  17400000
//SYSREC   DD DSN=input.tvoltble,DISP=(OLD,KEEP)                        17437500
//SYSUT1   DD DSN=workdsn.fortvol,DISP=(MOD,KEEP),UNIT=3390,            17475000
//         VOL=SER=dbpack                                               17512500
//SORTOUT  DD DSN=sortdsn.fortvol,DISP=(MOD,KEEP),UNIT=3390,            17550000
//         VOL=SER=dbpack                                               17587500
//SYSDISC  DD DSN=discard.fortvol,DISP=(MOD,KEEP),UNIT=3390,            17625000
//         VOL=SER=dbpack                                               17662500
//SYSMAP   DD DSN=smerge.map,DISP=(MOD,KEEP),UNIT=3390,                 17700000
//         VOL=SER=dbpack                                               17737500
//SYSERR   DD DSN=smerge.err,DISP=(MOD,KEEP),UNIT=3390,                 17775000
//         VOL=SER=dbpack                                               17812500
//SYSPRINT DD SYSOUT=*                                                  17850000
//UTPRINT  DD SYSOUT=*                                                  17875000
//SYSUDUMP DD SYSOUT=*                                                  17900000
//SYSIN    DD *                                                         17950000
                                                                        18000000
 LOAD DATA INDDN(SYSREC)                                                18050000
      RESUME YES                                                        18100000
      INTO TABLE TAPEVOL                                                18150000
    (VOLSER     POSITION(xx)     CHAR(6),                               18200000
     UNITNAME   POSITION(xx)     CHAR(8),                               18250000
     MEDIATYP   POSITION(xx)     CHAR(2),                               18300000
     STORGRP    POSITION(xx)     CHAR(8),                               18350000
     TYPE       POSITION(xx)     CHAR(1),                               18400000
     CREDATE    POSITION(xx)     DATE EXTERNAL(10),                     18450000
     MNTDATE    POSITION(xx)     DATE EXTERNAL(10),                     18500000
     WRTDATE    POSITION(xx)     DATE EXTERNAL(10),                     18550000
     EXPDATE    POSITION(xx)     DATE EXTERNAL(10),                     18600000
     CAPACITY   POSITION(xx)     INTEGER EXTERNAL(8),                   18650000
     FRESPACE   POSITION(xx)     INTEGER EXTERNAL(8),                   18700000
     LSTBLKID   POSITION(xx)     INTEGER EXTERNAL(8),                   18750000
     PFULL      POSITION(xx)     INTEGER EXTERNAL(2),                   18800000
     NUMLBLKS   POSITION(xx)     INTEGER EXTERNAL(8),                   18850000
     NUMLKBW    POSITION(xx)     INTEGER EXTERNAL(8),                   18900000
     NUMPKBW    POSITION(xx)     INTEGER EXTERNAL(8),                   18950000
     NUMLKBDE   POSITION(xx)     INTEGER EXTERNAL(8),                   19000000
     FULL       POSITION(xx)     CHAR(1),                               19050000
     READABLE   POSITION(xx)     CHAR(1),                               19100000
     WRITABLE   POSITION(xx)     CHAR(1),                               19150000
     INUSE      POSITION(xx)     CHAR(1),                               19200000
     COPIED     POSITION(xx)     CHAR(1),                               19250000
     AVOLSER    POSITION(xx)     CHAR(6),                               19300000
     COMPACT    POSITION(xx)     CHAR(1),                               19350000
     MEMBER     POSITION(xx)     CHAR(16),                              19365300
     EPI        POSITION(xx)     INTEGER EXTERNAL(2),                   19380600
     BKTYPE     POSITION(xx)     CHAR(1),                               19395900
     OUNITNAM   POSITION(xx)     CHAR(8),                               19411200
     DATACLAS   POSITION(xx)     CHAR(8),                               19426500
     DSNFMT     POSITION(xx)     CHAR(1),                               19441800
     SUBLEVEL   POSITION(xx)     CHAR(1),                               19457100
     CAPACITYO  POSITION(xx)     INTEGER EXTERNAL(8),                   19472400
     FRESPACEO  POSITION(xx)     INTEGER EXTERNAL(8),                   19487700
     NUMLKBWO   POSITION(xx)     INTEGER EXTERNAL(8),                   19503000
     NUMPKBWO   POSITION(xx)     INTEGER EXTERNAL(8),                   19518300
     NUMLKBDEO  POSITION(xx)     INTEGER EXTERNAL(8))                   19533600
 /*                                                                     19550000
```

CBRSG100

SAMPLIB member CBRSG100, as shown here, helps you merge OAM administration databases and catalog entries and OAM object storage databases for use with DB2 data sharing in an OAMplex. You must perform the prerequisites and modify and run the job successfully before you use OAM.

```
//CBRSG100 JOB MSGLEVEL=(1,1),MSGCLASS=A                                00050000
//******************************************************************** 00100000
//*                                                                     00150000
//*  $SEG(CBRSG100) COMP(DBM) PROD(OAM):                                00200000
//*                                                                     00250000
//*  OAM DB2 Database Migration Job (for OAM Administrative,            00250001
//*  object directores, and storage databases).                         00250002
```

```
//*                                                                     00250003
//*  Licensed Materials - Property of IBM                               00250004
//*  5650-ZOS                                                           00250005
//*  COPYRIGHT IBM CORP. 1999, 2017                                     00250006
//*                                                                     00250007
//*  This job is intended to assist in the task of merging OAM          00250008
//*  administration databases and OAM object directories and            00250009
//*  storage databases for use with DB2 datasharing in an OAMPLEX       00250010
//*  environment.                                                       00250011
//*                                                                     01350000
//*******************************************************************   01400000
//* WARNING: You MUST verify the following before proceeding:      @L1C 01400001
//*                                                                     01400002
//*    1. There are no two storage groups across any systems            01400003
//*       which are being combined, which have the same collection      01400004
//*       name associated with them. A collection CANNOT span object    01400005
//*       storage groups, therefore may belong to one and only one      01400006
//*       object storage group.                                         01400007
//*       If this condition exists:                                     01400008
//*          - the collection name on one of the systems being          01400009
//*            combined must change, and the ACS routines updated       01400010
//*            accordingly.                                             01400011
//*            OR                                                       01400012
//*          - the two storage groups must be combined into a           01400013
//*            single storage group.                                    01400014
//*                                                                     01400015
//*    2. There are no two collection names across any systems          01400016
//*       that are being combined, which have the same collection       01400017
//*       ID.  Objects are associated with a collection by its ID,      01400018
//*       and the collection ID is unique in the collection table.      01400019
//*       Any collections across systems being combined, which had      01400020
//*       the same collection ID, must have been changed, and the       01400021
//*       object directory entries using these collections IDs,         01400022
//*       must have been updated.                                       01400023
//*                                                                     01400024
//*    3. There are no two management classes across any systems       01400025
//*       that are being combined, which have the same management       01400026
//*       class ID.  Objects are associated with a management class     01400027
//*       in the object directory by its ID, and the management class   01400028
//*       ID is unique in the management class table.                   01400029
//*       Any management classes across systems being combined which    01400030
//*       had the same ID, must have been changed, and the object       01400031
//*       directory entries using the modified management class' ID     01400032
//*       must have been updated.                                       01400033
//*                                                                     01400034
//*    4. There are no two storage classes across any systems           01400035
//*       that are being combined, which have the same storage          01400036
//*       class ID.  Objects are associated with a storage class        01400037
//*       in the object directory by its ID, and the storage class      01400038
//*       ID is unique in the storage class table.                      01400039
//*       Any storage classes across systems being combined which       01400040
//*       had the same ID, must have been changed, and the object       01400041
//*       directory entries using the modified storage class' ID        01400042
//*       must have been updated.                                       01400043
//*                                                                     01400044
//*                                                                     01400045
//*   NOTE1:  In order to 'correct' a duplicate collection ID           01400046
//*           situation you can do the following:                       01400047
//*           - it is best to make changes to the data that is          01400048
//*             being moved rather than the data on the system          01400049
//*             where the data is being combined.                       01400050
//*           1. On the system where the data is to be combined,        01400051
//*              determine what the next available collection ID        01400052
//*              is that can be used.                                   01400053
//*           2. On the system where the data is coming from, with      01400054
//*              the duplicate collection ID, change the collection     01400055
//*              ID associated with the collection name in the          01400056
//*              collection table to the ID determined in step 1.       01400057
//*           3. Change all of the object directory entries in the      01400058
//*              object directory for the storage group to which the    01400059
//*              collection belongs, where the collection ID is the     01400060
//*              ID previously associated with the collection to        01400061
//*              the new collection ID used from step 2.                01400062
//*           4. Use IDCAMS to catalog the new collection in the        01400063
//*              catalog on the target system.                          01400064
//*           example:                                                  01400065
//*           Object directories on system 1 and system 2 are to be     01400066
//*           combined.  COLL.SYS1.DATA1997 on system 1 has the         01400067
//*           same collection ID as COLL.SYS2.DATA1997 on system 2.     01400068
//*                                                                     01400069
//*           BEFORE:                                                   01400070
//*           -------                                                   01400071
```

```
//*          System 1                       System 2
//*          Coll-name          Coll-ID     Coll-name          Coll-ID
//*
//*          OBJCOLL.SYS1.TEST   001        OBJCOLL.SYS2.TEST   001
//*          COLL.SYS1.DATA1997  002        COLL.SYS2.DATA1997  002
//*          COLL.SYS1.DATA1998  004        COLL.SYS2.DATA1998  003
//*          COLL.SYS1.REPORTS   005
//*
//*          System 1 will be the target system where data is combined
//*          OBJCOLL.SYS2.TEST will not be moved, as this is test data
//*          that is not needed on the combined system
//*
//*          Step 1:  The next available collection ID on
//*                   system 1 is 006
//*
//*          Step 2:  On System 2 before attempting the combine:
//*                   SQL UDPATE OAMADMIN.CBR_COLLECTION_TBL
//*                       SET ODCLID = 6 WHERE
//*                       ODCLNAME = 'COLL.SYS2.DATA1997';
//*                   COMMIT;
//*
//*          Step 3:  On System 2 before attempting the combine:
//*                   SQL UDPATE GROUPxx.OSM_OBJ_DIR
//*                       SET ODCLID = 6 WHERE
//*                           ODCLID = 2;
//*                   COMMIT;
//*
//*          Step 4:  On System 1 before attempting the combine:
//*                   IDCAMS DEFINE NONVSAM RECATALOG -
//*                       COLLECTION NAME(COLL.SYS2.DATA1997)
//*
//*          AFTER:
//*          ------
//*          System 1                       System 2
//*          Coll-name          Coll-ID     Coll-name          Coll-ID
//*
//*          OBJCOLL.SYS1.TEST   001        OBJCOLL.SYS2.TEST   001
//*          COLL.SYS1.DATA1997  002        COLL.SYS2.DATA1997  006
//*          COLL.SYS1.DATA1998  004        COLL.SYS2.DATA1998  003
//*          COLL.SYS1.REPORTS   005
//*
//*
//* WARNING:  In order to 'correct' a duplicate collection name
//*          situation, you will have to investigate your SMS
//*          CDS more closely and see where the storage group is
//*          assigned based on the collection name and see if the
//*          data can be combined under one storage group, or if
//*          objects in one collection need to be changed to another
//*          collection altogether.  Then the ACS routines would have
//*          to also be updated to handle the new collection.
//*          This needs to be done with the assistance of your
//*          application interface and your systems programmer to
//*          determine the best plan for your installation.
//*
//*   NOTE2:  In order to 'correct' a duplicate management class
//*          number situation you can do similar steps as the ones
//*          to correct duplicate collection IDs, with the exception
//*          that the catalog step is not needed.
//*          In brief the steps would be:
//*          1. determine the next available management class number
//*             'x' below is the new management class number
//*             'y' below is the old management class number
//*          2. Change the management class number on the
//*             'from' system
//*             SQL UDPATE OAMADMIN.CBR_MGT_CLASS_TBL
//*                 SET MCNUM = x WHERE
//*                 ODMCNAME = 'duplicate.mc.name';
//*             COMMIT;
//*          3. Change all of the object directory entries with the
//*             old management class number to the new management
//*             class number on the 'from' system
//*                   SQL UDPATE GROUPxx.OSM_OBJ_DIR
//*                       SET ODMCNUM = x WHERE
//*                           ODMCNUM = y;
//*                   COMMIT;
//*
//*   NOTE3:  In order to 'correct' a duplicate storage class number
//*          situation you can do similar steps as the ones to
//*          correct duplicate collection IDs, with the exception
//*          that the catalog step is not needed.
//*          In brief the steps would be:
//*          1. determine the next available storage class number
```

```
//*              'x' below is the new storage class number
//*              'y' below is the old storage class number
//*           2. Change the storage class ID on the 'from' system
//*              SQL UDPATE OAMADMIN.CBR_STO_CLASS_TBL
//*                  SET SCNUM = x WHERE
//*                  ODMCNAME = 'duplicate.sc.name';
//*              COMMIT;
//*           3. Change all of the object directory entries with the
//*              old storage class number to the new storage class
//*              number on the 'from' system
//*                    SQL UDPATE GROUPxx.OSM_OBJ_DIR
//*                        SET ODSCNUM = x WHERE
//*                            ODSCNUM = y;
//*                    COMMIT;
//*
//*******************************************************************
//*
//*  It is recommended that you create a DB2 image copy of the
//*  existing tables:
//*    OAM administration database tables:
//*       OAMADMIN.CBR_MGT_CLASS_TBL
//*       OAMADMIN.CBR_STO_CLASS_TBL
//*       OAMADMIN.CBR_COLLECTION_TBL
//*    OAM object directory and object storage database tables:
//*       all GROUPxx.OSM_OBJ_DIR tables
//*       all GROUPxx.OSM_04K_OBJ_TBL tables
//*       all GROUPxx.OSM_32K_OBJ_TBL tables
//*     where GROUPxx = GROUP00 - GROUP99
//*
//*******************************************************************
//*
//*  Before running this job, you must change the following:
//*
//*    1. Change the PARM='db2' in the JOB STEP statements to      @L1C
//*      the name of the DB2 Subsystem in your installation.
//*
//*     In a multiple OAM configuration, you will create a         @L1A
//*     unique job for each Object OAM instance and specify        @L1A
//*     the DB2 SSID for the DB2 subsystem associated with         @L1A
//*     the Object OAM instance.                                   @L1A
//*                                                                @L1A
//*    2. Change the data set name SYS1.db2xx.VxRxM0.SDSNLOAD      @L1C
//*      in the STEPLIB statements to the data set name used
//*      for the DB2 SDNSLOAD dataset if necessary.
//*
//*     In a multiple OAM configuration, you will create a         @L1A
//*     unique job for each Object OAM instance and specify        @L1A
//*     the SYS1.db2xx.VXRXM0.SDSNLOAD for the DB2 subsystem       @L1A
//*     associated with the Object OAM instance.                   @L1A
//*                                                                @L1A
//*    3. Change "dbpack" to the volume serials that your          @L1A
//*     target database should reside on.                          @L1A
//*                                                                @L1A
//*    4. Change RESUME YES to RESUME NO if you are loading        @L1C
//*      into empty tables.
//*
//*    5. Change the data set names in the job steps to the        @L1C
//*       appropriate data set names for loading the DB2
//*       tables from other systems (this sample job is set up
//*       as though datasets are pre-allocated):
//*
//*       sg100.map       = map dataset for DB2 in the job (reused
//*                         in this job, or can use separate data
//*                         sets for each job step if desired
//*       sg100.err       = error dataset for DB2 in the job (reused
//*                         in this job, or can use separate data
//*                         sets for each job step if desired
//*
//*       input.clntable  = DSN with the collection table row values
//*                         from the system to be merged
//*       workdsn.forcln  = work dataset for DB2 in job step
//*       sortdsn.forcln  = sort dataset for DB2 in job step
//*       discard.forcln  = DSN for the output of rows that could
//*                         not be merged from the other system
//*
//*       input.mctable   = DSN with the management class table row
//*                         values from the system to be merged
//*       workdsn.formc   = work dataset for DB2 in job step
//*       sortdsn.formc   = sort dataset for DB2 in job step
//*       discard.formc   = DSN for the output of rows that could
//*                         not be merged from the other system
//*
```

```
//*      input.sctable   = DSN with the storage class table row         01450093
//*                        values from the system to be merged          01450094
//*      workdsn.forsc   = work dataset for DB2 in job step             01450095
//*      sortdsn.forsc   = sort dataset for DB2 in job step             01450096
//*      discard.forsc   = DSN for the output of rows that could        01450097
//*                        not be merged from the other system          01450098
//*                                                                     01450099
//*      input.objdir    = DSN with the object directory table row      01450100
//*                        values from the system to be merged          01450101
//*      workdsn.forodir = work dataset for DB2 in job step             01450102
//*      sortdsn.forodir = sort dataset for DB2 in job step             01450103
//*      discard.forodir = DSN for the output of rows that could        01450104
//*                        not be merged from the other system          01450105
//*                                                                     01450106
//*      input.obj04k    = DSN with the object 4K table row             01450107
//*                        values from the system to be merged          01450108
//*      workdsn.for04k  = work dataset for DB2 in job step             01450109
//*      sortdsn.for04k  = sort dataset for DB2 in job step             01450110
//*      discard.for04k  = DSN for the output of rows that could        01450111
//*                        not be merged from the other system          01450112
//*                                                                     01450113
//*      input.obj32k    = DSN with the object 32K table row            01450114
//*                        values from the system to be merged          01450115
//*      workdsn.for32k  = work dataset for DB2 in job step             01450116
//*      sortdsn.for32k  = sort dataset for DB2 in job step             01450117
//*      discard.for32k  = DSN for the output of rows that could        01450118
//*                        not be merged from the other system          01450119
//*                                                                     01450120
//*     **NOTE: For these datasets, use size calculations that          01450121
//*             would be needed for your installation, using the        01450122
//*             DB2 guidelines in the DB2 Command and Utility           01450123
//*             Reference for the LOAD utility.                         01450124
//*                                                                     01450125
//*   6. Change storage_group_hlq in the job steps to the         @L1C 01450126
//*      high level qualifier for the object storage group              01450127
//*      tables that are being merged.                                  01450128
//*                                                                     01450129
//*   7. Load the information from the OAM administration         @L1C 01450130
//*      database tables and the object storage group databases   @L1C 01450131
//*      tables from the other OAM/system where data is to be     @L1C 01450132
//*      copied from into datasets on this system.                @L1C 01450133
//*      The simplest way to do this is to do an SQL SELECT * from     01450134
//*      the OAM tables to be merged on the other system and editing   01450135
//*      the column headers out of the resulting output so that just   01450136
//*      the data from the rows remains.                                01450137
//*                                                                     01450138
//*   8. Note the beginning, (and ending if needed), columns      @L1C 01450139
//*      where the data for each column resides in the table rows.     01450140
//*                                                                     01450141
//*   9. If necessary, change the POSITION(xx:yy) in the job      @L1C 01450142
//*      steps to correlate to the actual beginning (and ending)        01450143
//*      columns where the data for each column resides in the          01450144
//*      input datasets (the SYSREC DD statement dataset).              01450145
//*                                                                     01450146
//*      - For columns which are defined as specific length,            01450147
//*        like CHAR(2), only the start position is needed and          01450148
//*        the end will be determined by the length of the column       01450149
//*      - For columns which are defined as VARCHAR, only a             01450150
//*        start position is needed if the first 2 bytes of             01450151
//*        the data is the length of the following data.                01450152
//*        Otherwise, a beginning and ending designation are            01450153
//*        needed.                                                      01450154
//*                                                                     01450155
//*  10. The integer fields are set up as EXTERNAL(zz) in the job @L1C 01450156
//*      steps, be sure that any integer values in the columns that    01450157
//*      are not the full length are padded with preceding zeroes      01450158
//*      in the input dataset (the SYSREC DD statement dataset).       01450159
//*                                                                     01450160
//*  11. If your input dataset (DD SYSREC) does not have the data @L1C 01450161
//*      always in specific columns because of the varying length      01450162
//*      fields, the easiest way to change the POSITION(xx:yy)          01450163
//*      statements would be to make the POSITION start column          01450164
//*      an offset from the end of the previous field.  For             01450165
//*      example:                                                       01450166
//*                                                                     01450167
//*      LOAD DATA INDDN(SYSREC)                                        01450168
//*           REPLACE                                                   01450169
//*           INTO TABLE OAMADMIN.CBR_COLLECTION_TBL                    01450170
//*         (ODCLSCNM   POSITION(1 )  VARCHAR(30),                      01450171
//*          ODCLMCNM   POSITION(+2)  VARCHAR(30),                      01450172
//*          ODCLSGNM   POSITION(+2)  VARCHAR(30),                      01450173
//*          ODCLID     POSITION(+2)     INTEGER,                       01450174
```

```
//*           ODCLNAME   POSITION(+2)  VARCHAR(44));                     01450175
//*                                                                      01450176
//*       In this example, the data for each subsequent column is       01450177
//*       expected to be 2 positions from the end of the preceding      01450178
//*       data, with the first 2 bytes of the VARCHAR fields being      01450179
//*       the actual length of the field's data.                        01450180
//*                                                                      01450181
//*   12. Repeat steps CBRSG104-CBRSG106 for each set of object    @L1C 01450182
//*       storage group tables that are being merged.                   01450183
//*                                                                      01450184
//*                                                                      16200000
//*  After running this job, do the following:                          16250000
//*                                                                      16300000
//*    1. Check the return codes from the job to verify success         16350000
//*       or failure of the data merge.                                 16400000
//*                                                                      16450000
//*    2. Check the data sets below for any rows that could not         16500000
//*       be merged into the configuration database.  The most          16550000
//*       likely cause of failure would be duplicate rows.              16600000
//*                                                                      16650000
//*       discard.forcln  = DSN for the output of rows that could       16700000
//*                         not be merged from the other system         16750000
//*                                                                      16800000
//*       discard.formc   = DSN for the output of rows that could       16850000
//*                         not be merged from the other system         16900000
//*                                                                      16950000
//*       discard.forsc   = DSN for the output of rows that could       17000000
//*                         not be merged from the other system         17050000
//*                                                                      17100000
//*       discard.forodir = DSN for the output of rows that could       17150000
//*                         not be merged from the other system         17200000
//*                                                                      17250000
//*       discard.for04k  = DSN for the output of rows that could       17300000
//*                         not be merged from the other system         17350000
//*                                                                      17400000
//*       discard.for32k  = DSN for the output of rows that could       17450000
//*                         not be merged from the other system         17500000
//*********************************************************************   17550000
//*                                                                      17600000
//*                                                                      17650000
//*  CHANGE ACTIVITY:                                                    17700000
//*    $L0=HDZ11E0  150 970812 TUCLJT: Initial Release                   17750000
//*    $P1=K1B0989  R1B 090306 TUCBLC: Add new columns (ODLOBFL,   @P1A 17762500
//*                            ODBK2LOC, ODBK2SEC, ODSTATF,        @P1A 17775000
//*                            ODRETDT, ODINSTID)                  @P1A 17787500
//*    $L1=OAMR23M  R23 160405 TUCAED: Multi OAM AS Support        @L1A 17787501
//*                                                                      17800000
//********************************************************************** 17850000
//********************************************************************** 17900000
//*  Load configuration tables from different DB2 database               17950000
//********************************************************************** 18000000
//CBRSG101 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='db2',         18025000
//         REGION=4096K                                                  18050000
//STEPLIB  DD DSN=SYS1.db2xx.VxRxM0.SDSNLOAD,DISP=SHR                    18075000
//SYSTSPRT DD SYSOUT=*                                                   18100000
//SYSREC   DD DSN=input.clntable,DISP=(OLD,KEEP)                         18350000
//SYSUT1   DD DSN=workdsn.forcln,DISP=(MOD,KEEP),UNIT=3390,              18368100
//         VOL=SER=dbpack                                                18386200
//SORTOUT  DD DSN=sortdsn.forcln,DISP=(MOD,KEEP),UNIT=3390,              18404300
//         VOL=SER=dbpack                                                18422400
//SYSDISC  DD DSN=discard.forcln,DISP=(MOD,KEEP),UNIT=3390,              18440500
//         VOL=SER=dbpack                                                18458600
//SYSMAP   DD DSN=sg100.map,DISP=(MOD,KEEP),UNIT=3390,                   18476700
//         VOL=SER=dbpack                                                18494800
//SYSERR   DD DSN=sg100.err,DISP=(MOD,KEEP),UNIT=3390,                   18512900
//         VOL=SER=dbpack                                                18531000
//SYSPRINT DD SYSOUT=*                                                   18550000
//UTPRINT  DD SYSOUT=*                                                   18575000
//SYSUDUMP DD SYSOUT=*                                                   18600000
//SYSIN    DD *                                                          18650000
                                                                         18700000
 LOAD DATA INDDN(SYSREC)                                                 18750000
      REPLACE                                                            18800000
      INTO TABLE OAMADMIN.CBR_COLLECTION_TBL                             18850000
    (ODCLSCNM   POSITION(xx)     VARCHAR,                                18900000
     ODCLMCNM   POSITION(xx)     VARCHAR,                                18950000
     ODCLSGNM   POSITION(xx)     VARCHAR,                                19000000
     ODCLID     POSITION(xx)     INTEGER EXTERNAL(3),                    19050000
     ODCLNAME   POSITION(xx)     VARCHAR)                                19100000
/*                                                                       19300000
//********************************************************************** 19350000
//CBRSG102 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='db2',         19375000
```

```
//         REGION=4096K
//STEPLIB  DD DSN=SYS1.db2xx.VxRxM0.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSREC   DD DSN=input.mctable,DISP=(OLD,KEEP)
//SYSUT1   DD DSN=workdsn.formc,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SORTOUT  DD DSN=sortdsn.formc,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SYSDISC  DD DSN=discard.formc,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SYSMAP   DD DSN=sg100.map,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SYSERR   DD DSN=sg100.err,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 LOAD DATA INDDN(SYSREC)
      REPLACE
      INTO TABLE OAMADMIN.CBR_MGT_CLASS_TBL
    (ODMCNUM    POSITION(xx)     INTEGER EXTERNAL(2),
     ODMCNAME   POSITION(xx)     VARCHAR)
 /*
//*********************************************************************
//CBRSG103 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='db2',
//         REGION=4096K
//STEPLIB  DD DSN=SYS1.db2xx.VxRxM0.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSREC   DD DSN=input.sctable,DISP=(OLD,KEEP)
//SYSUT1   DD DSN=workdsn.forsc,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SORTOUT  DD DSN=sortdsn.forsc,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SYSDISC  DD DSN=discard.forsc,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SYSMAP   DD DSN=sg100.map,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SYSERR   DD DSN=sg100.err,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 LOAD DATA INDDN(SYSREC)
      REPLACE
      INTO TABLE OAMADMIN.CBR_STO_CLASS_TBL
    (ODSCNUM    POSITION(xx)     INTEGER EXTERNAL(2),
     ODSCNAME   POSITION(xx:yy)  VARCHAR)
 /*
//*********************************************************************
//*  Combine GROUPxx.OSM_OBJ_DIR tables from different
//*  DB2 systems for object storage group.
//*********************************************************************
//CBRSG104 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='db2',
//         REGION=4096K
//STEPLIB  DD DSN=SYS1.db2xx.VxRxM0.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSREC   DD DSN=input.objdir,DISP=(OLD,KEEP)
//SYSUT1   DD DSN=workdsn.forodir,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SORTOUT  DD DSN=sortdsn.forodir,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SYSDISC  DD DSN=discard.forodir,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SYSMAP   DD DSN=sg100.map,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SYSERR   DD DSN=sg100.err,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=dbpack
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 LOAD DATA INDDN(SYSREC)
      REPLACE
      INTO TABLE storage_group_hlq.OSM_OBJ_DIR
    (ODVER      POSITION(xx)     CHAR(1),
     ODSIZE     POSITION(xx)     INTEGER EXTERNAL(4),
     ODCREATS   POSITION(xx)     TIMESTAMP EXTERNAL(26),
```

```
     ODEXPDT    POSITION(xx)     DATE EXTERNAL(10),                     22950000
     ODLREFDT   POSITION(xx)     DATE EXTERNAL(10),                     23000000
     ODPENDDT   POSITION(xx)     DATE EXTERNAL(10),                     23050000
     ODMCASDT   POSITION(xx)     DATE EXTERNAL(10),                     23100000
     ODSCNUM    POSITION(xx)     INTEGER EXTERNAL(2),                   23150000
     ODMCNUM    POSITION(xx)     INTEGER EXTERNAL(2),                   23200000
     ODLOCFL    POSITION(xx)     CHAR(1),                               23250000
     ODLSLOC    POSITION(xx)     CHAR(6),                               23300000
     ODSECLOC   POSITION(xx)     INTEGER EXTERNAL(4),                   23350000
     ODBKLOC    POSITION(xx)     CHAR(6),                               23400000
     ODBKSEC    POSITION(xx)     INTEGER EXTERNAL(4),                   23450000
     ODCLID     POSITION(xx)     INTEGER EXTERNAL(4),                   23500000
     ODNAME     POSITION(xx:yy)  VARCHAR,                               23525000
     ODBK2LOC   POSITION(xx)     CHAR(6),                               23550000
     ODBK2SEC   POSITION(xx)     INTEGER EXTERNAL(4),                   23575000
     ODLOBFL    POSITION(xx)     CHAR(1),                               23600000
     ODSTATF    POSITION(xx)     INTEGER EXTERNAL(2),                   23625000
     ODRETDT    POSITION(xx)     DATE EXTERNAL(10),                     23650000
     ODINSTID   POSITION(xx)     INTEGER EXTERNAL(4))                   23675000
/*                                                                      23700000
//********************************************************************* 23750000
//*  Combine GROUPxx.OSM_04K_OBJ_DIR tables from different              23800000
//*  DB2 systems for object storage group.                              23850000
//********************************************************************* 23900000
//CBRSG105 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='db2',        23925000
//          REGION=4096K                                                23950000
//STEPLIB  DD DSN=SYS1.db2xx.VxRxM0.SDSNLOAD,DISP=SHR                   23975000
//SYSTSPRT DD SYSOUT=*                                                  24000000
//SYSREC   DD DSN=input.obj04k,DISP=(OLD,KEEP)                          24250000
//SYSUT1   DD DSN=workdsn.for04k,DISP=(MOD,KEEP),UNIT=3390,             24268100
//          VOL=SER=dbpack                                              24286200
//SORTOUT  DD DSN=sortdsn.for04k,DISP=(MOD,KEEP),UNIT=3390,             24304300
//          VOL=SER=dbpack                                              24322400
//SYSDISC  DD DSN=discard.for04k,DISP=(MOD,KEEP),UNIT=3390,             24340500
//          VOL=SER=dbpack                                              24358600
//SYSMAP   DD DSN=sg100.map,DISP=(MOD,KEEP),UNIT=3390,                  24376700
//          VOL=SER=dbpack                                              24394800
//SYSERR   DD DSN=sg100.err,DISP=(MOD,KEEP),UNIT=3390,                  24412900
//          VOL=SER=dbpack                                              24431000
//SYSPRINT DD SYSOUT=*                                                  24450000
//UTPRINT  DD SYSOUT=*                                                  24475000
//SYSUDUMP DD SYSOUT=*                                                  24500000
//SYSIN    DD *                                                         24550000
                                                                        24600000
 LOAD DATA INDDN(SYSREC)                                                24650000
      REPLACE                                                           24700000
      INTO TABLE storage_group_hlq.OSM_04K_OBJ_TBL                      24750000
    (OTVER      POSITION(xx)     CHAR(1),                               24800000
     OTSEG      POSITION(xx)     INTEGER EXTERNAL(2),                   24850000
     OTCLID     POSITION(xx)     INTEGER EXTERNAL(3),                   24900000
     OTNAME     POSITION(xx:yy)  VARCHAR,                               24950000
     OTOBJ      POSITION(xx:yy)  VARCHAR)                               25000000
/*                                                                      25150000
//********************************************************************* 25200000
//*  Combine GROUPxx.OSM_32K_OBJ_DIR tables from different              25250000
//*  DB2 systems for object storage group.                              25300000
//********************************************************************* 25350000
//CBRSG106 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='db2',        25375000
//          REGION=4096K                                                25400000
//STEPLIB  DD DSN=SYS1.db2xx.VxRxM0.SDSNLOAD,DISP=SHR                   25425000
//SYSTSPRT DD SYSOUT=*                                                  25450000
//SYSREC   DD DSN=input.obj32k,DISP=(OLD,KEEP)                          25470800
//SYSUT1   DD DSN=workdsn.for32k,DISP=(MOD,KEEP),UNIT=3390,             25491600
//          VOL=SER=dbpack                                              25512400
//SORTOUT  DD DSN=sortdsn.for32k,DISP=(MOD,KEEP),UNIT=3390,             25533200
//          VOL=SER=dbpack                                              25554000
//SYSDISC  DD DSN=discard.for32k,DISP=(MOD,KEEP),UNIT=3390,             25574800
//          VOL=SER=dbpack                                              25595600
//SYSMAP   DD DSN=sg100.map,DISP=(MOD,KEEP),UNIT=3390,                  25616400
//          VOL=SER=dbpack                                              25637200
//SYSERR   DD DSN=sg100.err,DISP=(MOD,KEEP),UNIT=3390,                  25658000
//          VOL=SER=dbpack                                              25678800
//SYSPRINT DD SYSOUT=*                                                  25700000
//UTPRINT  DD SYSOUT=*                                                  25725000
//SYSUDUMP DD SYSOUT=*                                                  25750000
//SYSIN    DD *                                                         25800000
                                                                        25850000
 LOAD DATA INDDN(SYSREC)                                                25900000
      REPLACE                                                           25950000
      INTO TABLE storage_group_hlq.OSM_32K_OBJ_TBL                      26000000
    (OTVER      POSITION(xx)     CHAR(1),                               26050000
     OTSEG      POSITION(xx)     INTEGER EXTERNAL(2),                   26100000
```

```
     OTCLID     POSITION(xx)     INTEGER EXTERNAL(3),                      26150000
     OTNAME     POSITION(xx:yy)  VARCHAR,                                  26200000
     OTOBJ      POSITION(xx:yy)  VARCHAR)                                  26250000
/*                                                                         26400000
```

Application plans

DB2 BIND (CBRxBIND) and GRANT (CBRxGRNT) jobs are provided to create and authorize the DB2 application plans necessary for OSR, OSMC, LCS, and ISMF.

CBRPBIND

CBRPBIND performs a DB2 BIND for the packages that are needed to access the OAM object storage group tables. The use of DB2 packages allows user defined qualifiers for the object storage group table definitions. For release-to-release consistency, the job provides binds packages for 100 object storage groups (GROUP00–GROUP99). For z/OS DFSMS, this job must be modified to match your installation requirements and must be run before executing CBRABIND, CBRHBIND, or CBRIBIND. In a classic OAM configuration only one job is needed. In a multiple OAM configuration, a unique job is needed for each Object OAM instance with each specifying the DB2 SSID, DB2 location name, and BIND statements for the storage groups that will be used by that instance. The following modifications need to be made:

```
//*    Change the name in the DSN SYSTEM(db2) statement in each step
//*    to the name of the DB2 subsystem in your installation.
//*
//*    Change the location loc in the package names to
//*    the DB2 location name for the DB2 subsystem. If the
//*    default location LOC1 is being used in a classic
//*    OAM configuration, the “loc.” prefix can be removed.
//*
//*    Add, remove, or modify the PACKAGE and QUALIFIER names
//*    in each of the first three steps to match the storage
//*    groups for your installation. In a multiple OAM configuration,
//*    each job only needs statements for the Object storage groups
//*    that are used by that OAM instance. Note that there are
//*    BIND statements for many different MEMBER names for each storage
//*    group, so ensure that all of them are added, changed, or removed
//*    for each storage group.
//*
//*    If you are not using optical storage and have deleted or never
//*    created the DB2 tables that are specific to optical processing
//*    (OLIBRARY, DRIVE, SLOT, VOLUME, DELOBJT), the BIND statements
//*    for packages CBRKCMD, CBRKCME, CBRKCMF, CBRKCMI, CBRKCMR, and
//*    CBRKISQL must be removed from the MISCPKG step to prevent bind errors.
```

CBRIBIND and CBRIGRNT

The CBRIBIND and CBRIGRNT SAMPLIB jobs create the OSR application plan, bind it to DB2, and grant authority for the plan to be used. Run these SAMPLIB jobs if you do not plan to create the OAM configuration database, but you do plan to store objects without starting the OAM address space. If you plan on creating the OAM configuration database and start the OAM address space, use CBRABIND and CBRAGRNT in place of these SAMPLIB members.

CBRHBIND and CBRHGRNT

The CBRHBIND and CBRHGRNT SAMPLIB jobs create the OSMC application plans, bind them to DB2, and grant authority for the plans to be used. Run these SAMPLIB jobs if you plan to start the OAM address space with OSMC.

CBRABIND and CBRAGRNT

The CBRABIND and CBRAGRNT SAMPLIB jobs create the LCS, OSR, and ISMF application plans, bind them to DB2, and grant authority for the plans to be used. Run these SAMPLIB jobs if you plan to create the OAM configuration database and start the OAM address space for object storage.

```
//CBRABIND JOB MSGLEVEL=(1,1),MSGCLASS=A
//******************************************************************
//*
//*  $SEG(CBRABIND) COMP(LCS) PROD(OAM):
//*
//*  OAM DB2 BIND Application Plan Job (for LCS and OSR).
//*
//*  Licensed Materials - Property of IBM
//*  5650-ZOS
//*  COPYRIGHT IBM CORP. 1989, 2017
//*
//*  This job will perform a DB2 BIND for the LCS application
//*  plans and the OSR application plans.
//*
//*  BEFORE RUNNING THIS JOB: CBRPBIND must be run to BIND the
//*  necessary packages to DB2 and DBRMs so that the following
//*  BINDs for the PLANs can find the PACKAGE definitions used
//*  by this job.
//*
//*  Prior to executing this job you need to make the
//*  following modifications:
//*
//*  1. Change the name in the DSN SYSTEM(db2) statement to       @P3C
//*     the name of the DB2 Subsystem in your installation.
//*
//*     In a multiple OAM configuration, you will create a        @P3A
//*     unique job for each Object OAM instance and specify       @P3A
//*     the DB2 SSID for the DB2 subsystem associated with        @P3A
//*     the Object OAM instance.                                  @P3A
//*                                                               @P3A
//*  2. Change the data set name in the DBRMLIB DD statement to
//*     the name of your installation's DBRM library.
//*
//*  3. In step OSRBIND, you have 2 options:
//*    A- In the PKLIST leave the package CBRIDBS prefixed with an
//*       asterisk '*' to indicate all packages defined in the
//*       job CBRPBIND for your installation
//*
//*       -or-
//*
//*    B- Include in the PKLIST all packages defined in the job
//*       CBRPBIND for your installation, followed by '.CBRIDBS' for
//*       the plan used by the OSR component of OAM.
//*       (E.G., GROUP00.CBRIDBS,GROUP01.CBRIDBS,OAMGRPXX.CBRIDBS)
//*
//*  4. In step CBRUTIL, you have 2 options:
//*    A- In the PKLIST leave the package CBRHDBS prefixed with an
//*       asterisk '*' to indicate all packages defined in the
//*       job CBRPBIND for your installation
//*
//*       -or-
//*
//*    B- Include in the PKLIST all packages defined in the job
//*       CBRPBIND for your installation, followed by '.CBRHDBS' for
//*       the plan used by the LCS component of OAM.
//*       (E.G., GROUP00.CBRHDBS,GROUP01.CBRHDBS,OAMGRPXX.CBRHDBS)
//*
//*
//*  A WARNING message will be issued when these BINDs are
//*  executed the first time, indicating the PLANs did not
//*  previously exist but have been added. This WARNING can
//*  be ignored.
//*
//*  CHANGE ACTIVITY:
//*    $L0=JDP3227 320 890601 TUCJRL: Initial Release
//*
//*    $L1=JDP3331 331 910614 TUCKSG: Added MEMBERs CBRKCME and
//*                                   CBRKCMF to PLAN CBROAM. Added
//*                                   VALIDATE(BIND) and RETAIN to
//*                                   all PLANs.
//*
//*    $01=OY61029  110 930105 TUCLJT: Copy CBRIBIND into CBRABIND
//*                                   CBRIBIND will only be required
//*                                   when running a DASD only user
```

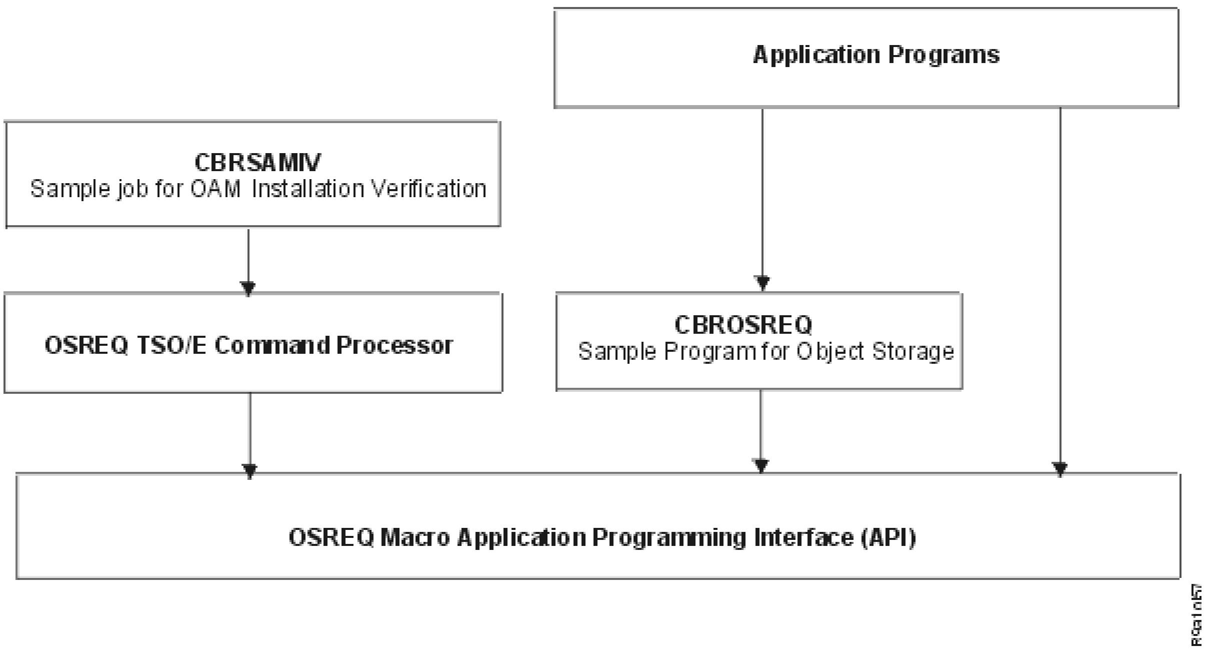
```
//*                                     environment (the OAM address
//*                                     space not started).
//*                                     CBRABIND will be required
//*                                     when running with an optical
//*                                     or tape configuration.
//*
//*    $P1=KB10010 120 930626 TUCFJK: Added member CBRKCMT to PLAN
//*                                     CBROAM for Object Tape support.
//*
//*    $P2=K130950 130 950110 TUCTNN: Corrected inconsistant names.
//*
//*    $L2=HDZ11E0 150 970812 TUCLJT: Rewrite using package sets for
//*                                     existing GROUP00-GROUP99 HLQs
//*    $RL=OW29956 140 970927 TUCSPP: Add new Plan CBRUTIL
//*    $02=OW46082 160 000905 TUCSMC: UPDATE FOR PACKAGE VERSIONS
//*    $L3=HDZ11G0 R13 010205 TUCLJT: Simplify job, genericizing hlqs
//*    $L4=OAMLOB2 R1A 060807 TUCJRA: Added member CBRIEDBS to plan
//*                                     CBRIDBS.
//*    $L5=OAMR1D  R1D 100223 TUCDEW: Add CBRKFDTA to plan CBROAM and
//*                                     CBRKFDTB to plan CBRIDBS
//*    $P3=130050  R23 160816 TUCAED: Missing prolog multi reference
//*    $P4=138323  R23 161101 TUCAED: Added CBRIDBSV package and
//*                                     updated prolog
//*
//*********************************************************************
//OAMPLAN  EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//DBRMLIB  DD  DISP=SHR,
//             DSN=SYS1.CBRDBRM
//SYSTSPRT DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD  *
  DSN SYSTEM(db2)
  BIND PLAN(CBROAM)                                  -
       PKLIST(CBRKCMR.*,CBRKCMD.*,CBRKCMI.*,         -
              CBRKCME.*,CBRKCMF.*,CBRKCMT.*,         -
              CBRKFDTA.*)                            -
       ACTION(REP)                                   -
       VALIDATE(BIND)                                -
       ISOLATION(CS)                                 -
       RETAIN
  BIND PLAN(CBRISMF)                                 -
       PKLIST(CBRKISQL.*)                            -
       ACTION(REP)                                   -
       VALIDATE(BIND)                                -
       ISOLATION(CS)                                 -
       RETAIN
  END
/*
//OSRBIND EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//DBRMLIB  DD  DSN=SYS1.CBRDBRM,DISP=SHR
//SYSTSPRT DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD  *
  DSN SYSTEM(DB2)
  BIND PLAN(CBRIDBS)                                 -
       PKLIST(*.CBRIDBS,*.CBRIEDBS,CBRIDBSV.*,       -
       CBRKCMF.*,CBRKFDTB.*,CBRHTBSV.*)              -
       ACTION(REP)                                   -
       VALIDATE(BIND)                                -
       ISOLATION(CS)                                 -
       ACQUIRE(USE)                                  -
       RELEASE(COMMIT)                               -
       RETAIN
  END
/*
//CBRUTIL EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//DBRMLIB  DD  DSN=SYS1.CBRDBRM,DISP=SHR
//SYSTSPRT DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD  *
  DSN SYSTEM(DB2)
  BIND PLAN(CBRUTIL)                                 -
       PKLIST(*.CBRHDBS,                             -
       CBRHCLLU.*)                                   -
       ACTION(REP)                                   -
       VALIDATE(BIND)                                -
       ISOLATION(CS)                                 -
       ACQUIRE(USE)                                  -
       RELEASE(COMMIT)                               -
```

```
        RETAIN
   END
 /*
```

OAM installation verification program and OAMUTIL

Sample jobs are provided for the invocation of the OAM installation verification program using TSO/E and the OAM utility, OAMUTIL, to reformat 3995 optical cartridges or to modify the default storage class and/or the default management class of a collection. CBRSAMIV is the sample job for the OAM Installation Verification Program (OAMIVP).

Installation verification ensures that the proper environment exists for applications to utilize the OSREQ Macro Application Programming Interface (API).



The previous diagram illustrates a number of different ways in which you may directly or indirectly invoke the OSREQ API:

- A TSO/E command interface (OSREQ TSO/E Command Processor) provides a simple mechanism to invoke individual OSREQ API functions. The usually required OSREQ ACCESS and UNACCESS functions are performed automatically within the OSREQ TSO/E command processor around the individual function to be performed. The OSREQ TSO/E command processor does not provide the full functionality of the OSREQ API, and you cannot store real data using the OSREQ TSO/E command processor STORE function; otherwise dummy data is created for the object. You can use the OSREQ TSO/E command processor as a simple mechanism to perform some other operations on real object data without having to write an application program. You can use the OSREQ TSO/E command processor to compare the primary copy of an object with a backup copy of the object, for example, with the intent of ensuring both copies match to meet audit requirements. The OSREQ TSO/E command processor can be invoked in a native TSO environment at the READY prompt or within a CLIST, from ISPF, or by invoking the TSO/E program IKJEFT01 in the background through a batch job and providing OSREQ TSO/E commands as an input. For the OSREQ TSO/E command processor syntax, see "OSREQ TSO/E command syntax" on page 243.
- The CBRSAMIV sample job is provided to perform OAM installation verification. Within the CBRSAMIV job, the TSO/E program IKJEFT01 is used to provide OSREQ TSO/E commands to store an object and perform several other operations on the object and ultimately deleting the object. This provides a simple mechanism to verify that the OAM installation has been performed successfully and that the OSREQ API invocations are working as expected. The CBRSAMIV sample job is described later in this section.

- The CBROSREQ sample contains an assembler program that is provided as an example to illustrate how to invoke the individual OSREQ API functions. Application developers, however, can also assemble, link-edit, and then by directly calling the sample program, execute the sample program as an interface to the OSREQ API. For the description of the CBROSREQ sample program, see *z/OS DFSMS OAM Application Programmer's Reference*.
- Application developers can also develop assembler programs that directly invoke the OSREQ API. For the syntax for the OSREQ API, see *z/OS DFSMS OAM Application Programmer's Reference*.

CBRSAMIV

The OSREQ command is a TSO/E command processor. OSREQ can be invoked for any of the following situations:

- At the TSO/E READY prompt when logged on to TSO/E
- Under option 6 "Command" under ISPF/PDF
- On the ISPF command line by prefacing the OSREQ command with the characters "TSO"
- From within a TSO/E CLIST
- From a batch job, by invoking the TSO/E terminal monitor program (TMP) in the batch job

SAMPLIB member CBRSAMIV invokes the OSREQ macro from an MVS batch job, which invokes the TSO/E terminal monitor program (IKJEFT01) that runs in the background. This job allows you to verify the installation of OAM to ensure object storage success.

```
//CBRSAMIV  JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)                       00050000
//********************************************************************* 00100000
//*                                                                   * 00150000
//*  $SEG(CBRSAMIV) COMP(OSR) PROD(OAM):                              * 00200000
//*                                                                   * 00250000
//* THIS SAMPLE JOB INVOKES THE OSREQ COMMAND PROCESSOR TO PERFORM    * 00300000
//* THE FOLLOWING ACTIONS:                                            * 00350000
//*                                                                     00350001
//*  Licensed Materials - Property of IBM                               00350002
//*  5650-ZOS                                                           00350003
//*  COPYRIGHT IBM CORP. 1994, 2017                                     00350004
//*                                                                   * 00400000
//* 1. STORE A 1 MB (1048576) OBJECT.                                 * 00450000
//*                                                                   * 00500000
//* 2. ISSUE AN OSREQ QUERY TO LIST THE OAM DIRECTORY INFORMATION.    * 00550000
//*                                                               3@L1D* 00600000
//* 3. ISSUE AN OSREQ CHANGE REQUEST TO CHANGE THE RETENTION      @L1C* 00800000
//*    PERIOD ASSOCIATED WITH THE OBJECT TO 365 DAYS.             @L1C* 00850000
//*                                                                   * 00900000
//* 4. ISSUE AN OSREQ QUERY TO LIST THE OAM DIRECTORY INFORMATION.@L1C* 00950000
//*                                                                   * 01000000
//* 5. ISSUE AN OSREQ RETRIEVE TO RETRIEVE THE PRIMARY COPY OF    @L1C* 01050000
//*    THE OBJECT AND CHECK TO SEE IF IT CONTAINS THE PRE-DEFINED @L1C* 01100000
//*    DATA.                                                      @L1A* 01100001
//*                                                                   * 01150000
//* 6. ISSUE AN OSREQ DELETE REQUEST TO DELETE THE OBJECT.        @L1C* 01200000
//*                                                                   * 01250000
//*                                                                   * 01300000
//* PRIOR TO EXECUTING THIS JOB YOU MAY NEED TO MAKE THE FOLLOWING    * 01305000
//* MODIFICATIONS:                                                    * 01310000
//*                                                                   * 01315000
//*  1. CHANGE THE JOB CARD PER YOUR INSTALLATION REQUIREMENTS.       * 01320000
//*                                                                   * 01325000
//*  2. CHANGE THE collection NAME, object NAME AND LENGTH OF     @L1C* 01330000
//*     OBJECT TO CONFORM TO YOUR INSTALLATION REQUIREMENTS.          * 01335000
//*                                                                   * 01335001
//*  3. IN A MULTIPLE OAM SUBSYSTEM AND OPTIONALLY ASSOCIATED     @L1A* 01335002
//*     OAM ADDRESS SPACE CONFIGURATION.                          @L1A* 01340000
//*                                                               @L1A* 01340001
//*     SPECIFICATION OF A DB2ID(db2) KEYWORD IS REQUIRED ON ALL  @L1A* 01340002
//*     OSREQ FUNCTIONS, WHERE db2 IS THE DB2 SSID OR GROUP       @L1A* 01340003
//*     ATTACHMENT NAME FOR THE DB2 SUBSYSTEM ASSOCIATED WITH THE @L1A* 01340004
//*     OBJECT OAM INSTANCE.                                      @L1A* 01340005
//*                                                               @L1A* 01340006
//*     IF SPECIFIED IN A CLASSIC OAM CONFIGURATION IT MUST MATCH @L1A* 01340007
//*     ANY DB2 SSID CONFIGURED FOR OAM.                          @L1A* 01340009
//*                                                                   * 01340010
//*  CHANGE ACTIVITY:                                                 * 01350000
//*    $L0=OBJTAPE 120 930525 TUCFJK: INITIAL RELEASE OF THE OAM IVP  * 01400000
```

```
//*                                  SAMPLE JCL.                     * 01450000
//*    $01=OW02037 120 940201 TUCFJK: ADDED DOCUMENTATION NOTE THAT   * 01460000
//*                                  JCL MAY NEED MODIFICATION PER   * 01470000
//*                                  INSTALLATION REQUIREMENT.       * 01480000
//*                                  (PTM # K120990)                 * 01490000
//*    $P1=K190248 R19 060815 TUCYHL: CHANGE JOB CARD OAMIVP TO       * 01510000
//*                                  CBRSAMIV                        * 01530000
//*    $L1=OAMR23M R23 160405 TUCAED: Multi OAM AS Support        @L1A* 01530001
//******************************************************************* 01550000
//STEP1    EXEC PGM=IKJEFT01,REGION=4096K                             01600000
//SYSPRINT DD   SYSOUT=*                                              01650000
//SYSTSPRT DD   SYSOUT=*                                              01700000
//SYSTSIN  DD   *                                                     01750000
 OSREQ STORE    collection object LENGTH(1048576) DB2ID(db2)          01800000
 OSREQ QUERY    collection object DB2ID(db2)                          01850000
 OSREQ CHANGE   collection object RP(365) DB2ID(db2)                  01950000
 OSREQ QUERY    collection object DB2ID(db2)                          02000000
 OSREQ RETRIEVE collection object COMPARE VIEW(PRIMARY) DB2ID(db2)    02050000
 OSREQ DELETE   collection object DB2ID(db2)                          02100000
/*                                                                    02150000
```

CBRSAMUT

CBRSAMUT, as shown here, uses the OAMUTIL utility to reformat a 3995 optical disk volume or both sides of a 3995 optical cartridge as well as modifying the default storage class, the default management class of a collection, or both.

For more information on this utility command, see "Reformatting a 3995 optical disk" on page 303.

```
//OAMUTIL  JOB CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1)                      00050000
//******************************************************************* 00100000
//*                                                                 * 00150000
//*  $SEG(CBRSAMUT) COMP(LCS) PROD(OAM):                            * 00200000
//*                                                                 * 00250000
//*                                                                 * 00300000
//* This sample job provides examples of invoking OAMUTIL command    * 00350000
//*      processor to reformat the 3995 optical disk cartridge      * 00375000
//*      and also to change the default management class and/or     * 00400000
//*      storage class of a specified collection.                   * 00425000
//*                                                                 * 00425001
//*  Licensed Materials - Property of IBM                           * 00425002
//*  5650-ZOS                                                       * 00425003
//*  COPYRIGHT IBM CORP. 1998, 2017                                 * 00425004
//*                                                                 * 00450000
//* Following are examples using the reformat keyword:              * 00500000
//*                                                                 * 00550000
//* NOTE: The OAMUTIL REFORMAT command can only be used in a     @L1A* 00550001
//*       classic OAM configuration.                             @L1A* 00583300
//*                                                              @L1A* 00583301
//* example #01A - conditionally reformat one side, ONE is the      * 00583302
//*                default:                                         * 00616600
//*                                                                 * 00650000
//*             OAMUTIL REFORMAT oldvs1                             * 00700000
//*                                                                 * 00708300
//* example #01B - conditionally reformat one side specifying the   * 00716600
//*                ONE keyword explicitly:                          * 00724900
//*                                                                 * 00733200
//*             OAMUTIL REFORMAT oldvs1 ONE                         * 00741500
//*                                                                 * 00750000
//* example #02- conditionally reformat one side on a specific      * 00800000
//*              operator accessible drive:                         * 00850000
//*                                                                 * 00900000
//*             OAMUTIL REFORMAT oldvs1 DRIVENAME(P21D5)            * 00950000
//*                                                                 * 01000000
//* example #03- conditionally reformat one side and rename it      * 01050000
//*              to a new volume serial number:                     * 01100000
//*                                                                 * 01150000
//*             OAMUTIL REFORMAT oldvs1 NEWVOL1(newvs1)             * 01200000
//*                                                                 * 01250000
//* example #04- Conditionally reformat both sides:                 * 01300000
//*                                                                 * 01350000
//*             OAMUTIL REFORMAT oldvs1 BOTH                        * 01400000
//*                                                                 * 01450000
//* example #05- Conditionally reformat both sides and              * 01500000
//*              return them to SCRATCH storage group:              * 01550000
//*                                                                 * 01600000
//*             OAMUTIL REFORMAT oldvs1 BOTH SCRATCH                * 01650000
//*                                                                 * 01700000
```

```
//* example #06- Conditionally reformat both sides on                  * 01750000
//*              a specific operator accessible drive:                 * 01800000
//*                                                                    * 01850000
//*             OAMUTIL REFORMAT oldvs1 BOTH DRIVENAME(P52D1)          * 01900000
//*                                                                    * 01950000
//* example #07- Conditionally reformat both sides and                 * 02000000
//*              rename them to new volume serial numbers:             * 02050000
//*                                                                    * 02100000
//*             OAMUTIL REFORMAT oldvs1 BOTH NV1(newvs1) NV2(newvs2)   * 02150000
//*                                                                    * 02200000
//* example #08- Conditionally reformat both sides,                    * 02250000
//*              return them to SCRATCH storage group, and             * 02300000
//*              rename them to new volume serial numbers:             * 02350000
//*                                                                    * 02400000
//*             OAMUTIL REFORMAT oldvs1 BOTH NV1(newvs1)             + * 02450000
//*                                          NV2(newvs2) SCRATCH       * 02500000
//*                                                                    * 02550000
//* example #09- Conditionally reformat both sides on                  * 02600000
//*              a specific operator accessible drive,                 * 02650000
//*              return them to SCRATCH storage group, and             * 02700000
//*              rename them to new volume serial numbers:             * 02750000
//*                                                                    * 02800000
//*             OAMUTIL REFORMAT oldvs1 BOTH NV1(newvs1)             + * 02850000
//*                                     NV2(newvs2) D(P52D1) SCRATCH   * 02900000
//*                                                                    * 02950000
//* example #10- Unconditionally reformat one side:                    * 03000000
//*                                                                    * 03050000
//*             OAMUTIL REFORMAT oldvs1 FORCE                          * 03100000
//*                                                                    * 03150000
//* example #11- Unconditionally reformat one side on a specific       * 03200000
//*              operator accessible drive:                            * 03250000
//*                                                                    * 03300000
//*             OAMUTIL REFORMAT oldvs1 DRIVENAME(P21D5) FORCE         * 03350000
//*                                                                    * 03400000
//* example #12- Unconditionally reformat one side and rename it       * 03450000
//*              to a new volume serial number:                        * 03500000
//*                                                                    * 03550000
//*             OAMUTIL REFORMAT oldvs1 NEWVOL1(newvs1) FORCE          * 03600000
//*                                                                    * 03650000
//* example #13- Unconditionally reformat both sides:                  * 03700000
//*                                                                    * 03750000
//*             OAMUTIL REFORMAT oldvs1 BOTH FORCE                     * 03800000
//*                                                                    * 03850000
//* example #14- Unconditionally reformat both sides and               * 03900000
//*              return them to SCRATCH storage group:                 * 03950000
//*                                                                    * 04000000
//*             OAMUTIL REFORMAT oldvs1 BOTH SCRATCH FORCE             * 04050000
//*                                                                    * 04100000
//* example #15- Unconditionally reformat both sides on                * 04150000
//*              a specific operator accessible drive:                 * 04200000
//*                                                                    * 04250000
//*             OAMUTIL REFORMAT oldvs1 BOTH DRIVENAME(P52D1) FORCE    * 04300000
//*                                                                    * 04350000
//* example #16- Unconditionally reformat both sides and               * 04400000
//*              rename them to new volume serial numbers:             * 04450000
//*                                                                    * 04500000
//*       OAMUTIL REFORMAT oldvs1 BOTH NV1(newvs1) NV2(newvs2) FORCE   * 04550000
//*                                                                    * 04600000
//* example #17- Unconditionally reformat both sides,                  * 04650000
//*              return them to SCRATCH storage group, and             * 04700000
//*              rename them to new volume serial numbers:             * 04750000
//*                                                                    * 04800000
//*             OAMUTIL REFORMAT oldvs1 BOTH NV1(newvs1)             + * 04850000
//*                                         NV2(newvs2) SCRATCH FORCE  * 04900000
//*                                                                    * 04950000
//* Following are examples using the CHGCOL keyword:                   * 04975000
//*                                                                    * 05000000
//* example #01- modify the default storage class and management       * 05001800
//*              class of a collection in a classic OAM            @L1C* 05003600
//*              configuration:                                    @L1C* 05003601
//*                                                                    * 05005400
//*   OAMUTIL CHGCOL collection-name                                 + * 05007200
//*                  SGN(storagegroup-name)                          + * 05009000
//*                  OLDSCN(current-default-storageclass-name)       + * 05010800
//*                  NEWSCN(new-default-storageclass-name)           + * 05012600
//*                  OLDMCN(current-default-managementclass-name)    + * 05014400
//*                  NEWMCN(new-default-managementclass-name)          * 05016200
//*                                                                    * 05018000
//* example #02- modify only the default storage class of a            * 05019800
//*              collection in a classic OAM configuration:        @L1C* 05021600
//*                                                                    * 05023400
//*   OAMUTIL CHGCOL collection-name                                 + * 05025200
```

```
//*                 SGN(storagegroup-name)                       + * 05027000
//*                 OLDSCN(current-default-storageclass-name)    + * 05028800
//*                 NEWSCN(new-default-storageclass-name)          * 05030600
//*                                                                * 05032400
//* example #03- modify only the default management class of a     * 05034200
//*             collection in a classic OAM configuration:      @L1C* 05036000
//*                                                                * 05037800
//*   OAMUTIL CHGCOL collection-name                             + * 05039600
//*                 SGN(storagegroup-name)                       + * 05041400
//*                 OLDMCN(current-default-managementclass-name) + * 05043200
//*                 NEWMCN(new-default-managementclass-name)       * 05045000
//*                                                                * 05046800
//* example #04- modify only the default management class of a  @L1A* 05046801
//*             collection in a multiple OAM configuration:     @L1A* 05046802
//*                                                             @L1A* 05046803
//*   Note: DB2ID keyword added                                 @L1A* 05046804
//*                                                             @L1A* 05046805
//*   OAMUTIL CHGCOL collection-name                             + * 05046806
//*                 DB2ID(id)                                    + * 05046807
//*                 SGN(storagegroup-name)                       + * 05046808
//*                 OLDMCN(current-default-managementclass-name) + * 05046809
//*                 NEWMCN(new-default-managementclass-name)    @L1A* 05046810
//*                                                             @L1A* 05050000
//* PRIOR TO EXECUTING THIS JOB YOU MAY NEED TO MAKE THE FOLLOWING * 05050001
//* MODIFICATIONS:                                                 * 05100000
//*                                                                * 05150000
//*  1. CHANGE THE JOB CARD PER YOUR INSTALLATION REQUIREMENTS.    * 05200000
//*                                                                * 05250000
//*  2. FOR REFORMAT:                                              * 05259300
//*     1) CHANGE THE VOLUME SERIAL NUMBER(S),                     * 05268600
//*        OPERATOR ACCESSIBLE DRIVE TO CONFORM TO YOUR INSTALLAION * 05277900
//*        REQUIREMENTS.                                           * 05287200
//*     2) DELETE OR COMMENT OUT STEP2, WHICH PERTAINS TO CHGCOL.  * 05296500
//*                                                                * 05305800
//*  3. FOR CHGCOL:                                                * 05315100
//*     1) CHANGE THE COLLECTION NAME (collection-name)            * 05324400
//*        STORAGE GROUP NAME (SGN),                               * 05333700
//*        OLD STORAGE CLASS NAME (OLDSCN),                        * 05343000
//*        NEW STORAGE CLASS NAME (NEWSCN),                        * 05352300
//*        OLD MANAGEMENT CLASS NAME (OLDMCN),                     * 05361600
//*        NEW MANAGEMENT CLASS NAME (NEWMCN) TO CONFORM TO YOUR   * 05370900
//*        INSTALLATION REQUIREMENTS.                              * 05380200
//*     2) IN A MULTIPLE OAM CONFIGURATION, ADD DB2ID           @L1A* 05380201
//*     3) DELETE OR COMMENT OUT STEP1, WHICH PERTAINS TO       @L1C* 05389500
//*        REFORMAT.                                            @L1A* 05389501
//*                                                                * 05400000
//*                                                                * 05450000
//*  CHANGE ACTIVITY:                                              * 05500000
//*    $RF=OW30138 140 971111 TUCSPP: INITIAL Reformat APAR        * 05550000
//*    $00=OAMR1B  R1B 080303 TUCBJF: OAMR1B CHGCOL                * 05560000
//*    $L1=OAMR23M R23 160405 TUCAED: Multi OAM AS Support      @L1A* 05560002
//*                                                                * 05570000
//****************************************************************** 05580000
//*  STEP1    REFORMAT:                                              05590000
//*                                                                * 05600000
//****************************************************************** 05650000
//STEP1    EXEC PGM=IKJEFT01,REGION=4096K                            05700000
//SYSPRINT DD   SYSOUT=*                                             05750000
//SYSTSPRT DD   SYSOUT=*                                             05800000
//SYSTSIN  DD   *                                                    05850000
 OAMUTIL REFORMAT oldvs1 NV1(newvs1)                                 05900000
 OAMUTIL REFORMAT oldvs1 BOTH NV1(newvs1) NV2(newvs2) D(P52D1)       05950000
 OAMUTIL REFORMAT oldvs1                                            +06000000
                  NV1(newvs1) NV2(newvs2) D(P21D5) SCRATCH FORCE     06050000
//****************************************************************** 06052100
//*  STEP2    CHGCOL:                                                06054200
//*                                                                  06056300
//STEP2    EXEC PGM=IKJEFT01,REGION=4096K                            06058400
//****************************************************************** 06060500
//SYSPRINT DD   SYSOUT=*                                             06062600
//SYSTSPRT DD   SYSOUT=*                                             06064700
//SYSTSIN  DD   *                                                    06066800
 OAMUTIL CHGCOL collection-name                                     +06068900
                 SGN(storagegroup-name)                             +06071000
                 OLDSCN(current-default-storageclass-name)          +06073100
                 NEWSCN(new-default-storageclass-name)              +06075200
                 OLDMCN(current-default-managementclass-name)       +06077300
                 NEWMCN(new-default-managementclass-name)            06079400
 OAMUTIL CHGCOL collection-name                                     +06081500
                 SGN(storagegroup-name)                             +06083600
                 OLDSCN(current-default-storageclass-name)          +06085700
                 NEWSCN(new-default-storageclass-name)               06087800
```

```
 OAMUTIL CHGCOL collection-name                                      +06089900
                SGN(storagegroup-name)                               +06092000
                OLDMCN(current-default-managementclass-name)         +06094100
                NEWMCN(new-default-managementclass-name)              06096200
/*                                                                    06100000
```

Automatic class selection

SAMPLIB members CBRHSC, CBRHMC, and CBRHSG are sample automatic class selection routines for the OAM environments.

CBRHSC

SAMPLIB member CBRHSC, as shown here, provides a storage class ACS routine for STORE, CHANGE, and CTRANS.

```
/*******************************************************************/ 00050000
/*                                                                 */ 00100000
/* $SEG(CBRHSC) COMP(OSMC) PROD(OAM):                              */ 00150000
/*                                                                 */ 00200000
/* OAM Sample Storage Class ACS Routine                            */ 00250000
/*                                                                 */ 00250001
/*  Licensed Materials - Property of IBM                           */ 00250002
/*  5650-ZOS                                                       */ 00250003
/*  COPYRIGHT IBM CORP. 1989, 1992                                 */ 00250004
/*                                                                 */ 00300000
/* FUNCTION: SUPPLY A STORAGE CLASS FOR OAM OBJECTS                */ 00350000
/*                                                                 */ 00400000
/* OPERATION: Supply a storage class for the following environments: */ 00450000
/*                                                                 */ 00500000
/*             STORE   - Assign initial storage class of DASD or   */ 00550000
/*                       optical library based on collection name. */ 00600000
/*             CHANGE  - The storage class of an object has been   */ 00650000
/*                       requested to change.                      */ 00700000
/*             CTRANS  - The object is moved in the hierarchy      */ 00750000
/*                       according to management class.            */ 00800000
/*                                                                 */ 00850000
/* NOTES:     In this implementation, the collection name is used as */ 00900000
/*            the basis for determining whether explicit values will */ 00950000
/*            be considered. This approach and all of the processing */ 01000000
/*            indicated below is one of many differing possibilities */ 01050000
/*            and is only for purposes of illustration to demonstrate*/ 01100000
/*            the types of processing that can be accomplished in an */ 01150000
/*            ACS routine. Actual implementations will vary.         */ 01200000
/*                                                                 */ 01250000
/* ASSUMPTIONS:                                                    */ 01300000
/*   Collection name format                                        */ 01350000
/*     CLLCT0mn                                                    */ 01400000
/*       where m = 0 or 1 (0 indicates that explicit values will be */ 01450000
/*                         ignored, 1 indicates that explicit values */ 01500000
/*                         are considered in some cases and may     */ 01550000
/*                         result in an override)                   */ 01600000
/*       where n = 0 - 9                                           */ 01650000
/*   Object name format - has 5 levels as follows:                 */ 01700000
/*     xxxx.xxx.xxxxxxxx.xxxxxxx.xxx                               */ 01750000
/*   Valid storage classes:                                        */ 01800000
/*     DB2DASD  - DASD                                             */ 01850000
/*     OLIBRARY - optical                                          */ 01900000
/*     TAPESC   - tape                                             */ 01950000
/*                                                                 */ 02000000
/*   CHANGE ACTIVITY:                                              */ 02050000
/*     $L0=JDP3227 320 890601 TUCKSG: Initial Release              */ 02100000
/*     $L1=JDP3331 331 910614 TUCKSG: Added sample definitions     */ 02150000
/*     $00=OW00750 120 931203 TUCSMC: Added tape samples           */ 02175000
/*                                                                 */ 02200000
/*******************************************************************/ 02250000
/*              L O G I C        O V E R V I E W                   */ 02300000
/*                                                                 */ 02350000
/* If STORE environment                                            */ 02400000
/* | If object name not specified (i.e. this is an invocation for  */ 02450000
/* | the entire collection)                                        */ 02500000
/* | | Select                                                      */ 02550000
/* | | When the collection is in the set that we are defining to   */ 02600000
/* | | allow overrides                                             */ 02650000
/* | | | If storage class specified is not 'OLIBRARY'              */ 02700000
```

```
/* | | | | Set the storage class to 'DB2DASD'                        */
/* | | | Endif                                                       */
/* | | When the collection is in the set that we are defining to     */
/* | | not allow overrides                                           */
/* | | | Set the storage class to 'OLIBRARY'                         */
/* | | Otherwise                                                     */
/* | | | Set error code                                              */
/* | | End                                                           */
/* | Else (an object name was specified)                             */
/* | | Select                                                        */
/* | | When the collection is in the set that we are defining to     */
/* | | allow overrides                                               */
/* | | | If storage class specified is not 'OLIBRARY'                */
/* | | | | Set the storage class to 'DB2DASD'                        */
/* | | | Endif                                                       */
/* | | When the collection is in the set that we are defining to     */
/* | | not allow overrides                                           */
/* | | | If the object name has exactly 5 levels and the 5th         */
/* | | | level indicates that the object should have a particular    */
/* | | | storage class                                               */
/* | | | | Set the storage class to 'DB2DASD'                        */
/* | | | Else                                                        */
/* | | | | Set the storage class to 'OLIBRARY'                       */
/* | | | Endif                                                       */
/* | | Otherwise                                                     */
/* | | | Set error code                                              */
/* | | End                                                           */
/* | Endif (object name specified)                                   */
/* Endif (STORE environment)                                         */
/*                                                                   */
/*                                                                   */
/* If CHANGE environment                                             */
/* | If the storage class specified is not a valid storage class     */
/* | | Set error code                                                */
/* | Endif                                                           */
/* Endif (CHANGE environment)                                        */
/*                                                                   */
/*                                                                   */
/* If CLASS_TRANSITION environment                                   */
/* | Select                                                          */
/* | When storage class is 'DB2DASD'                                 */
/* | | Set storage class to 'OLIBRARY'                               */
/* | When storage class is 'OLIBRARY'                                */
/* | | Set storage class to 'TAPESC'                                 */
/* | Otherwise                                                       */
/* | | Set error code                                                */
/* | End                                                             */
/* Endif                                                             */
/*********************************************************************/
/*      S T O R A G E    C L A S S   D E F I N I T I O N S           */
/*                                                                   */
/*Relevant                                                           */
/*Fields    DB2DASD OLIBRARY TAPESC                                  */
/*--------- ------- -------- -------                                 */
/*INITIAL                                                            */
/*ACCESS                                                             */
/*RESPONSE                                                           */
/*SECONDS       0      20      900                                   */
/*                                                                   */
/*SUSTAINED                                                          */
/*DATA                                                               */
/*RATE         n/a      1        3                                   */
/*                                                                   */
/*********************************************************************/


PROC STORCLAS                          /* Select a storage class     */

  FILTLIST BLANK    INCLUDE ('        ','')

/*********************************************************************/
/* STORE                                                             */
/*********************************************************************/
 IF &ACSENVIR = 'STORE' THEN           /* Object is being stored     */
   IF &MEMN = &BLANK THEN              /* If the object name is not
                                          specified (i.e. indicating
                                          an invocation for the entire
                                          collection)                */
     SELECT
     WHEN (&DSN = 'CLLCT010' | &DSN = 'CLLCT011' | &DSN = 'CLLCT012'
         | &DSN = 'CLLCT013' | &DSN = 'CLLCT014' | &DSN = 'CLLCT015'
         | &DSN = 'CLLCT016' | &DSN = 'CLLCT017' | &DSN = 'CLLCT018'
```

```
         | &DSN = 'CLLCT019')          /* For the collections which
                                          allow overrides            */
       IF &STORCLAS ^= 'OLIBRARY' THEN /* If the storage class
                                          specified is not the one
                                          explicit storage class value
                                          that is considered valid
                                          for these collections      */
         SET &STORCLAS = 'DB2DASD'     /* Set the storage class to the
                                          desired value for these
                                          collections                */
     WHEN (&DSN = 'CLLCT000' | &DSN = 'CLLCT001' | &DSN = 'CLLCT002'
         | &DSN = 'CLLCT003' | &DSN = 'CLLCT004' | &DSN = 'CLLCT005'
         | &DSN = 'CLLCT006' | &DSN = 'CLLCT007' | &DSN = 'CLLCT008'
         | &DSN = 'CLLCT009')          /* For the collections which
                                          do not allow overrides     */
       SET &STORCLAS = 'OLIBRARY'      /* Set the storage class to the
                                          desired value for these
                                          collections                */
     OTHERWISE                         /* Otherwise the collection name
                                          is invalid                 */
       EXIT CODE(10)                   /* Indicate that an error
                                          occurred                   */
     END                               /* Select                     */
   ELSE                                /* If the object name is
                                          specified (i.e. indicating
                                          an invocation for the
                                          specific collection)       */
     SELECT
     WHEN (&DSN = 'CLLCT010' | &DSN = 'CLLCT011' | &DSN = 'CLLCT012'
         | &DSN = 'CLLCT013' | &DSN = 'CLLCT014' | &DSN = 'CLLCT015'
         | &DSN = 'CLLCT016' | &DSN = 'CLLCT017' | &DSN = 'CLLCT018'
         | &DSN = 'CLLCT019')          /* For the collections which
                                          allow overrides            */
       IF &STORCLAS ^= 'OLIBRARY' THEN /* If the storage class
                                          specified is not the one
                                          explicit storage class value
                                          that is considered valid
                                          for these collections      */
         SET &STORCLAS = 'DB2DASD'     /* Set the storage class to the
                                          desired value for these
                                          collections                */
     WHEN (&DSN = 'CLLCT000' | &DSN = 'CLLCT001' | &DSN = 'CLLCT002'
         | &DSN = 'CLLCT003' | &DSN = 'CLLCT004' | &DSN = 'CLLCT005'
         | &DSN = 'CLLCT006' | &DSN = 'CLLCT007' | &DSN = 'CLLCT008'
         | &DSN = 'CLLCT009')          /* For the collections which
                                          do not allow overrides     */
       IF &MEMN = *.*.*.*.DZX THEN     /* If the object name contains
                                          a value indicating that this
                                          object should be treated
                                          differently than the
                                          other objects in these
                                          collections                */
         SET &STORCLAS = 'DB2DASD'     /* Set the storage class to the
                                          desired value for these
                                          objects that are treated
                                          differently                */
       ELSE                            /* Otherwise there is nothing
                                          special about this object
                                          name                       */
         SET &STORCLAS = 'OLIBRARY'    /* Set the storage class to the
                                          desired value for the objects
                                          in these collections       */
     OTHERWISE                         /* Otherwise the collection name
                                          is invalid                 */
       EXIT CODE(11)                   /* Indicate that an error
                                          occurred                   */
     END                               /* Select                     */

/*********************************************************************/
/* CHANGE                                                            */
/*********************************************************************/
 IF &ACSENVIR = 'CHANGE' THEN          /* Object is being changed    */
   IF &STORCLAS ^= 'DB2DASD'  AND
      &STORCLAS ^= 'OLIBRARY' AND
      &STORCLAS ^= 'TAPESC'  THEN      /* If the storage class specified
                                          is not a storage class that
                                          is considered valid        */
        EXIT CODE(12)                  /* Indicate that an error
                                          occurred                   */
/*********************************************************************/
/* CLASS TRANSITION                                                  */
/*********************************************************************/
```

```
 IF &ACSENVIR = 'CTRANS' THEN          /* Object is being processed
                                          as a result of a class
                                          transition                  */
   SELECT (&STORCLAS)

   WHEN ('DB2DASD')                    /* If current storage class
                                          indicates that object is on
                                          DASD                        */
     SET &STORCLAS = 'OLIBRARY'        /* Set storage class to indicate
                                          that the object should reside
                                          on optical                  */
   WHEN ('OLIBRARY')                   /* If current storage class
                                          indicates that object is on
                                          optical                     */
     SET &STORCLAS = 'TAPESC'          /* Set storage class to indicate
                                          that the object should reside
                                          on tape                     */

   OTHERWISE                           /* Otherwise the storage class
                                          is invalid                  */
     EXIT CODE(13)                     /* Indicate that an error
                                          occurred                    */
   END
END
```

CBRHMC

SAMPLIB member CBRHMC provides a management class ACS routine for OAM objects.

```
/*******************************************************************/
/*                                                                   */
/* $SEG(CBRHMC) COMP(OSMC) PROD(OAM):                                */
/*                                                                   */
/* OAM Sample Management Class ACS Routine                           */
/*                                                                   */
/*  Licensed Materials - Property of IBM                             */
/*  5650-ZOS                                                         */
/*  COPYRIGHT IBM CORP. 1989, 1992                                   */
/*                                                                   */
/* FUNCTION: SUPPLY A MANAGEMENT CLASS FOR OAM OBJECTS               */
/*                                                                   */
/*  OPERATION: Supply a management class for the following           */
/*             environments:                                         */
/*                                                                   */
/*              STORE   - Assign an initial management class based  */
/*                        on collection name and/or object name     */
/*                        and/or storage class.                     */
/*              CHANGE  - Validate a request to change the          */
/*                        management class for an object.           */
/*              CTRANS  - The object is moved in the hierarchy      */
/*                        according to its previous management      */
/*                        class.                                    */
/*                                                                   */
/* NOTES:     In this implementation, the collection name is used as */
/*            the basis for determining whether explicit values will */
/*            be considered. This approach and all of the processing */
/*            indicated below is one of many differing possibilities */
/*            and is only for purposes of illustration to demonstrate*/
/*            the types of processing that can be accomplished in an */
/*            ACS routine. Actual implementations will vary.         */
/*                                                                   */
/* ASSUMPTIONS:                                                      */
/*   Collection name format                                          */
/*     CLLCT0mn                                                      */
/*       where m = 0 or 1 (0 indicates that explicit values will be  */
/*                        ignored, 1 indicates that explicit values */
/*                        are considered in some cases and may      */
/*                        result in an override)                    */
/*       where n = 0 - 9                                             */
/*   Object name format - has 5 levels as follows:                   */
/*     xxxx.xxx.xxxxxxxx.xxxxxxx.xxx                                 */
/*   Valid storage classes:                                          */
/*     DB2DASD  - DASD                                               */
/*     OLIBRARY - optical                                            */
/*     TAPESC   - Tape                                               */
/*   Valid management classes:                                       */
/*     MAGONLY  - 30 days on DASD, then expire                       */
/*     MAG30D   - 30 days on DASD, then transition                   */
/*     MAG30LIB - 6 months on optical, then transition               */
```

```
/*     TAPSEVEN - 7 years on tape, then expire                         */
/*     OPT6D    - 0 days on DASD, then transition                      */
/*     OPT6LIB  - 6 months on optical, then transition                 */
/*     OPTTAPE  - 7 years on tape, the expire                          */
/*                                                                     */
/*   CHANGE ACTIVITY:                                                  */
/*     $L0=JDP3227 320 890601 TUCKSG: Initial Release                  */
/*     $L1=JDP3331 331 910614 TUCKSG: Added sample definitions         */
/*     $00=OW00750 120 931221 TUCSMC: updated to add Tape samples      */
/*                                                                     */
/***********************************************************************/
/*             L O G I C        O V E R V I E W                        */
/*                                                                     */
/* If STORE environment                                                */
/* | If object name not specified (i.e. this is an invocation for      */
/* | the entire collection)                                            */
/* | | Select                                                          */
/* | | When the collection is in the set that we are defining to       */
/* | | allow overrides                                                 */
/* | | | Select                                                        */
/* | | | When the storage class is 'DB2DASD' (i.e. DASD)               */
/* | | | | If management class specified is not 'MAGONLY'              */
/* | | | | | Set the management class to 'MAG30D'                      */
/* | | | | Endif                                                       */
/* | | | When the storage class is 'OLIBRARY' (i.e. Optical)           */
/* | | | | Set the management class to 'OPT6D'                         */
/* | | | Otherwise                                                     */
/* | | | | Set error code                                              */
/* | | | End                                                           */
/* | | When the collection is in the set that we are defining to       */
/* | | not allow overrides                                             */
/* | | | Select                                                        */
/* | | | When the storage class is 'DB2DASD' (i.e. DASD)               */
/* | | | | Set the management class to 'MAG30D'                        */
/* | | | When the storage class is 'OLIBRARY' (i.e. Optical)           */
/* | | | | Set the management class to 'OPT6D'                         */
/* | | | Otherwise                                                     */
/* | | | | Set error code                                              */
/* | | | End                                                           */
/* | | Otherwise                                                       */
/* | | | Set error code                                                */
/* | | End                                                             */
/* | Else (an object name was specified)                               */
/* | | Select                                                          */
/* | | When the collection is in the set that we are defining to       */
/* | | allow overrides                                                 */
/* | | | If the object name has exactly 5 levels and the 5th           */
/* | | | level indicates that the object may have the management       */
/* | | | class overridden and the storage class is 'DB2DASD'           */
/* | | | | If management class specified is not 'MAGONLY'              */
/* | | | | | Set the management class to 'MAG30D'                      */
/* | | | | Endif                                                       */
/* | | | Else                                                          */
/* | | | | Select                                                      */
/* | | | | When the storage class is 'DB2DASD' (i.e. DASD)             */
/* | | | | | Set the management class to 'MAG30D'                      */
/* | | | | When the storage class is 'OLIBRARY' (i.e. Optical)         */
/* | | | | | Set the management class to 'OPT6D'                       */
/* | | | | Otherwise                                                   */
/* | | | | | Set error code                                            */
/* | | | | End                                                         */
/* | | | Endif                                                         */
/* | | When the collection is in the set that we are defining to       */
/* | | not allow overrides                                             */
/* | | | Select                                                        */
/* | | | When the storage class is 'DB2DASD' (i.e. DASD)               */
/* | | | | Set the management class to 'MAG30D'                        */
/* | | | When the storage class is 'OLIBRARY' (i.e. Optical)           */
/* | | | | Set the management class to 'OPT6D'                         */
/* | | | Otherwise                                                     */
/* | | | | Set error code                                              */
/* | | | End                                                           */
/* | | Otherwise                                                       */
/* | | | Set error code                                                */
/* | | End                                                             */
/* | Endif (object name specified)                                     */
/* Endif (STORE environment)                                           */
/*                                                                     */
/*                                                                     */
/* If CHANGE environment                                               */
/* | Select                                                            */
/* | When storage class is 'DB2DASD'                                   */
```

```
/* | | If management class is not 'MAGONLY' or 'MAG30D'              */
/* | | | Set management class to 'MAG30D'                            */
/* | | Endif                                                         */
/* | When storage class is 'OLIBRARY'                                */
/* | | Set management class to 'OPT6LIB'                             */
/* | When storage class is 'TAPESC'                                  */
/* | | Set management class to 'OPTTAPE'                             */
/* | Otherwise                                                       */
/* | | Set error code                                                */
/* | End                                                             */
/* Endif (CHANGE environment)                                        */
/*                                                                   */
/*                                                                   */
/* If CLASS_TRANSITION environment                                   */
/* | Select                                                          */
/* | When storage class is 'OLIBRARY'                                */
/* | | Select                                                        */
/* | | When management class is 'OPT6D'                              */
/* | | | Set management class to 'OPT6LIB'                           */
/* | | When management class is 'MAG30D'                             */
/* | | | Set management class to 'MAG30LIB'                          */
/* | | Otherwise                                                     */
/* | | | Set error code                                              */
/* | | End                                                           */
/* | When storage class is 'TAPESC'                                  */
/* | | Select                                                        */
/* | | When management class is 'OPT6LIB'                            */
/* | | | Set management class to 'OPTTAPE'                           */
/* | | When management class is 'MAG30LIB'                           */
/* | | | Set management class to 'TAPSEVEN'                          */
/* | | Otherwise                                                     */
/* | | | Set error code                                              */
/* | | End                                                           */
/* | Otherwise                                                       */
/* | | Set error code                                                */
/* | End                                                             */
/* Endif                                                             */
/*********************************************************************/
/*  M A N A G E M E N T   C L A S S    D E F I N I T I O N S        */
/*                                                                   */
/*Relevant                                                           */
/*Fields    MAGONLY MAG30D  MAG30LIB TAPSEVEN OPT6D   OPT6LIB OPTTAPE*/
/*--------- ------- ------- -------- -------- ------- ------- -------*/
/*EXPIRE                                                             */
/*AFTER                                                              */
/*DAYS                                                               */
/*NON-USAGE NOLIMIT NOLIMIT NOLIMIT  NOLIMIT  NOLIMIT NOLIMIT NOLIMIT*/
/*                                                                   */
/*EXPIRE                                                             */
/*AFTER                                                              */
/*DATE/DAYS    30     2557    2557     2557     2557    2557    2557 */
/*                                                                   */
/*MAXIMUM                                                            */
/*RETENTION                                                          */
/*PERIOD       30   NOLIMIT NOLIMIT  NOLIMIT  NOLIMIT NOLIMIT NOLIMIT*/
/*                                                                   */
/*AUTO                                                               */
/*BACKUP      YES      NO      NO       NO       NO     YES      NO  */
/*                                                                   */
/*TIME                                                               */
/*SINCE                                                              */
/*CREATION                                                           */
/*YEARS        --      --      --       07       --      --      07  */
/*                                                                   */
/*TIME                                                               */
/*SINCE                                                              */
/*CREATION                                                           */
/*MONTHS       --      --      06       --       --      06      --  */
/*                                                                   */
/*TIME                                                               */
/*SINCE                                                              */
/*CREATION                                                           */
/*DAYS         --      30      --       --       00      --      --  */
/*                                                                   */
/*********************************************************************/

PROC MGMTCLAS                          /* Select an Management class */

   FILTLIST BLANK   INCLUDE ('        ','')

/*********************************************************************/
/* STORE                                                             */
```

```
/*********************************************************************/
 IF &ACSENVIR = 'STORE' THEN           /* Object is being stored     */
   IF &MEMN = &BLANK THEN              /* If the object name is not
                                          specified (i.e. indicating
                                          an invocation for the entire
                                          collection)                */
     SELECT
     WHEN (&DSN = 'CLLCT010' | &DSN = 'CLLCT011' | &DSN = 'CLLCT012'
         | &DSN = 'CLLCT013' | &DSN = 'CLLCT014' | &DSN = 'CLLCT015'
         | &DSN = 'CLLCT016' | &DSN = 'CLLCT017' | &DSN = 'CLLCT018'
         | &DSN = 'CLLCT019')          /* For the collections which
                                          allow overrides            */
       SELECT (&STORCLAS)

       WHEN ('DB2DASD')                /* If current storage class
                                          indicates that object is on
                                          DASD                       */
         IF &MGMTCLAS ^= 'MAGONLY' THEN /* If the specified management
                                          class value is not a valid
                                          override                   */
           SET &MGMTCLAS = 'MAG30D'    /* Set management class to
                                          indicate the DASD management
                                          specifications             */
       WHEN ('OLIBRARY')               /* If current storage class
                                          indicates that object is on
                                          optical                    */
         SET &MGMTCLAS = 'OPT6D'       /* Set management class to
                                          indicate the optical
                                          management specifications  */
       OTHERWISE                       /* Otherwise the storage class
                                          is invalid                 */
         EXIT CODE(20)                 /* Indicate that an error
                                          occurred                   */
       END
     WHEN (&DSN = 'CLLCT000' | &DSN = 'CLLCT001' | &DSN = 'CLLCT002'
         | &DSN = 'CLLCT003' | &DSN = 'CLLCT004' | &DSN = 'CLLCT005'
         | &DSN = 'CLLCT006' | &DSN = 'CLLCT007' | &DSN = 'CLLCT008'
         | &DSN = 'CLLCT009')          /* For the collections which
                                          do not allow overrides     */
       SELECT (&STORCLAS)

       WHEN ('DB2DASD')                /* If current storage class
                                          indicates that object is on
                                          DASD                       */
         SET &MGMTCLAS = 'MAG30D'      /* Set management class to
                                          indicate the DASD management
                                          specifications             */
       WHEN ('OLIBRARY')               /* If current storage class
                                          indicates that object is on
                                          optical                    */
         SET &MGMTCLAS = 'OPT6D'       /* Set management class to
                                          indicate the optical
                                          management specifications  */
       OTHERWISE                       /* Otherwise the storage class
                                          is invalid                 */
         EXIT CODE(21)                 /* Indicate that an error
                                          occurred                   */
       END
     OTHERWISE                         /* Otherwise the collection name
                                          is invalid                 */
       EXIT CODE(22)                   /* Indicate that an error
                                          occurred                   */
     END                               /* Select                     */
   ELSE                                /* If the object name is
                                          specified (i.e. indicating
                                          an invocation for the
                                          specific collection)       */
     SELECT
     WHEN (&DSN = 'CLLCT010' | &DSN = 'CLLCT011' | &DSN = 'CLLCT012'
         | &DSN = 'CLLCT013' | &DSN = 'CLLCT014' | &DSN = 'CLLCT015'
         | &DSN = 'CLLCT016' | &DSN = 'CLLCT017' | &DSN = 'CLLCT018'
         | &DSN = 'CLLCT019')          /* For the collections which
                                          allow overrides            */
       IF &MEMN = *.*.*.*.IAX AND
          &STORCLAS = 'DB2DASD' THEN   /* If the object name contains
                                          a value indicating that this
                                          object should be treated
                                          differently than the
                                          other objects in these
                                          collections (i.e. only
                                          specific objects within
                                          these collections allow
```

```
                                          overrides) and the storage
                                          class indicates that the
                                          object is on DASD          */
         IF &MGMTCLAS ^= 'MAGONLY' THEN /* If the specified management
                                          class value is not a valid
                                          override                   */
           SET &MGMTCLAS = 'MAG30D'    /* Set management class to
                                          indicate the DASD management
                                          specifications             */
         ELSE
           DO
           END
       ELSE                            /* Otherwise there is nothing
                                          special about this object
                                          name (i.e. so overrides
                                          will not be allowed)       */
         SELECT (&STORCLAS)

         WHEN ('DB2DASD')              /* If current storage class
                                          indicates that object is on
                                          DASD                       */
           SET &MGMTCLAS = 'MAG30D'    /* Set management class to
                                          indicate the DASD management
                                          specifications             */
         WHEN ('OLIBRARY')             /* If current storage class
                                          indicates that object is on
                                          optical                    */
           SET &MGMTCLAS = 'OPT6D'     /* Set management class to
                                          indicate the optical
                                          management specifications  */
         OTHERWISE                     /* Otherwise the storage class
                                          is invalid                 */
           EXIT CODE(23)               /* Indicate that an error
                                          occurred                   */
         END
     WHEN (&DSN = 'CLLCT000' | &DSN = 'CLLCT001' | &DSN = 'CLLCT002'
         | &DSN = 'CLLCT003' | &DSN = 'CLLCT004' | &DSN = 'CLLCT005'
         | &DSN = 'CLLCT006' | &DSN = 'CLLCT007' | &DSN = 'CLLCT008'
         | &DSN = 'CLLCT009')          /* For the collections which
                                          do not allow overrides     */
       SELECT (&STORCLAS)

       WHEN ('DB2DASD')                /* If current storage class
                                          indicates that object is on
                                          DASD                       */
         SET &MGMTCLAS = 'MAG30D'      /* Set management class to
                                          indicate the DASD management
                                          specifications             */
       WHEN ('OLIBRARY')               /* If current storage class
                                          indicates that object is on
                                          optical                    */
         SET &MGMTCLAS = 'OPT6D'       /* Set management class to
                                          indicate the optical
                                          management specifications  */
       OTHERWISE                       /* Otherwise the storage class
                                          is invalid                 */
         EXIT CODE(24)                 /* Indicate that an error
                                          occurred                   */
       END
     OTHERWISE                         /* Otherwise the collection name
                                          is invalid                 */
       EXIT CODE(25)                   /* Indicate that an error
                                          occurred                   */
     END                               /* Select                     */

/**********************************************************************/
/* CHANGE                                                             */
/**********************************************************************/
 IF &ACSENVIR = 'CHANGE' THEN          /* Object is being changed    */
   SELECT (&STORCLAS)

   WHEN ('DB2DASD')                    /* If current storage class
                                          indicates that object is on
                                          DASD                       */
     IF &MGMTCLAS ^= 'MAGONLY' AND
        &MGMTCLAS ^= 'MAG30D' THEN     /* If the specified management
                                          class value is not a valid
                                          override                   */
       SET &MGMTCLAS = 'MAG30D'        /* Set management class to
                                          indicate the DASD management
                                          specifications             */
   WHEN ('OLIBRARY')                   /* If current storage class
```

```
                                          indicates that object is on
                                          optical                    */
     SET &MGMTCLAS = 'OPT6LIB'         /* Set management class to
                                          indicate the optical
                                          management specifications  */
   WHEN ('TAPESC')                     /* If current storage class
                                          indicates that object is on
                                          tape                       */
     SET &MGMTCLAS = 'OPTTAPE'         /* Set management class to
                                          indicate the tape
                                          management specifications  */
   OTHERWISE                           /* Otherwise the storage class
                                          is invalid                 */
     EXIT CODE(26)                     /* Indicate that an error
                                          occurred                   */
   END
/*********************************************************************/
/* CLASS TRANSITION                                                  */
/*********************************************************************/
 IF &ACSENVIR = 'CTRANS' THEN          /* Object is being processed
                                          as a result of a class
                                          transition                 */
   SELECT (&STORCLAS)

   WHEN ('OLIBRARY')                   /* If current storage class
                                          indicates that object is on
                                          optical                    */
     SELECT (&MGMTCLAS)

     WHEN ('OPT6D')                    /* If current management class
                                          indicates optical management
                                          specifications             */
       SET &MGMTCLAS = 'OPT6LIB'       /* Set management class to
                                          indicate the appropriate
                                          optical management
                                          specifications             */
     WHEN ('MAG30D')                   /* If current management class
                                          indicates DASD management
                                          specifications             */
       SET &MGMTCLAS = 'MAG30LIB'      /* Set management class to
                                          indicate the appropriate
                                          optical management
                                          specifications             */
     OTHERWISE                         /* Otherwise the management
                                          class is invalid           */
       EXIT CODE(27)                   /* Indicate that an error
                                          occurred                   */
     END
   WHEN ('TAPESC')                     /* If current storage class
                                          indicates that object is on
                                          tape                       */
     SELECT (&MGMTCLAS)

     WHEN ('OPT6LIB')                  /* If current management class
                                          indicates optical management
                                          specifications             */
       SET &MGMTCLAS = 'OPTTAPE'       /* Set management class to
                                          indicate the appropriate
                                          tape management
                                          specifications             */
     WHEN ('MAG30LIB')                 /* If current management class
                                          indicates optical management
                                          specifications             */
       SET &MGMTCLAS = 'TAPSEVEN'      /* Set management class to
                                          indicate the appropriate
                                          tape management
                                          specifications             */
     OTHERWISE                         /* Otherwise the management
                                          class is invalid           */
       EXIT CODE(28)                   /* Indicate that an error
                                          occurred                   */
     END
   OTHERWISE                           /* Otherwise the storage class
                                          is invalid                 */
     EXIT CODE(29)                     /* Indicate that an error
                                          occurred                   */
   END
END
```

CBRHSG

SAMPLIB member CBRHSG, as shown here, provides a storage group ACS routines for OAM objects.

```
/*******************************************************************/ 00050000
/*                                                                 */ 00100000
/* $SEG(CBRHSG) COMP(OSMC) PROD(OAM):                              */ 00150000
/*                                                                 */ 00200000
/* OAM Sample Storage Group ACS Routine                            */ 00250000
/*                                                                 */ 00250001
/*  Licensed Materials - Property of IBM                           */ 00250002
/*  5650-ZOS                                                       */ 00250003
/*  COPYRIGHT IBM CORP. 1989, 1992                                 */ 00250004
/*                                                                 */ 00300000
/* FUNCTION: SUPPLY A STORAGE GROUP FOR OAM OBJECTS                */ 00350000
/*                                                                 */ 00400000
/* OPERATION: Select a storage group based upon the collection name  */ 00450000
/*            specified                                            */ 00500000
/*                                                                 */ 00550000
/* NOTES:     In this implementation, the collection name is used as */ 00600000
/*            the basis for determining whether explicit values will */ 00650000
/*            be considered. This approach and all of the processing */ 00700000
/*            indicated below is one of many differing possibilities */ 00750000
/*            and is only for purposes of illustration to demonstrate*/ 00800000
/*            the types of processing that can be accomplished in an */ 00850000
/*            ACS routine. Actual implementations will vary.        */ 00900000
/*                                                                 */ 00950000
/* ASSUMPTIONS:                                                    */ 01000000
/*   Collection name format                                        */ 01050000
/*     CLLCT0mn                                                    */ 01100000
/*       where m = 0 or 1 (0 indicates that explicit values will be  */ 01150000
/*                        ignored, 1 indicates that explicit values */ 01200000
/*                        are considered in some cases and may      */ 01250000
/*                        result in an override)                   */ 01300000
/*       where n = 0 - 9                                           */ 01350000
/*   Valid storage groups:                                         */ 01400000
/*     SGROUP00 - SGROUP09                                         */ 01450000
/*                                                                 */ 01500000
/*   CHANGE ACTIVITY:                                              */ 01550000
/*     $L0=JDP3227 320 890601 TUCKSG: Initial Release              */ 01600000
/*     $L1=JDP3331 331 910614 TUCKSG: Added sample definitions     */ 01650000
/*                                                                 */ 01700000
/*******************************************************************/ 01750000
/*      S T O R A G E   G R O U P   D E F I N I T I O N S          */ 01800000
/*                                                                 */ 01850000
/*Relevant                                                         */ 01900000
/*Fields    SGROUP00 SGROUP01 SGROUP02 SGROUP03 SGROUP04 SGROUP05    */ 01950000
/*--------- -------- -------- -------- -------- -------- --------    */ 02000000
/*SG                                                               */ 02050000
/*TYPE       OBJECT   OBJECT   OBJECT   OBJECT   OBJECT   OBJECT     */ 02100000
/*                                                                 */ 02150000
/*QUALIFIER  GROUP00  GROUP01  GROUP02  GROUP03  GROUP04  GROUP05    */ 02200000
/*                                                                 */ 02250000
/*CYCLE                                                            */ 02300000
/*START        00       01       02       03       04       05       */ 02350000
/*                                                                 */ 02400000
/*CYCLE                                                            */ 02450000
/*END          03       04       05       06       07       08       */ 02500000
/*                                                                 */ 02550000
/*LIBRARY                                                          */ 02600000
/*NAMES       LIB1     LIB1     LIB1     LIB1     LIB1     LIB1      */ 02650000
/*                              LIB2     LIB3     LIB2     LIB2      */ 02700000
/*                                                LIB3     LIB3      */ 02750000
/*                                                                 */ 02800000
/*VOLUME                                                           */ 02850000
/*FULL         32       32       32       32       32       32       */ 02900000
/*                                                                 */ 02950000
/*DRIVE                                                            */ 03000000
/*START       099      099      099      099      099      099       */ 03050000
/*                                                                 */ 03100000
/*WRITE                                                            */ 03150000
/*ERROR       YES      YES      YES      YES      YES      YES       */ 03200000
/*                                                                 */ 03250000
/*                                                                 */ 03300000
/*                                                                 */ 03350000
/*Relevant                                                         */ 03400000
/*Fields    SGROUP06 SGROUP07 SGROUP08 SGROUP09                      */ 03450000
/*--------- -------- -------- -------- --------                      */ 03500000
/*SG                                                               */ 03550000
/*TYPE       OBJECT   OBJECT   OBJECT   OBJECT                       */ 03600000
```

```
/*                                                                   */
/*                                                                   */
/*QUALIFIER  GROUP06  GROUP07  GROUP08  GROUP09                      */
/*                                                                   */
/*CYCLE                                                              */
/*START        06       07       08       09                         */
/*                                                                   */
/*CYCLE                                                              */
/*END          09       10       11       12                         */
/*                                                                   */
/*LIBRARY                                                            */
/*NAMES       LIB1     LIB1     LIB1     LIB1                        */
/*            LIB2     LIB2     LIB2     LIB2                        */
/*            LIB3     LIB3     LIB3     LIB3                        */
/*                                                                   */
/*VOLUME                                                             */
/*FULL         32       32       32       32                         */
/*                                                                   */
/*DRIVE                                                              */
/*START       099      099      099      099                         */
/*                                                                   */
/*WRITE                                                              */
/*ERROR       YES      YES      YES      YES                         */
/*                                                                   */
/*********************************************************************/

 PROC STORGRP

 FILTLIST DSN_NAMES INCLUDE(CLLCT0%%)

 IF &DSN = &DSN_NAMES THEN             /* If the first 6 characters of
                                          the collection name are
                                          valid                      */
   IF &DSN ^= ' ' THEN                 /* If the collection name is
                                          not blank (this test will
                                          always pass, but allows for
                                          the apparent assignment of
                                          the 'POOL' storage group
                                          which is a requirement of
                                          a storage group ACS
                                          routine)                   */
/*********************************************************************/
/* Map the collection name to a storage group, where the last digit  */
/* in the collection name corresponds to the last digit of the       */
/* storage group.                                                    */
/*********************************************************************/
     SELECT
     WHEN (&DSN = 'CLLCT000' | &DSN = 'CLLCT010')
       SET &STORGRP = 'SGROUP00'
     WHEN (&DSN = 'CLLCT001' | &DSN = 'CLLCT011')
       SET &STORGRP = 'SGROUP01'
     WHEN (&DSN = 'CLLCT002' | &DSN = 'CLLCT012')
       SET &STORGRP = 'SGROUP02'
     WHEN (&DSN = 'CLLCT003' | &DSN = 'CLLCT013')
       SET &STORGRP = 'SGROUP03'
     WHEN (&DSN = 'CLLCT004' | &DSN = 'CLLCT014')
       SET &STORGRP = 'SGROUP04'
     WHEN (&DSN = 'CLLCT005' | &DSN = 'CLLCT015')
       SET &STORGRP = 'SGROUP05'
     WHEN (&DSN = 'CLLCT006' | &DSN = 'CLLCT016')
       SET &STORGRP = 'SGROUP06'
     WHEN (&DSN = 'CLLCT007' | &DSN = 'CLLCT017')
       SET &STORGRP = 'SGROUP07'
     WHEN (&DSN = 'CLLCT008' | &DSN = 'CLLCT018')
       SET &STORGRP = 'SGROUP08'
     WHEN (&DSN = 'CLLCT009' | &DSN = 'CLLCT019')
       SET &STORGRP = 'SGROUP09'
     OTHERWISE
       EXIT CODE(30)
     END
   ELSE
     DO
       SET &STORGRP = 'POOL'
       EXIT CODE(31)
     END
 ELSE
   EXIT CODE(32)
 END
```

Appendix C. Example file system configuration for OAM usage

This topic provides an example of the file system configuration for OAM usage. This description supplements the information provided in "3 Configure the z/OS Unix file system" on page 103.

There are many ways to accomplish the individual tasks identified below. This is just one representative example.

From the planning phase, assume that the following values are known:

Attribute	**Values used in this example**
Security Product OAM Group Name	OAMGRP
Security Product OAM User Name	OAM
Object Storage Group Name	SG3
File System Directory Name (L2DIR)	/myoam/sg3
File System Directory Type (L2TYPE)	ZFS

A zFS aggregate with the name "OAM.SG3.ZFS" has been created for the OAM object storage group named SG3. Note that the value used for the File System Directory Name is case sensitive, so the file system directory name used in the z/OS UNIX file system hierarchy must exactly match the value later specified in the OAM configuration on the SETDISK statement of the CBROAM*xx* member of PARMLIB.

Assume that an existing directory named /myoam in the z/OS UNIX file system hierarchy will be used to contain the mount points for all of the file systems to be used for OAM object storage groups.

A z/OS UNIX superuser in a z/OS UNIX shell:

1. Creates a new directory named sg3 for the OAM storage group named SG3. The current directory is /myoam.

   ```
   mkdir sg3
   ```

2. Mounts the physical file system "OAM.SG3.ZFS" at the mount point /myoam/sg3.

   ```
   mount -v -t ZFS -f "OAM.SG3.ZFS" /myoam/sg3
   ```

3. Changes the owner of the mounted file system to the security product "OAM" user name

   ```
   chown OAM /myoam/sg3
   ```

4. Changes the group of the mounted file system to the security product "OAMGRP" group name

   ```
   chgrp OAMGRP /myoam/sg3
   ```

5. Creates the permissions of the mounted file system to limit access only to the "OAM" user name

   ```
   chmod 700 /myoam/sg3
   ```

6. Creates the required sentinel file named 4oamonly within the mounted file system

   ```
   echo 0 > /myoam/sg3/4oamonly
   ```

7. Changes the owner of the sentinel file to the security product "OAM" user name

   ```
   chown OAM /myoam/sg3/4oamonly
   ```


8. Changes the group of the sentinel file to the security product "OAMGRP" group name

   ```
   chgrp OAMGRP /myoam/sg3/4oamonly
   ```

9. Creates the permissions of the sentinel file to limit access only to the "OAM" user name

   ```
   chmod 600 /myoam/sg3/4oamonly
   ```

   Then the OAM administrator:

10. Adds a SETDISK statement to the CBROAM*xx* member of PARMLIB to associate the location of the mounted file system /myoam/sg3 with the OAM object storage group SG3. Because the L2DIR directory value is case sensitive, when editing in TSO/ISPF remember to set "CAPS OFF" so that any lower case values are preserved when changes to the CBROAM*xx* member of PARMLIB are saved.

    ```
    SETDISK STORAGEGROUP(SG3 L2DIR(/myoam/sg3) L2TYPE(ZFS))
    ```

To use the file system sublevel, the SMS administrator must add or update one or more SMS storage class definitions to specify the file system sublevel. See “Understanding the storage class construct” on page 24 for more information. Also review and update ACS routines as needed to use the storage class.

Appendix D. Understanding OAM databases

OAM uses DB2 databases to store information about objects and to store the objects themselves. This appendix documents diagnosis and tuning information to help diagnose OAM problems. It contains information on DB2 databases and should be used only for diagnosis.

OAM uses the following DB2 databases:

- **Object Storage Databases**—Contain an object directory table and optional object storage tables.
- **Object Administration Database**—Contains the relationship between identifiers and the names of storage classes, management classes, and collections.
- **OAM Configuration Database**—Contains information about the optical hardware configuration and the optical disk volumes.

Object storage databases

The object storage databases are a set of DB2 databases containing two types of data:

- Descriptive information about objects
- Actual data for the objects stored at the DB2 sublevel of the OAM storage hierarchy

Each Object storage group has an object storage database.

Each object storage database contains tables for an object directory and object storage. The object directory table contains descriptive information about each object. Object storage tables contain the objects. A separate table space exists for each table. Each database has three required tables:

- An object directory table (contains descriptive information about objects)
- A 4 KB table (contains the data for small objects)
- A 32 KB table (contains the data for larger objects up to 256M)

LOB support requires the LOB storage structure in addition to the three tables mentioned above. The new structure stores object data that is greater than 32 KB or 32640 bytes for storage groups that are LOB enabled. The LOB storage structure consists of two tables:

- A LOB base table (resembles the 32 KB table with the addition of a ROWID column and changing the OTOBJ column datatype from 'long varchar' to BLOB.
- A LOB auxiliary table (contains the actual BLOB object represented by the OTOBJ column in the LOB base table).

Note: Reference to the LOB storage structure refers to both the LOB base table and the LOB auxiliary table.

OAM uses multiple object storage databases, each containing the three tables mentioned above. Table 50 on page 483 shows the tables and table space names.

Table 50. Object storage database naming conventions

Database Name — hlq		
Table Name	**Table Space Name**	**Contents**
hlq.OSM_OBJ_DIR	OSMDTS	Object directory
hlq.OSM_04 KB_OBJ_TBL	OSMOTS04	Small objects

Table 50. Object storage database naming conventions (continued)		
Database Name — hlq		
hlq.OSM_32 KB_OBJ_TBL	OSMOTS32	Large objects (LOB support enabled)
hlq.OSM_LOB_BASE_TBL	OSMLBTS	LOB base table
hlq.OSM_LOB_AUX_TBL	OTLOBAX1	Large objects (LOB support enabled)

Sample programs that define these databases and tables are shipped with OAM. You must update these programs to meet the requirements of your installation before they are run.

The tables defined by the storage administrator will not be used by OAM unless they are related to an Object storage group through the services of ISMF. This relationship results in the definition of the DASD level of the OAM hierarchy for the specified Object storage group, and the object directory for all levels of the object storage hierarchy in that storage group. (See Figure 68 on page 484.)

The object directory table from each three-table set contains an entry for each object stored in an Object storage group. The object itself might exist in one of the object storage tables on DASD or it might exist on optical disk or tape.

Table indexes are necessary for performance. Within an Object storage group, the directory table has three indexes, and each object storage table has a single index. All indexes are unique. Indexes are searched in ascending sequence (ASC).

Note: You must calculate the space required for the indexes separately because it is not included in the directory and object field sizes outlined in each table.

For the sample jobs CBRIALC0 and CBRISQL0, see Appendix B, “Sample library members,” on page 421.

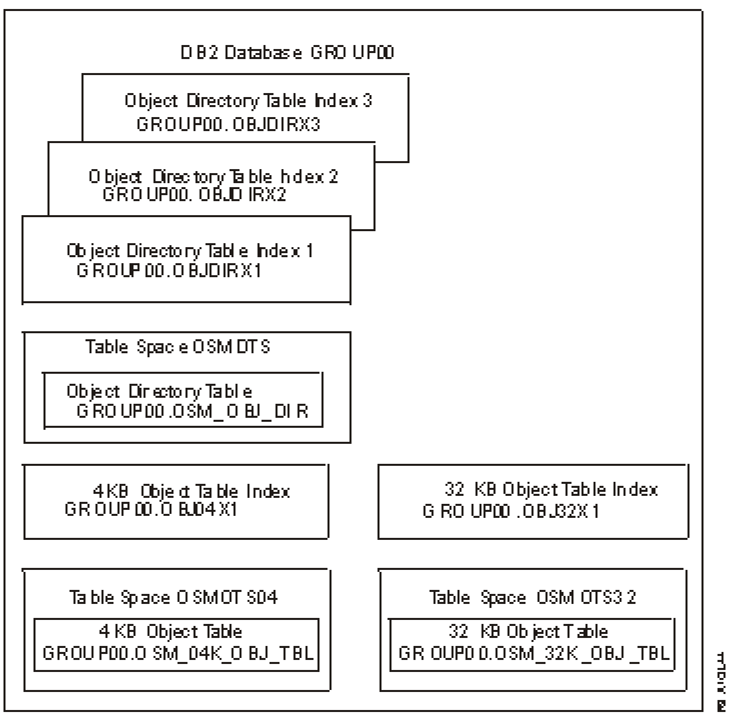


Figure 68. Object Storage Group Database Structure

Object directory tables

The object directory tables contain information about objects. OAM keeps track of all objects in the storage hierarchy by recording the collection name identifier, the object name, and other pertinent information in the object directory tables. The object directory tables contain entries for locating and describing objects in the storage hierarchy. OSR creates a directory entry for each object when the object is stored. OSMC uses the directory table to determine which objects need to be processed during each management cycle for an Object storage group.

Table 51 on page 485 shows the fields for an entry in an object directory table.

The object directory table has three indexes:

- **Index 1**—Object Creation Time Stamp
- **Index 2**—Collection Name Identifier, Pending Action Date, and Object Creation Time Stamp
- **Index 3**—Object Name and Collection Name Identifier

Table 51. Object directory table

Column description	Column name	DB2 field type and data size	Indexes where used
Data Format Version	ODVER	CHAR (1)	
Object Size	ODSIZE	INTEGER	
Object Creation Time Stamp	ODCREATS	TIMESTAMP	Index 1, Index 2
Expiration Date	ODEXPDT	DATE	
Last Referenced Date	ODLREFDT	DATE	
Pending Action Date	ODPENDDT	DATE	Index 2
Management Class Assignment Date	ODMCASDT	DATE	
Storage Class Identifier	ODSCNUM	SMALLINT	
Management Class Identifier	ODMCNUM	SMALLINT	
Large Object Support Flag	ODLOBFL	CHAR (1)	
Object Location Flag	ODLOCFL	CHAR (1)	
Object Active Volume Serial Number	ODLSLOC	CHAR (6)	
Primary Copy Location Token	ODSECLOC	INTEGER	
Object Backup Volume Serial Number	ODBKLOC	CHAR (6)	
Backup Copy Location Token	ODBKSEC	INTEGER	
Second Backup Copy Volume Serial Number	ODBK2LOC	CHAR (6)	
Second Backup Copy Location Token	ODBK2SEC	INTEGER	
Collection Name Identifier	ODCLID	INTEGER	Index 2, Index 3
Object Name	ODNAME	VARCHAR (44)	Index 3
Object Status Flags	ODSTATF	SMALLINT	
Object Retention Date	ODRETDT	DATE	
Instance ID	ODINSTID	INTEGER	

Table 51. Object directory table (continued)			
Column description	**Column name**	**DB2 field type and data size**	**Indexes where used**
Note: 1. All columns are created with the NOT NULL attribute. 2. All indexes are unique by concatenation of identified columns. 3. All indexes are ordered in ascending value sequence. 4. Index 1 is a cluster index. 5. Maximum size of an object directory entry in bytes: 175.			

Table 52 on page 486 explains the column contents of an object directory table entry.

Table 52. Object directory table field contents	
Column description	**Column contents**
Data Format Version	4
Object Size	Object size in bytes
Object Creation Time Stamp	Compressed form (DB2 format) (*yyyy.mm.dd-hh*:*mm*:*ss.mmmmmm*)
Expiration Date	Compressed form (DB2 format) **0001-01-01** Use MC expiration ***yyyy-mm-dd*** Explicit expiration date **9999-12-31** Never expire **0002–02–02** Awaiting a future event (event-based-retention)
Last Referenced Date	Compressed form (DB2 format) Set to 0001-01-01 upon creation
Pending Action Date	Compressed form (DB2 format)
Management Class Assignment Date	Compressed form (DB2 format) Set to creation date on creation; otherwise, last date MC changed
Storage Class Identifier	Number identifying this storage class (associated with storage class name by the Storage Class Identifier Table)
Management Class Identifier	Number identifying this management class (associated with management class name by the Management Class Identifier Table)

Table 52. Object directory table field contents (continued)	
Column description	**Column contents**
Large Object Support Flag	Indicates whether this object resides in a LOB tablespace. Valid values are: • L—object currently resides in a LOB storage structure. • Blank—object does not currently reside in a LOB storage structure.
Object Location Flag	Blank—Optical copy; D—DASD copy; E—File system DSL2: R—Recalled state; T—Tape SUBLEVEL 1 copy; U—Tape SUBLEVEL 2 copy; 2—Recalled to DSL2 :
Object Active Volume Serial Number	Standard MVS volume serial number (or blanks)
Primary Copy Location Token	If optical volume: Token for relative sector location (or zeros) of VTOC entry. If tape volume: Tape blockid
Object Backup Volume Serial Number	Standard MVS volume serial number (or blanks)
Backup Copy Location Token	If optical volume: Token for relative sector location (or zeros) of VTOC entry. If tape volume: Tape blockid.
Second Backup Copy Volume Serial Number	Volume serial number for the optical or tape volume that contains the second backup copy of the object in the corresponding row in the table
Second Backup Copy Location Token	The optical volume sector location or the tape volume block ID on the volume in the ODBK2LOC field where the second backup copy of the object in the corresponding row in the table resides
Collection Name Identifier	Number identifying the collection name (associated with collection name by the Collection Name Identifier Table)
Object Name	Standard MVS data set name
Object Status Flags	ODSTATF contains the object status flags. See Table 53 on page 488 valid values for ODSTATF and how they map to status flags.
Object Retention Date	ODRETDT contains the latest expiration date derived for a retention-protected object. The retention-protected object cannot be deleted prior to the date specified in this column. For objects that are not retention-protected, ODRETDT is not used and contains 0001–01–01.
Instance ID	Unique object instance ID.

The following table shows valid values for ODSTATF and how they map to status flags.

Table 53. Valid values for ODSTATF			
ODSTATF Value	**ODSTATF_EBR**	**ODSTATF_DELHOLD**	**ODSTATF_RETPROT**
0	OFF	OFF	OFF
1	OFF	OFF	ON
2	OFF	ON	OFF
3	OFF	ON	ON
4	ON	OFF	OFF
5	ON	OFF	ON
6	ON	ON	OFF
7	ON	ON	ON

Object storage tables

The object storage tables provide DASD storage for objects. Objects are stored in the 4 KB or 32 KB table or LOB storage structure, depending on size and whether or not LOB support is enabled. If an object is 3980 bytes or smaller, it is stored in the 4 KB table. If the object is larger than 3980 bytes but smaller or equal to 32640 bytes, it is stored in the 32 KB table. If the object is larger than 32640 bytes, but less than or equal to 256M and LOB support is disabled, then it is stored in multiple rows in the 32 KB table. If the object is larger than 32640 bytes and LOB support is enabled, then it is stored in a LOB storage structure. Note that objects greater than 256M can only be stored in a LOB storage structure. Refer to “5 Changing system libraries” on page 104 for more information on LOB support.

Objects stored in the 32 KB table might be broken into segments and stored as rows. Each row in the 32 KB table can contain up to 32 640 bytes of object data.

Table 54 on page 488 shows the contents of an entry in a 4 KB or 32 KB object storage table.

Each object storage table has one index. The 4 KB table index is the concatenation of the collection name ID and object name in ascending-order sequence. The 32 KB table index is the concatenation of the collection name ID, object name and segment number in ascending-order sequence. When objects are retrieved, they are ordered by object segment number.

Table 54. Object storage table			
Column description	**Column name**	**DB2 field type and data size**	**Indexes where used**
Data Format Version	OTVER	CHAR (1)	
Segment Number	OTSEG	SMALLINT	Index 1
Collection Name Identifier	OTCLID	INTEGER	Index 1
Object Name	OTNAME	VARCHAR (44)	Index 1
Object Data Segment	OTOBJ	LONG VARCHAR	

Table 54. Object storage table (continued)			
Column description	**Column name**	**DB2 field type and data size**	**Indexes where used**

Note:

1. All columns are created with the NOT NULL attribute.
2. The object table columns are the same for the 4 KB and 32 KB tables.
3. Segment number is *not* used in the 4 KB table.
4. The index on each table is a unique cluster index.
5. Maximum sizes of object table entries:

    4 KB table is 3 980
    32 KB table is 32 640

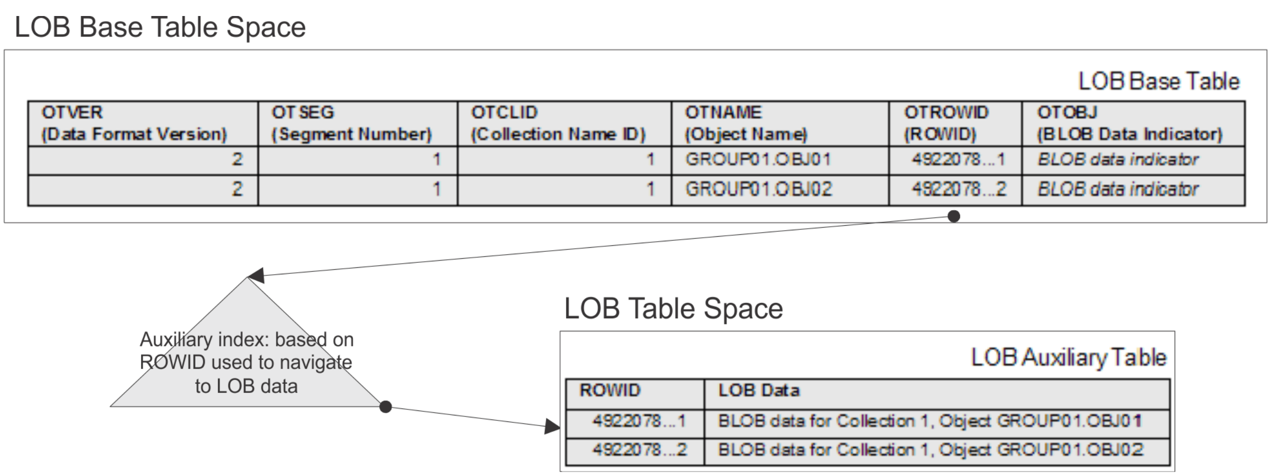


Figure 69. LOB Storage Structure

Table 55 on page 489 shows the contents of an entry in a LOB storage table.

Table 55. LOB storage table			
Column description	**Column name**	**DB2 field type and data size**	**Indexes where used**
Data Format Version	OTVER	CHAR (1)	
Segment Number	OTSEG	SMALLINT	
Collection Name Identifier	OTCLID	INTEGER	Index 1
Object Name	OTNAME	VARCHAR (44)	Index 1
Row ID	OTROWID	ROWID	
Object Data Segment	OTOBJ	BLOB(2G)	

Table 55. LOB storage table (continued)			
Column description	**Column name**	**DB2 field type and data size**	**Indexes where used**
Note: 1. All columns are created with the NOT NULL attribute. 2. The index on each table is a unique cluster index. 3. Maximum sizes of object table entries are 2000M (2,097,152,000 bytes). 4. The LOB auxiliary table is managed exclusively by DB2 and is transparent to OAM. 5. Columns OTVER and OTSEG are reserved for future use.			

Object administration database

Each object stored in the OAM storage hierarchy is part of a collection and is assigned a storage class and management class. These assignments are recorded in the object's directory entry. To conserve DASD space, OAM stores an identifier that represents those names instead of recording the names in each directory entry. OAM requires two tables to relate the identifiers to the storage class and management class names, and a third table to describe collections (see Figure 70 on page 490). Table 56 on page 491 through Table 64 on page 504 are used for diagnostic reference.

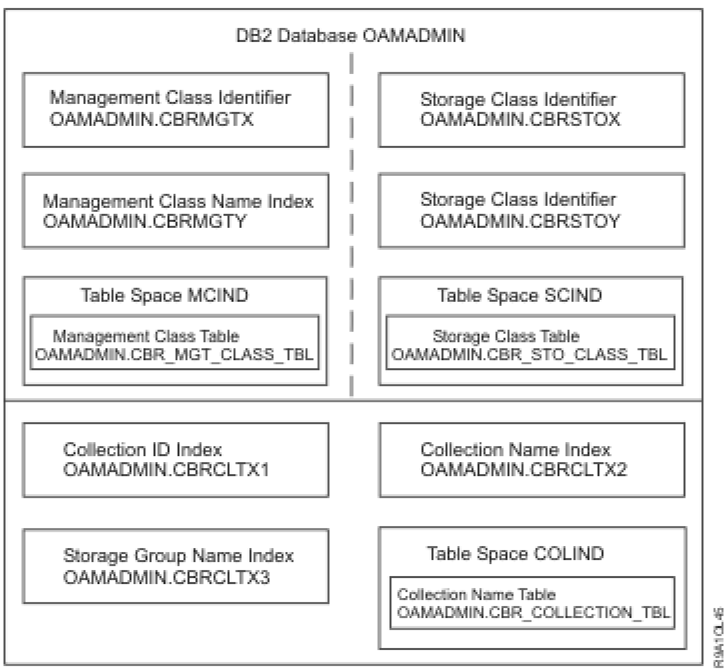


Figure 70. Object Administration Database Structure

Storage class identifier table

Each object stored in the OAM storage hierarchy is assigned a storage class. This assignment is recorded in the object's directory entry as a storage class identifier. The storage class identifier table maps the identifier to the storage class name (see Table 56 on page 491).

The storage class identifier table has one index: a unique, ascending cluster index on the storage class identifier.

Table 56. Storage class identifier table				
Column description	**Column name**	**DB2 field type and data size**	**Max byte**	**Index structure**
Storage Class Identifier	ODSCNUM	SMALLINT	(2)	Unique ASC
Storage Class Name	ODSCNAME	VARCHAR (30)	(32)	Unique ASC
Index Structure: ASC (ascending sequence)				

Management class identifier table

Each object stored in the OAM storage hierarchy is assigned a management class. This assignment is recorded in the object's directory entry as a management class identifier. The management class identifier table maps the identifier to the management class name (see Table 57 on page 491).

The management class identifier table has one index: a unique, ascending cluster index on the management class identifier.

Table 57. Management class identifier table				
Column description	**Column name**	**DB2 field type and data size**	**Max byte**	**Index definition**
Management Class Identifier	ODMCNUM	SMALLINT	(2)	Unique ASC
Management Class Name	ODMCNAME	VARCHAR (30)	(32)	Unique ASC

Collection name identifier table

Each object stored in the OAM storage hierarchy is a member of a collection. The name of the collection to which an object belongs is recorded in the object's directory entry as a collection name identifier. The collection name identifier table maps the identifier to the collection name (see Table 58 on page 491). In addition, the collection name identifier table contains information about the storage class and management class for the collection and the name of the storage group that contains all members of the collection (see Table 59 on page 492).

Table 58. Collection name identifier table				
Column description	**Column name**	**DB2 field type and data size**	**Max byte**	**Index structure**
Storage Class Name	ODCLSCNM	VARCHAR (30)	(32)	
Management Class Name	ODCLMCNM	VARCHAR (30)	(32)	
Directory Token (*)	ODCLSGNM	VARCHAR (30)	(32)	ASC
Collection Name Identifier (*)	ODCLID	INTEGER (4)	(4)	Unique ASC Cluster
Collection Name (*)	ODCLNAME	VARCHAR (44)	(46)	Unique ASC
Total Bytes per Table Entry	138	146		

Table 59. Collection name identifier table contents	
Column description	**Contents**
Storage Class Name	Default initial storage class for all objects in this collection (can be overridden by explicit storage class on OSREQ STORE)
Management Class Name	Default initial management class for all objects in this collection (can be overridden by explicit management class on OSREQ STORE)
Directory Token	Storage group name
Collection Name Identifier	Numeric index identifying the collection that includes this object (used to improve DASD space usage in object tables)
Collection Name	Standard MVS data set name

OAM configuration database

The OAM configuration database (CBROAM) contains configuration information related to the target destinations for objects including tape volumes, optical libraries, drives, slots, and volumes. CBROAM also identifies objects to be ultimately deleted by OAM from optical volumes and the file system. It is a DB2 database and consists of the following tables:

- "Library table (OLIBRARY)" on page 494
- "Drive table (DRIVE)" on page 496
- "Slot table (SLOT)" on page 498)
- "Volume table (VOLUME)" on page 499
- "Deleted objects table (DELOBJT)" on page 504
- "Tape volume table (TAPEVOL)" on page 504
- "File system delete table (FSDELETE)" on page 512)

Figure 71 on page 493 shows the organization of the OAM configuration database. There are seven table spaces, each containing a different table and its associated indexes.

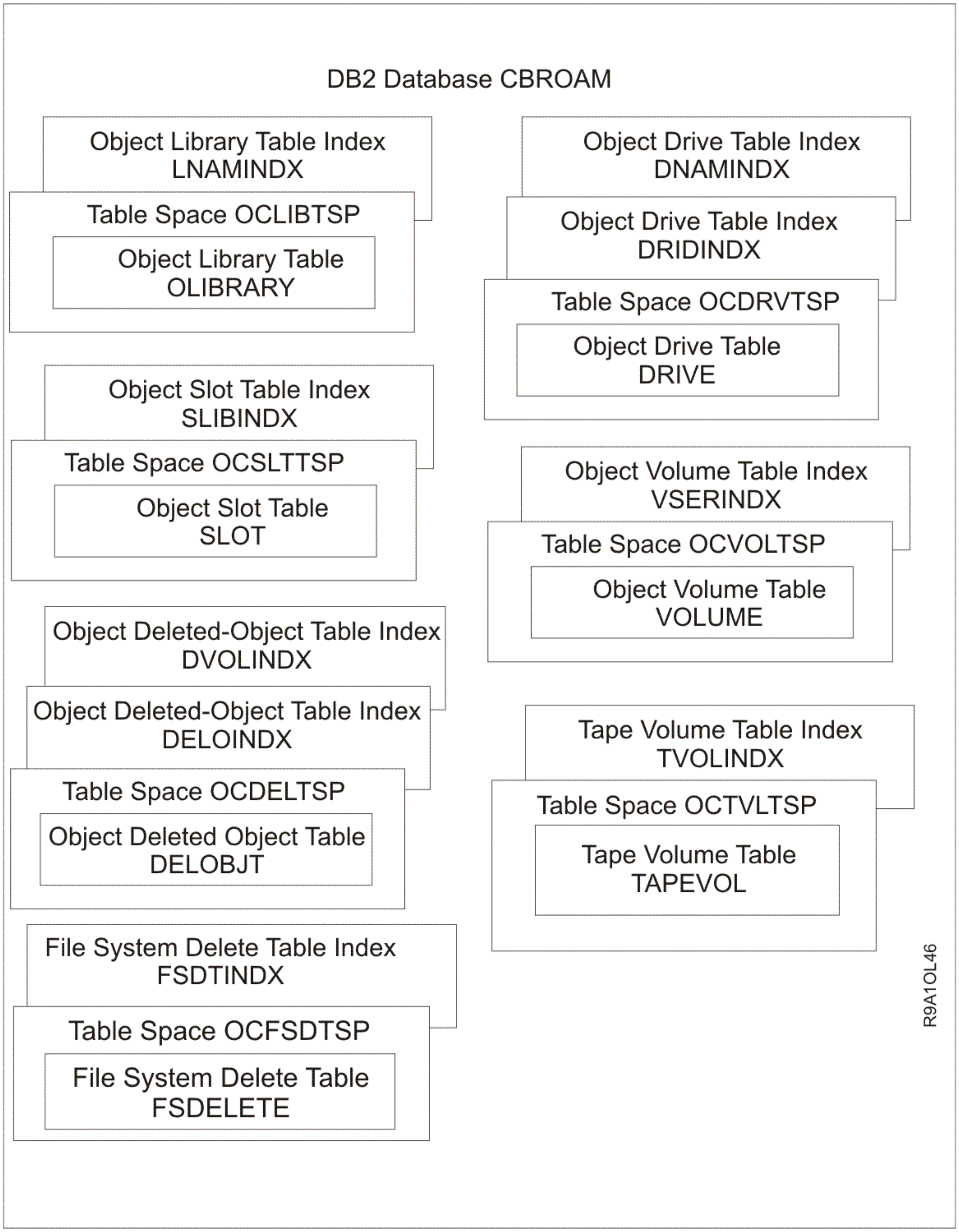


Figure 71. OAM Configuration Database

Table 60 on page 494, Table 61 on page 497, Table 62 on page 498, Table 63 on page 499, Table 64 on page 504, and Table 65 on page 505 describe the tables in the OAM configuration database. A description of special characteristics follows each table.

Each table is in its own table space and each table has at least one index.

Note: If DB2 searches a table without using an index, it must search the entire table space. Each table is within its own table space to decrease search time when an index is not used.

The deleted-objects table is used only with OAM rewritable media.

The following explains the information in the DB2 Attribute column of each figure:

GAI
This column is declared with the GENERATED ALWAYS and AS IDENTITY attributes.

NN
This column is declared with the NOT NULL attribute. A value must be supplied.

NND
This column is declared with the NOT NULL WITH DEFAULT attribute. If some other value is not given, the following DB2 data types and default values are supplied:

CHARACTER
blanks

SMALLINT
zero

INTEGER
zero

DATE
current date

The value column of each figure shows the columns that require specific values. When the value column contains any information other than blank, that column must contain a specific value. The possible values are listed in the value column. The explanations about each table define the specific values.

Library table (OLIBRARY)

The library table contains one row for each optical library. The DB2 name of the library table is OLIBRARY. The table is defined in table space OCLIBTSP. Table 60 on page 494 describes the attributes of the columns in the OLIBRARY table. A row is inserted into the library table the first time that an optical library with a given name is defined using the ISMF Library Management application.

Table 60. Library table column description

Column description	Name	Index	DB2 type	DB2 attribute	Value	Report label
Optical library name	NAME	U	CHAR(8)	NN		NAME
Online status	ONLINE		CHAR(1)	NN	Y|N	ONLINE
Operational status	OPERATNL		CHAR(1)	NN	Y|N	OPERATIONAL
Path status	PATHSTAT**		CHAR(1)	NN	P|A	CURRENT_PATH
Current command	COMMAND		CHAR(5)	NND		CURRENT_COMMAND
Primary CTC device number	PRIMCTC		CHAR(4)	NND		PRIMARY_CTC
Primary port	PRIMPORT**		CHAR(1)	NND		PRIMARY_PORT
Alternate CTC device number	ALTCTC		CHAR(4)	NND		ALTERNATE_CTC
Alternate port	ALTPORT**		CHAR(1)	NND		ALTERNATE_PORT
Fault code	FAULT**		CHAR(3)	NND		FAULT_CODE
Library type	OLIBTYPE		CHAR(1)	NN	P|R	LIBRARY_TYPE
Number of slots	NUMSLOTS		INTEGER	NND		SLOTS
Number of empty slots	NUMESLOT		INTEGER	NND		EMPTY_SLOTS

Table 60. Library table column description (continued)						
Column description	**Name**	**Index**	**DB2 type**	**DB2 attribute**	**Value**	**Report label**
Number of drives	NUMDRVS		SMALLINT	NN		DRIVES
Current recovery command	RCOMMAND**		CHAR(5)	NND		RECOVERY_COMMAND
Device type	DEVTYPE		CHAR(8)	NND		DEVICE_TYPE
Library description	LIBRDES		CHAR(120)	NND		LIBRARY_ DESCRIPTION
Controlling Library	CLIBRARY		CHAR(8)	NND		CONTROLLING_LIBRARY
Default media type	MEDIATYP		CHAR(8)	NND		DEFAULT_MEDIA_TYPE
Library index	LIBINDEX		SMALLINT	NND		LIBRARY_INDEX
OAM XCF member name	MEMBER		CHAR(16)	NND		OAM_XCF_MEMBER
Default pseudo library	PLIBRARY		CHAR(8)	NND		DEFAULT_PSEUDO_LIBRARY

The NAME column has a unique index. Therefore, each optical library defined in the library table must have a unique name.

The online status, operational status, path status, library type, and device type columns in the library table require specific values:

ONLINE
Indicates the online status of the library to a specific system.

Y
Library is online to a system.

N
Library is offline to a system.

OPERATNL
Indicates the operational status of the library.

Y
Library is operational.

N
Library is not operational.

PATHSTAT
Indicates which path is being used.

P
Primary path is being used.

A
Alternate path is being used.

Note: If this is a pseudo library, the PATHSTAT column is blank. Path status does not apply to pseudo libraries.

OLIBTYPE
Indicates the library type.

P
This is a pseudo library.

R
This is a real library.

DEVTYPE
Indicates the device type associated with the library.

3995-111
This device is a 3995 Model 111.

3995-112
This device is a 3995 Model 112.

3995-113
This device is a 3995 Model 113.

3995-131
This device is a 3995 Model 131.

3995-132
This device is a 3995 Model 132.

3995-133
This device is a 3995 Model 133.

3995-SW3
This device is a 3995 Model SW3. Valid for pseudo libraries only.

3995-SW4
This device is a 3995 Model SW4. Valid for pseudo libraries only.

3995-C3A
This device is a 3995 Model C3A.

3995-C12
This device is a 3995 Model C12.

3995-C16
This device is a 3995 Model C16.

3995-C18
This device is a 3995 Model C18.

3995-C32
This device is a 3995 Model C32.

3995-C34
This device is a 3995 Model C34.

3995-C36
This device is a 3995 Model C36.

3995-C38
This device is a 3995 Model C38.

MEMBER
The instance of OAM, to which the library is online, that is managing the library within the Parallel Sysplex.

PLIBRARY
This is either the name of the default target pseudo library for volumes ejected from this library if the library is a real library type, or this field is blank.

Drive table (DRIVE)

The drive table contains one row for each optical drive, whether operator-accessible or library-resident. The DB2 name of the drive table is DRIVE. The table is defined in table space OCDRVTSP. Table 61 on page 497 describes the attributes of the columns in the drive table. A row is inserted into the drive table the first time that an optical disk drive with a given name is defined using the ISMF Library Management application.

Table 61. Drive table column description						
Column description	**Name**	**Index**	**DB2 type**	**DB2 attribute**	**Value**	**Report label**
Optical drive name	NAME	U	CHAR(8)	NN		NAME
Optical library name	OLIBRARY		CHAR(8)	NN		OLIBRARY
CTC device number	CTC	P	CHAR(4)	NN		CTC
SCSI bus address	SCSI**	P	CHAR(1)	NN		SCSI
Logical unit number	LUN**	P	CHAR(1)	NN		LUN
Online status	ONLINE		CHAR(1)	NN	Y|N	ONLINE
Operational status	OPERATNL		CHAR(1)	NN	Y|N	OPERATIONAL
Library drive number	LDRIVENO		CHAR(1)	NND		DRIVE_NUMBER
Drive type	DRIVTYPE		CHAR(1)	NN	L|S	DRIVE_TYPE
Device type	DEVTYPE		CHAR(8)	NND		DEVICE_TYPE
Drive description	DRIVDES		CHAR(120)	NND		DRIVE_DESCRIPTION
Physical Drive Number	DRIVENUM		SMALLINT	NND		PHY_DRIVE_NUMBER
OAM XCF member name	MEMBER		CHAR(16)	NND		OAM_XCF_MEMBER
Note: ** No longer used.						

The NAME column has a unique index. Therefore, each optical disk drive defined in the drive table must have a unique name.

The combination of CTC, SCSI, and LUN must be unique for each optical drive defined in the drive table, because there is a partitioned index on the CTC, SCSI, and LUN columns. CTC, SCSI, and LUN constitute the device address of the optical disk drive. This address is used by OAM to address the optical drive during I/O operations.

The online status, operational status, drive type, and device type columns in the drive table require specific values:

ONLINE
Indicates whether the optical drive is online or offline to a particular system.

Y
The optical drive is online to a specific system.

N
The optical drive is offline to a specific system.

OPERATNL
Indicates the operational status of the optical drive.

Y
Optical drive is operational.

N
Optical drive is not operational.

DRIVTYPE
Indicates the type of optical drive.

L
Optical drive is in a library.

S
Optical drive is a operator-accessible (stand-alone).

DEVTYPE
Indicates the device type of the optical disk drive.

3995-131
This device is a 3995 Model 131.

3995-132
This device is a 3995 Model 132.

3995-133
This device is a 3995 Model 133.

3995SW3
This device is a 3995 Model SW3.

3995SW4
This device is a 3995 Model SW4.

MEMBER
The instance of OAM, to which the drive is online, that is managing the optical drive within the Parallel Sysplex.

Slot table (SLOT)

The slot table contains one row for each slot in an optical library. The DB2 name of the slot table is SLOT. The table is defined in table space OCSLTTSP. The row for each slot gives the status of the slot. Also, a row exists for the optical library input/output station and the optical library cartridge access mechanism. Table 62 on page 498 describes the attributes of the columns in the SLOT table.

Table 62. Slot table column description

Column description	Name	Index	DB2 type	DB2 attribute	Value	Report label
Slot name	NAME	P	CHAR(3)	NN		NAME
Optical library name	OLIBRARY	P	CHAR(8)	NN		OLIBRARY
Occupied status	OCCUPIED		CHAR(1)	NN	Y|N	OCCUPIED
Operational status	OPERATNL		CHAR(1)	NN	Y|N	OPERATIONAL
Side 0 volume serial number	VOLSER0		CHAR(6)	NND		VOLSER0
Side 1 volume serial number	VOLSER1		CHAR(6)	NND		VOLSER1

Note: This table is not used for 3995 libraries.

The combination of NAME and OLIBRARY must be unique for each slot defined in the slot table because there is a partitioned index on the NAME and OLIBRARY columns.

During OAM initialization, all necessary rows are inserted into the slot table based on the optical libraries defined in the library table.

The occupied status and operational status columns in the slot table require specific values:

OCCUPIED
Indicates the status of the slot within the library.

Y
Slot is occupied.

N
Slot is not occupied.

OPERATNL
Indicates the operational status of the slot within the library.

Y
Slot is operational.

N
Slot is not operational.

Volume table (VOLUME)

The volume table contains one row for each optical disk volume. The DB2 name of the volume table is VOLUME. The table is defined in table space OCVOLTSP. Table 63 on page 499 describes the attributes of the columns in the volume table. Two rows are inserted into the volume table when the two optical volumes comprising an optical disk are identified to OAM. The two optical volumes are identified to OAM when the following conditions exist:

- The two volumes are labeled on an operator-accessible optical disk drive in response to an OAM LABEL command.
- The two volumes are labeled on a library-resident optical disk drive when the operator enters an unlabeled optical cartridge into the input/output station of an optical library.
- The two volumes comprising an already labeled, but unknown, optical cartridge are verified as part of the cartridge being entered into an optical library.

Should OAM discover minor discrepancies with the Volume table at initialization, the following recovery actions are automatically invoked to circumvent failure of the initialization:

- The row that is in error is skipped over, and a corresponding volume or tape volume control block is not built. A message is issued indicating the row that was skipped and the reason it was bypassed.
- The table row is corrected when a valid value is easily recognizable, and a message is issued stating the correction that is made by OAM and what steps can be taken if the correction is not acceptable to the customer.
- More detailed messages containing recovery actions are provided and issued during OAM initialization for database discrepancies.

Table 63. Volume table column description

Column description	Name	DB2 type	DB2 attribute	Value	Report label
Volume serial number	VOLSER	CHAR(6)	NN		VOLSER
Other side VOLSER	OVOLSER	CHAR(6)	NN		OTHER_VOLSER
Location	LOCATION	CHAR(1)	NN	L|S	LOCATION
Slot name	SLOT[2]	CHAR(3)	NN		SLOT
Library name	OLIBRARY	CHAR(8)	NN		OLIBRARY
Shelf location	SHELFLOC	CHAR(32)	NND		SHELF_LOCATION
Last-mounted date	MNTDATE	DATE	NND		DATE_LAST_MOUNTED
Last-written date	WRTDATE	DATE	NND		DATE_LAST_WRITTEN
Expiration date	EXPDATE	DATE	NND		EXPIRATION_DATE
Eject/Enter date	EJECTDAT	DATE	NND		EJECT/ENTER_DATE
Address of last data sector	LASTDATA[2]	INTEGER	NN		LAST_DATA_SECTOR
Address of last logical OVTOC sector	LASTVTCL[2]	INTEGER	NN		LAST_LOGICAL_VTOC_SECTOR
Address of last physical OVTOC sector	LASTVTCP[2]	INTEGER	NN		LAST_PHYSICAL_VTOC_SECTOR
Storage group name	VOLUMSET	CHAR(8)	NN		STORAGE_GROUP

Table 63. Volume table column description (continued)					
Column description	**Name**	**DB2 type**	**DB2 attribute**	**Value**	**Report label**
Volume type	TYPE	CHAR(1)	NN	B|G|S	TYPE
Orientation	ORIENT[2]	CHAR(1)	NN	0|1	ORIENTATION
Full status	FULL	CHAR(1)	NN	Y|N|P	FULL
Readable	READABLE	CHAR(1)	NN	Y|N	VOLUME_READABLE_STATUS
Writable	WRITABLE	CHAR(1)	NN	Y|N	VOLUME_WRITABLE_STATUS
Write-protected status	WRTPROT	CHAR(1)	NN	Y|N	WRITE_PROTECTED
Owner information part	OWNERP	CHAR(1)	NND	1|2	OWNER_INFORMATION_POSITION
Owner information	OWNER	CHAR(32)	NND		OWNER_INFORMATION
Free space	FRESPACE	INTEGER	NND		FREE_SPACE
Deleted space	DELSPACE[1]	INTEGER	NND		DELETED_SPACE
Number of deleted objects	DELCOUNT[1]	INTEGER	NND		DELETED_OBJECTS
Fragmentation index	FRAGIDX[1]	SMALLINT	NND		FRAGMENTATION_INDEX
Media type	MEDIATYP	CHAR(2)	NND		MEDIA_TYPE
Volume creation date	CREDATE	DATE	NND		CREATE_DATE
Volume error status	ERRSTAT[3]	SMALLINT	NND		VOLUME_ERROR_STATUS
Volume empty	VOLEMPTY[1]	CHAR(1)		Y|N	VOLUME_EMPTY
Deleted objects recount	RECOUNT[1]	SMALLINT		0|1	DELETED_OBJECTS_ RECOUNT
Volume Capacity	CAPACITY	INTEGER	NND		CAPACITY
OAM XCF member name	MEMBER	CHAR(16)	NND		OAM_XCF_MEMBER
Pseudo library name	PLIBRARY	CHAR(8)	NND		PSEUDO_LIBRARY_FOR_VOLUME
Backup type	BKTYPE	CHAR(1)	NND	1|2|blank	BKTYPE

Note:

- [1] DELSPACE, DELCOUNT, FRAGIDX, VOLEMPTY, and RECOUNT apply only to OAM rewritable media.
- [3] Applies only to the 3995.

The VOLSER column has a unique index. Therefore, each optical disk volume defined in the volume table must have a unique volume serial number. The optical disk volume serial number must be unique across all types of media used by the installation. The optical disk volume serial number must not conflict with the volume serial number of a tape volume being used by OAM. The optical disk volume serial number must not conflict with the serial number of any SMS-managed DASD volume or any mounted non-SMS-managed DASD volume.

The columns labeled volume location, volume type, volume orientation, volume full, volume readable, volume writable, write-protected status, owner information position, media type, volume error status, volume empty, and deleted objects recount in the volume table require specific values:

VOLSER
: The volume serial number on one side of the optical disk.

OVOLSER
: The volume serial number on the opposite side of the optical disk.

LOCATION
: The location of the optical volume

  L
  : This volume is in a library.

  S
  : This volume is on the shelf.

SLOT
The library slot location for the optical volume.

OLIBRARY
The library name in which the volume resides.

SHELFLOC
The shelf location of the shelf-resident optical volume.

MNTDATE
The date that OAM last mounted the volume.

WRTDATE
The date that OAM last wrote to the volume.

EXPDATE
The expiration date of the volume. This date is the latest expiration date of all the objects that reside on the optical volume.

EJECTDAT
This date is when the volume was last entered into or ejected out of the optical library.

LASTDATA
The address of the last data sector on the optical volume. This field is not used for 3995 volumes.

LASTVTCL
The address of the last logical optical VTOC sector on the optical volume. This field is not used for 3995 volumes.

LASTVTCP
The address of the last physical optical VTOC sector on the optical volume. This field is not used for 3995 volumes.

VOLUMESET
The name of the storage group to which the optical volume is associated.

TYPE
The type of the optical volume.

B
This is a BACKUP volume.

G
This is a GROUPED volume.

S
This is a SCRATCH volume.

FULL
The capacity of the optical volume.

Y
This volume is full.

N
This volume is not full.

READABLE
Specifies whether the optical volume is readable.

Y
This volume can be read.

N
This volume cannot be read.

P
This volume is marked permanently full.

WRITABLE
Specifies whether the optical volume is writable.

Y
This volume can be written on.

N
This volume cannot be written on.

WRTPROT
Specifies whether the optical volume is write-protected.

Y
This volume is write-protected.

N
This volume is not write-protected.

OWNERP
Position of the owner information.

1
This is part 1 of the owner information.

2
This is part 2 of the owner information.

OWNER
The volume owner information.

FRESPACE
The available free space left for writing data, in kilobyte units (1 kilobyte = 1 024 bytes) on the optical volume.

DELSPACE
The amount of deleted space on a rewritable optical disk.

Note: When updating DELSPACE, the size of a deleted object is rounded up to the next 1 kilobyte for this field, so deleting a 100 byte object will result in DELSPACE being incremented by 1 kilobyte, even though the actual amount of data deleted is significantly less than 1 kilobyte.

DELCOUNT
The amount of deleted objects marked for deletion from the rewritable optical disk.

FRAGIDX
The fragmentation index on a rewritable optical disk.

MEDIATYP
The media type of the optical volume.

01
3995 5.25-inch, single-density, rewritable volume

03
3995 5.25-inch, single-density, WORM volume

11
3995 5.25-inch, double-density, rewritable volume

13
3995 5.25-inch, double-density, WORM volume

15
3995 5.25-inch, double-density, CCW volume

21
3995 5.25-inch, quad-density, rewritable volume

23
3995 5.25-inch, quad-density, WORM volume

25
3995 5.25-inch, quad-density, CCW volume

31
: 3995 5.25-inch, 8*x*-density, rewritable volume

33
: 3995 5.25-inch, 8*x*-density, WORM volume

35
: 3995 5.25-inch, 8*x*-density, CCW volume

CREDATE
The date the optical volume was created.

ERRSTAT
The error status of the optical volume.

0
No error status. This is the initial setting.

101
This volume has an entry in the OCDB but AUDIT found no corresponding entry in the outboard inventory.

102
The cartridge is missing from its assigned slot in the library (empty slot found).

103
AUDIT found the wrong volser in the slot.

105
An error occurred when attempting to read the volume serial number while auditing a volume.

201
This volume has an entry in the OCDB but REMAP found no corresponding entry in the outboard inventory.

VOLEMPTY
Specifies whether the optical volume can be erased.

Y
This 3995 rewritable volume can be reformatted.

N
This 3995 write-once-read-many volume cannot be erased.

RECOUNT
Specifies whether a recount of the logically deleted objects, or a summing up of the available deleted kilobytes, is performed.

0
No recount of the number of logically deleted objects, or summing up of the available kilobytes that have been deleted, is performed.

1
A recount of the number of logically deleted objects, or summing up of the available kilobytes that have been deleted, is performed.

MEMBER
Specifies the name of the OAM within a Parallel Sysplex that is managing this optical volume.

- For library-resident optical volumes, this member name is the equivalent of the member name for the library in which this volume resides if the library is online. This MEMBER field is blank if the library is offline.
- For shelf-resident optical volumes that are mounted on operator-accessible drives, this member name is the equivalent of the member name for the operator-accessible drive (where the operator-accessible drive is online).
- For shelf-resident optical volumes that are not mounted on an operator-accessible drive, this member name is blank.

PLIBRARY
The name of the pseudo library the volume is assigned to when it is no longer a library-resident volume.

- For shelf-resident optical volumes, this field value is the same as the OLIBRARY column in the DB2 row that represents the volume.
- For library-resident optical volumes the field value is either of the following values:
  - The pseudo library that the volume was associated with, if it was shelf-resident prior to being entered into the library
  - Blank if the volume was not shelf-resident prior to being entered into the library.

BKTYPE
This indicates whether this volume is used for first or second backup copies of an object when the volume has a type of "B", which indicates that it is a backup volume belonging to an Object Backup storage group.

Deleted objects table (DELOBJT)

The deleted objects table contains one row for each object to be deleted. The name of the deleted objects table is DELOBJT. The table is defined in table space OCDELTSP. Table 64 on page 504 describes the attributes of the columns in the deleted objects table.

The combination of the COLNAME, OBJNAME, VOLSER, and VTOCTOKN columns must be unique throughout the table. However, multiple entries in the table might have the same VOLSER number.

Table 64. Deleted object table column description

Column description	Name	Index	DB2 type	DB2 attribute	Report label
Collection name	COLNAME	P	CHAR(44)	NN	COLLECTION_NAME
Object name	OBJNAME	P	CHAR(44)	NN	OBJECT_NAME
Volume serial number	VOLSER	PN	CHAR(6)	NN	VOLSER
VTOC token	VTOCTOKN	P	INTEGER	NN	VTOC_TOKEN
Object size	OBJSIZE		INTEGER	NN	OBJECT_SIZE

Note: The deleted-objects table is used with OAM rewritable media.

Tape volume table (TAPEVOL)

The tape volume table contains one row for each tape volume used by OAM. The DB2 name of the tape volume table is TAPEVOL. The table is defined in table space OCTVLTSP. Table 65 on page 505 describes the attributes of the columns in the OCTVLTSP table. A row is inserted into the tape volume table for each tape volume used by OAM to track its status.

Note: Should OAM discover minor discrepancies with the tape volume table at initialization, the following recovery actions are automatically invoked to circumvent failure of the initialization:

- The row that is in error is skipped over, and a corresponding volume or tape volume control block is not built. A message is issued indicating the row that was skipped and the reason it was bypassed.
- The table row is corrected when a valid value is easily recognizable, and a message is issued stating the correction that is made by OAM and what steps can be taken if the correction is not acceptable to the customer.
- More detailed messages containing recovery actions are provided and issued during OAM initialization for database discrepancies.

Column description	Name	DB2 type	DB2 attribute	Value	Report label
Volume Serial number	VOLSER	CHAR(6)	NN		VOLSER
Unit Name	UNITNAME	CHAR(8)	NN		UNIT_NAME
Media type	MEDIATYP	CHAR(2)	NN		MEDIA_TYPE
Storage group name	STORGRP	CHAR(8)	NN		STORAGE_GROUP
Volume type	TYPE	CHAR(1)	NN	B|G|S	TYPE
Volume creation date	CREDATE	DATE	NN		CREATION_DATE
Last mounted date	MNTDATE	DATE	NN		DATE_LAST_MOUNTED
Last written date	WRTDATE	DATE	NN		DATE_LAST_WRITTEN
Expiration date	EXPDATE	DATE	NN		EXPIRATION_DATE
Capacity of tape	CAPACITY	INTEGER	NN		CAPACITY
Free space remaining	FRESPACE	INTEGER	NN		FREE_SPACE
Block id of last data block written	LSTBLKID	INTEGER	NN		LAST_BLOCKID
Percent Full	PFULL	SMALLINT	NN		PERCENT_FULL
Number of logical blocks written	NUMLBLKS	INTEGER	NN		LOGICAL_BLOCKS_WRITTEN
Number of logical kilobytes written	NUMLKBW	INTEGER	NN		LOGICAL_KILOBYTES_WRITTEN
Number of physical kilobytes written	NUMPKBW	INTEGER	NN		PHYSICAL_KILOBYTES_WRITTEN
Number of logical kilobytes deleted	NUMLKBDE	INTEGER	NND		LOGICAL_KILOBYTES_DELETED
Full status	FULL	CHAR(1)	NN	Y|N|P	FULL
Readable	READABLE	CHAR(1)	NN	Y|N	VOLUME_READABLE_STATUS
Writable	WRITABLE	CHAR(1)	NN	Y|N	VOLUME_WRITABLE_STATUS
In use status	INUSE	CHAR(1)	NN		IN_USE
Copied status	COPIED	CHAR(1)	NN		COPIED
Alternate volume	AVOLSER	CHAR(6)	NN		ALTERNATE_VOLUME
Tape compaction indicator	COMPACT	CHAR(1)	NND		TAPE_COMPACTION_INDICATOR
OAM XCF member name	MEMBER	CHAR(16)	NND		OAM_XCF_MEMBER
Physical Identifier	EPI	SMALLINT	NND		EPI
Backup type	TYPE	CHAR(1)	NND	1|2|blank	BKTYPE
Original unit name	OUNITNAM	CHAR(8)	NND		ORIGINAL_UNIT_NAME
Data class	DATACLAS	CHAR(8)	NND		DATACLAS
Data set name format	DSNFMT	CHAR(1)	NND		DATASET_NAME_FORMAT
Tape sublevel	SUBLEVEL	CHAR(1)	NND		TAPE_SUBLEVEL
Capacity overflow	CAPACITYO	INTEGER	NND		CAPACITY_OVERFLOW
Freespace overflow	FRESPACEO	INTEGER	NND		FREE_SPACE_OVERFLOW
Logical Kilobytes Written Overflow	NUMLKBWO	INTEGER	NND		LOGICAL_KBS_WRITTEN_OVERFLOW
Physical Kilobytes Written Overflow	NUMPKBWO	INTEGER	NND		PHYSICAL_KBS_WRITTEN_OVERFLOW
Logical Kilobytes Deleted Overflow	NUMLKBDEO	INTEGER	NND		LOGICAL_KBS_DELETED_OVERFLOW
Volume Attribute Flags	VOLATTRF	SMALLINT	NND		VOLUME_ATTRIBUTE_FLAGS

Table 65. Tape volume table column description

The **VOLSER** column has a unique index. Therefore, each tape volume used by OAM must have a unique volume serial number. The tape volume serial number must be unique across all types of media used by the installation. The tape volume serial number must not conflict with the volume serial number of an optical volume being used by OAM. The tape volume serial number must not conflict with the serial number of any SMS-managed DASD volume or any mounted non-SMS-managed DASD volume.

Rows are dynamically inserted into the TAPEVOL table as unknown scratch tape volumes mounted in a response to a mount scratch request during allocation.

The following describes the columns in the tape volume table:

VOLSER
The volume serial number of the tape volume. All other columns in the tape volume table row apply to this volume.

UNITNAME
The MVS unit name used when the tape volume is initially mounted for OAM use. This unit name is used by OAM whenever this tape volume is subsequently allocated by OAM. This parameter is valid only for stand-alone tape drives. If the tape volume is library-resident, or if an automated or manual tape library dataserver is chosen for the request at the time of allocation, this parameter is ignored (in the case of a library-resident volume mount request), or overridden (in the case of an ATLDS or MTLDS being chosen to handle the request at allocation).

MEDIATYP
The media type of the tape volume

02
IBM Cartridge System Tape (MEDIA1)

04
IBM Enhanced Capacity Cartridge System Tape (MEDIA2)

05
IBM High Performance Cartridge Tape (MEDIA3)

06
Extended High Performance Cartridge Tape (MEDIA4)

07
Enterprise Tape Cartridge (MEDIA5)

08
Enterprise Economy Tape Cartridge (MEDIA6)

09
Enterprise WORM Tape Cartridge (MEDIA7)

10
Enterprise Economy WORM Tape Cartridge (MEDIA8)

12
IBM Enterprise Extended Tape Cartridge (MEDIA9)

14
IBM Enterprise Extended WORM Tape Cartridge (MEDIA10)

16
IBM Enterprise Advanced Tape Cartridge (MEDIA11)

17
IBM Enterprise Advanced WORM Tape Cartridge (MEDIA12)

18
IBM Enterprise Advanced Economy Tape Cartridge (MEDIA13)

STORGRP
The name of the Object or Object Backup storage group to which the tape volume is associated.

TYPE
The type of tape volume:

B
: Backup volume associated with an Object Backup storage group.

G
: Group volume associated with an Object storage group.

S
: Scratch volume that can be assigned to either an Object or Object Backup storage group when another volume is needed by OAM.

CREDATE
: The date that the volume was first used by OAM and when the row for this volume was created in the TAPEVOL table.

MNTDATE
: The date that the volume was last mounted by OAM.

WRTDATE
: The date that the volume was last written by OAM.

EXPDATE
: The expiration date of the volume. The expiration date of the volume is the latest expiration date of all objects that reside on the volume.

CAPACITY
: The approximate number of kilobytes of data which can be written for the volume allowing variance for different manufacturers.

218 554
: Represents the approximate number of kilobytes of data that can be written for an IBM Cartridge System Tape written in 18-track format on an IBM 3480 or 3490 Magnetic Tape Subsystem. The installation can overwrite this default capacity by specifying a value between 1 and 2 147 483 646 kilobytes using the TAPECAPACITY parameter of the SETOAM command.

437 109
: Represents the approximate number of kilobytes of data that can be written for an IBM Cartridge System Tape written in 36-track format on an IBM 3490E Magnetic Tape Subsystem. The installation can overwrite this default capacity by specifying a value between 1 and 2 147 483 646 kilobytes using the TAPECAPACITY parameter of the SETOAM command.

874 218
: Represents the approximate number of kilobytes of data that can be written for an IBM Enhanced Capacity Cartridge System Tape written in 36-track format on an IBM 3490E Magnetic Tape Subsystem. The installation can overwrite this default capacity by specifying a value between 1 and 2 147 483 646 kilobytes using the TAPECAPACITY parameter of the SETOAM command.

9 764 864
: Represents the approximate number of kilobytes of data for an IBM High Performance Cartridge Tape written in 128-track format on an IBM TotalStorage High Performance Tape System 3590 Model B subsystem.

  If the 3590 Model B subsystem is installed in native non-emulation mode, this value is returned from the drive and is used as an approximation that is close to the actual value.

19 530 752
: Represents the approximate number of kilobytes of data for an IBM Extended High Performance Cartridge Tape written in 128-track format on a 3590 Model B subsystem.

  If the 3590 Model B subsystem is installed in native non-emulation mode, this value is returned from the drive and is used as an approximation that is close to the actual value.

19 530 752
: Represents the approximate number of kilobytes of data for an IBM High Performance Cartridge tape written in 256-track recording technology on an IBM TotalStorage High Performance Tape System 3590 Model E subsystem.

This value is returned from the drive and is used as an approximation that is close to the actual value.

39 061 504
Represents the approximate number of kilobytes of data for an IBM Extended High Performance Cartridge tape written in 256-track recording technology on an 3590 Model E subsystem.

This value is returned from the drive and is used as an approximation that is close to the actual value.

29 296 640
Represents the approximate number of kilobytes of data for an IBM High Performance Cartridge tape written in 348-track recording technology on an IBM TotalStorage High Performance Tape System 3590 Model H subsystem.

This value is returned from the drive and is used as an approximation that is close to the actual value.

58 593 280
Represents the approximate number of kilobytes of data for an IBM Extended High Performance Cartridge tape written in 348-track recording technology on a 3590 Model H subsystem.

This value is returned from the drive and is used as an approximation that is close to the actual value.

60 653 568
Represents the approximate number of kilobytes of data for an IBM Enterprise Economy Tape Cartridge or IBM Enterprise Economy WORM Tape Cartridge written in EFMT1 recording format on an IBM 3592 Model J or Model E05 Enterprise Tape subsystem.

This value is returned from the drive and is used as an approximation that is close to the actual value.

292 968 448
Represents the approximate number of kilobytes of data for an IBM Enterprise Tape Cartridge or IBM Enterprise WORM Tape Cartridge written in EFMT1 recording format on an IBM 3592 Model J or Model E05 Enterprise Tape subsystem. This value is returned from the drive and is used here as an approximation that is close to actual value.

97 655 808
Represents the approximate number of kilobytes of data for an IBM Enterprise Economy Tape Cartridge or IBM Enterprise Economy WORM Tape Cartridge that is written in EFMT2 or EEFMT2 recording format on an IBM 3592 Model E05 or Model E06 Enterprise Tape subsystem. This value is returned from the drive and is used here as an approximation that is close to the actual value.

488 281 088
Represents the approximate number of kilobytes of data for an IBM Enterprise Tape Cartridge or IBM Enterprise WORM Tape Cartridge that is written in EFMT2 or EEFMT2 recording format on an IBM 3592 Model E05 or Model E06 Enterprise Tape subsystem. This value is returned from the drive and is used here as an approximation that is close to the actual value.

683 593 728
Represents the approximate number of kilobytes of data for an IBM Enterprise Extended Tape Cartridge or IBM Enterprise Extended WORM Tape Cartridge that is written in EFMT2 or EEFMT2 recording format on an IBM 3592 Model E05 or Model E06 Enterprise Tape subsystem. This value is returned from the drive and is used here as an approximation that is close to the actual value.

124 999 680
Represents the approximate number of kilobytes of data for an IBM Enterprise Economy Tape Cartridge or IBM Enterprise Economy WORM Tape Cartridge that is written in EFMT3 or EEFMT3 recording format on an IBM 3592 Model E06 Enterprise Tape subsystem. This value is returned from the drive and is used here as an approximation that is close to the actual value.

624 999 424
: Represents the approximate number of kilobytes of data for an IBM Enterprise Tape Cartridge or IBM Enterprise WORM Tape Cartridge that is written in EFMT3 or EEFMT3 recording format on an IBM 3592 Model E06 Enterprise Tape subsystem. This value is returned from the drive and is used here as an approximation that is close to the actual value.

976 562 176
: Represents the approximate number of kilobytes of data for an IBM Enterprise Extended Tape Cartridge or IBM Enterprise Extended WORM Tape Cartridge that is written in EFMT3 or EEFMT3 recording format on an IBM 3592 Model E06 Enterprise Tape subsystem or IBM 3592 Model E07 Enterprise Tape subsystem. This value is returned from the drive and is used here as an approximation that is close to the actual value.

1 562 499 072
: Represents the approximate number of kilobytes of data for an IBM Enterprise Extended Tape Cartridge or IBM Enterprise Extended WORM Tape Cartridge that is written in EFMT4 or EEFMT4 recording format on an IBM 3592 Model E07 Enterprise Tape subsystem. This value is returned from the drive and is used here as an approximation that is close to the actual value.

3 906 249 728
: Represents the approximate number of kilobytes of data for an IBM Enterprise Advanced Tape Cartridge or IBM Enterprise Advanced WORM Tape Cartridge that is written in EFMT4 or EEFMT4 recording format on an IBM 3592 Model E07 Enterprise Tape subsystem. This value is returned from the drive and is used here as an approximation that is close to the actual value.

488 281 088
: Represents the approximate number of kilobytes of data for an IBM Enterprise Advanced Economy Tape Cartridge that is written in EFMT4 or EEFMT4 recording format on an IBM 3592 Model E07 Enterprise Tape subsystem. This value is returned from the drive and is used here as an approximation that is close to the actual value.

Note: For tape volumes written using 18-track or 36-track format on IBM Cartridge System Tape or IBM Enhanced Capacity Cartridge System Tape on an IBM 3480, 3490, or 3490E Magnetic Tape Subsystem, the user can specify any capacity from 1 to 2 147 483 646 kilobytes of data.

The user defines the tape capacity in OAM with SETOAM TAPECAPACITY statement of SYS1.PARMLIB(CBROAM*xx*). This statement enables the user to set capacities higher or lower than the standard capacities described previously. If the user has specified a capacity with the SETOAM TAPECAPACITY statement that is higher than the tape volume is physically capable of managing , the data is written to the tape volume until the natural end of volume is reached. In this case, the displayed capacity value is the value that is indicated on the SETOAM TAPECAPACITY statement, even though it is not possible to write to that capacity.

For 3592 tape media, if the cartridge is scaled to its optimal performance capacity, the approximate number of kilobytes of data is 20% of the cartridge capacity.

FRESPACE
: The available free space left for writing data in kilobyte units on the volume. This value reflects the reduction of the percent-full (PFULL) value associated with the storage group for this tape volume.

LSTBLKID
: The block ID of the last block written on the volume.

PFULL
: An indication of what percent of the tape has been written.

NUMLBLKS
: The number of logical blocks of data that OAM has written to the volume.

NUMLKBW
: The number of logical kilobytes of data that OAM has written to the volume. This value includes OAM control information recorded on the tape volume as well as user object data.

NUMPKBW
: The number of physical kilobytes of data that has been physically recorded on the tape medium. This includes OAM control information recorded on the tape volume as well as user object data.

Note: If the data on a tape represented by a row in this table is being written in a compacted format, the number of logical KB of data on the tape and the number of physical KB of data on the tape might not be the same.

NUMLKBDE
The number of logical kilobytes (KB) of data which have been deleted from the tape volume.

Note: This number is an **approximation**. Due to the fact that the application can issue a DB2 ROLLBACK for the OSREQ DELETE, and OSMC can issue a DB2 ROLLBACK after the TAPEVOL row update for NUMLKBDE has been submitted to the OAM address space, this number might be greater than the actual amount of data which has been deleted from this tape volume. Also, the object size is rounded up to the next 1KB for this field, so deleting a 100 byte object will result in NUMLKBDE being incremented by 1, even though the actual amount of data deleted is significantly less than 1KB.

FULL
An indication of whether the volume is considered full by OAM. The possible values of this column are listed below:

Y
The volume is considered full by OAM.

N
The volume is not full.

P
The volume is marked permanently full.

READABLE
An indication of whether the volume is considered readable by OAM. The possible values are listed below:

Y
The volume is readable.

N
The volume is not readable.

WRITABLE
An indication of whether the volume is considered writable by OAM.

Y
The volume is writable.

N
The volume is not writable. The WRITABLE column is set to N when a permanent data check occurs while writing data to the volume. This prevents the volume from being selected by OAM for the writing of additional objects.

INUSE
An indication of whether the volume is in use by an OAM process. The possible values are listed below:

Y
The volume is in use by an OAM process.

N
The volume is not in use by an OAM process.

COPIED
Reserved for future use.

AVOLSER
Reserved for future use.

COMPACT
The tape compaction indicator for this tape volume. The only valid values for this field are:

Y
The tape volume was written in compacted format.

N
The tape volume was written in noncompacted format.

EPI
The ERDS Physical Identifier which indicates the real underlying device type that is used to write OAM objects to this volume. This column is used to assist in problem diagnosis in a mixed device environment where native and emulated devices coexist.

MEMBER
Indicates the name of the OAM within a Parallel Sysplex that is managing the tape volume.

- For tape volumes that are mounted and allocated on a tape drive for use by OAM, this member name is the member name of the OAM on the system to which the tape drive is online and allocated.
- For tape volumes that are not mounted and allocated on a tape drive for use by OAM, this member name is blank.

BKTYPE
The first or second backup copies of an object when the volume has a type of "B", which indicates that it is a backup volume belonging to an Object Backup storage group.

OUNITNAM
The original esoteric or generic name for the tape volume. The esoteric or generic device type is inherited from the Object or Object Backup storage group for the volume. If the allocated tape drive is in an IBM tape library, this field is stored before the UNITNAME field is changed to a generic device type. If the allocated tape drive is not in an IBM tape library, the OUNITNAM field is the same as UNITNAME field. OAM uses this field to allocate the tape volume to an Object or Object Backup storage group after OAM returns the volume to OAM scratch status. Options for initializing the OUNITNAM field include:

- Leaving the OUNITNAM field blank.
- Enabling OAMSCRATCHSYNCH mode using the SETOAM statement. OAM fills in the esoteric name automatically.
- Copying the value from the UNITNAME field to the OUNITNAM field, if you are not using tapes in an Automated Tape Library.
- Setting the OUNITNAM field to the appropriate esoteric name, if you are using tapes in an Automated Tape Library and plan to return expired tape volumes to OAM scratch status.

DATACLAS
This field represents the data class that is associated with a tape volume. The data class is inherited from the Object storage group that this volume was assigned to when it was originally allocated. If SMS does not assign a data class when it allocates this volume, the DATACLAS field is blank. OAM uses this field to allocate the tape volume to an Object or Object Backup storage group after OAM returns the volume to OAM scratch status. Options for initializing the DATACLAS field include:

- Leaving the DATACLAS field blank.
- Enabling OAMSCRATCHSYNCH mode using the SETOAM OAMSCRATCHSYNCH statement.
- Setting the DATACLAS field that is based on the Object or Object Backup storage group to which the volume belongs.

DSNFMT
This field indicates the data set name format written on the tape volume.

- If this field contains a blank, then the data set name written on the volume is the original data set name format with no storage group name low-level qualifier or the volume has no current OAM data set written on the tape volume.
- If this field contains a character "G" for group, then the data set name written on the volume has the storage group name appended.

SUBLEVEL
This field indicates with which tape sublevel this volume is associated. Tape sublevels are defined by the OAM sublevel (OSL) parameter within the SMS storage class construct definition, and associated

with the TAPEUNITNAME and L2TAPEUNITNAME keywords in the SETOAM statement of the CBROAMxx Parmlib member. Valid values are:

- **1** This volume is associated with sublevel 1.
- **2** This volume is associated with sublevel 2.
- **blank** This volume is not associated with a sublevel. Only applies to OAM scratch or backup volumes.

CAPACITYO
This field represents the overflow value for tape capacity. It reflects the number of times the capacity value has exceeded 2 gigabytes.

FRESPACEO
This field represents the overflow value for free space on the tape volume. It reflects the number of times the freespace value has exceeded 2 gigabytes.

NUMLKBWO
This field represents the overflow value for logical kilobytes written on the tape volume. It reflects the number of times the number of kilobytes logically written to the tape volume has exceeded 2 gigabytes.

NUMPKBWO
This field represents the overflow value for physical kilobytes written on the tape volume. It reflects the number of times the number of kilobytes physically written to the tape volume has exceeded 2 gigabytes.

NUMLKBDEO
This field represents the overflow value for logical kilobytes deleted from the tape volume. It reflects the number of times the number of kilobytes logically deleted from the tape volume has exceeded 2 gigabytes.

VOLATTRF
This 16-bit field contains metadata associated with this tape volume.

Bits 0-13
Reserved.

Bit 14
ON if this volume can have a block size greater than 32760.

Bit 15
ON if this is a logical WORM volume.

File system delete table (FSDELETE)

The file system delete table (FSDELETE) is used to identify objects to be ultimately physically deleted by OAM from the Unix file system hierarchy. These objects are not represented in the OAM object directory. The objects to be deleted can be a result of normal deletion activity (application initiated delete or OSMC expiration processing) or cleanup cases for objects written to the file system that have not been committed (application rollback).

The File System Delete Table has one index: a unique, ascending cluster index on the unique identifier.

Table 66. Description of file system delete table (FSDELETE)

Column description	Name	DB2 type	DB2 attribute	Value	Report label
Row ID	ID	INTEGER	Primary Key Generated Always as Identity		ID
Delete Time	DELTIME	INTEGER	NND		DELETE_TIME

Table 66. Description of file system delete table (FSDELETE) (continued)					
Column description	**Name**	**DB2 type**	**DB2 attribute**	**Value**	**Report label**
Delete Reason	DELREAS	CHAR(1)	NND	One of: D, O, or R	REASON
System Name	SYSNAME	CHAR(8)	NND		SYS_NAME
Storage Group Name	SGNAME	CHAR(8)	NND		SG_NAME
Collection Name	COLNAME	CHAR(44)	NND		COLLECTION_NAME
Object Name	OBJNAME	CHAR(44)	NND		OBJECT_NAME
Instance ID	INSTID	INTEGER	NND		INSTANCE_ID

The following describes the columns in the file system delete table:

ID
A system generated value uniquely identifying this row in the FSDELETE table .

DELTIME
The time (expressed as the high-order word of the system clock) after which the object identified by this row can be removed from the file system

DELREAS
The reason for the removal of the object from the file system:

D
This is the normal deletion of an object as a result of an application delete request for a primary object that resides in the file system

O
This is the normal deletion of an object as a result of OSMC related transition or expiration processing for a primary object that resides in the file system

R
This a "rollback" of an application store request for an object that was written to the file system when the application has issued a "rollback" or when the store has not been committed by the application within the required amount of time.

SYSNAME
The name of the system from which the request to remove the file from the file system originated

SGNAME
The name of the Object storage group in which the object to be removed from the file system resides

COLNAME
The name of the collection to which the object to be removed from the file system belongs

OBJNAME
The name of the object to be removed from the file system

INSTID
The unique identifier for this instance of the named object to be removed from the Unix file system hierarchy

Appendix E. OAM System Management Facility records

The following information provides details concerning the OAM System Management Facility (SMF) records, which measure OAM performance at the OSREQ macro interface level. The SMF Type for all Subtypes listed is SMF Type 85.

For an overview of this function, see "Measuring OAM transaction performance using SMF" on page 213.

OAM SMF record header

The OAM SMF record header, as shown in Table 67 on page 515, is at the beginning of each SMF record written by OAM:

Table 67. Header format for OAM SMF records

Offsets	Name	Length	Format	Description
0 0	SMF85LEN	2	binary	Record length.
2 2	SMF85SEG	2	binary	Segment descriptor.
4 4	SMF85FLG	1	binary	System indicator. **Bit** **Meaning WHEN SET** **0** Reserved. **1** Subtypes are valid. **2** Reserved. **3** MVS/SP Version 4 and above. bits 3, 4, 5, and 6 are on. See note. **4** MVS/SP Version 3. bits 4, 5, and 6 are on. **5** MVS/SP Version 2. bits 5 and 6 are on. **6** VS2 bit 6 is on. **7** Reserved. **Recommendation:** Use record type 30 to obtain the z/OS product level.
5 5	SMF85RTY	1	binary	Record type (decimal 85, hexadecimal X'55').
6 6	SMF85TME	4	binary	Time since midnight in hundredths of a second that the record was presented to SMF.

Table 67. Header format for OAM SMF records (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
10 A	SMF85DTE	4	binary	Date the record was presented to SMF in the form of 0*cyydddF*, where F is the sign.
14 E	SMF85SID	4	EBCDIC	System Identification (from SID parameter).
18 12	SMF85SSI	4	EBCDIC	Subsystem identification, contains ’OAM‘ for all OAM SMF records.
22 16	SMF85STY	2	binary	Record subtype.
24 18	SMF85TRN	2	binary	Number of triplets in this record. A triplet is a set of offset/length/number values that defines a section of the record.
26 1A	SMF85PSO	4	binary	Offset to OAM product section.
30 1E	SMF85PSL	2	binary	Length of OAM product section.
32 20	SMF85PSN	2	binary	Number of OAM product section.
34 22	SMF85OSO	4	binary	Offset to OAM subtype data section.
38 26	SMF85OSL	2	binary	Length of OAM subtype data section.
40 28	SMF85OSN	2	binary	Number of OAM subtype data sections.
42 2A	*	6	binary	Reserved.

OAM SMF record product section

Each OAM SMF record has a 112-byte OAM product section following the standard SMF record header. The OAM product section contains product identification information and common information to all OAM SMF record subtypes. If a field is not used for a particular subtype and the format of the field is shown as EBCDIC in the FORMAT column of the table describing the SMF record subtype, the field contains EBCDIC blanks. If a field is not used for a particular subtype and the format of the field is shown as binary in the FORMAT column of the table describing the SMF record subtype, the field contains binary zeros. The Table 68 on page 516 describes the format of the product section:

Table 68. Product section format for OAM SMF subtypes				
Offsets	**Name**	**Length**	**Format**	**Description**
0 0	R85PCID	9	EBCDIC	Component ID for OAM. For DFSMS, this field contains the characters ‘5695DF180’.
9 9	R85PVID	1	binary	Version number for DFSMS.
10 A	R85PRID	1	binary	Release number for DFSMS.
11 B	R85PMID	1	binary	Modification level for DFSMS.
12 C	R85PFMID	8	EBCDIC	SMP/E FMID for OAM.
20 14	*	4	binary	Reserved.
24 18	R85PCPUI	8	binary	CPU ID as stored by S/390® Store CPU ID (STIDP) instruction.
32 20	R85PJOBN	8	EBCDIC	Job name.

Table 68. Product section format for OAM SMF subtypes (continued)

Offsets	Name	Length	Format	Description
40 28	R85PSTPN	8	EBCDIC	Step name.
48 30	R85PPRCN	8	EBCDIC	Procedure name.
56 38	R85PPGMN	8	EBCDIC	Contains the job step program name. The job step program name is the name of the program that is specified on the job control language (JCL) EXEC statement with the PGM= keyword.
64 40	R85USRID	8	EBCDIC	User identification or blanks.
72 48	R85PTRXN	8	EBCDIC	Contains the transaction name for subtypes 2-6. The transaction names are specified as follows. For all other subtypes, this field contains blanks. **Environment** **Meaning** **CICS** The name of the CICS transaction that started the OSREQ macro. **IMS** The name of the IMS transaction that started the OSREQ macro. **OTHER** This field contains blanks, if the OSREQ macro was started from any other environment.
80 50	R85PSTRT	8	binary	Starting time of the function in 8-byte STCK (S/390 STORE CLOCK) format.
88 58	R85PENDT	8	binary	Ending time of the function in 8-byte STCK (S/390 STORE CLOCK) format.
96 60	R85PRESP	4	binary	Elapsed time of the function in milliseconds (.001 second units).
100 64	R85POSUB	4	EBCDIC	OAM subsystem ID.
104 68	R85PSSID	4	EBCDIC	DB2 subsystem ID associated with the OAM subsystem.
108 6C	*	4	binary	Reserved.
Note: STCK = S/390 STORE CLOCK				

OSREQ activity subtypes 1–10 data section format

The format of the subtype data section for all OSREQ macro functions is identical; although, not all of the fields are applicable to all OSREQ functions. Also, with the exception of the OSREQ return code (ST1RC), the rest of the fields are not valid if the OSREQ function fails. The following are subtypes and descriptions for the functions of the OSREQ macro:

1
: OSREQ ACCESS

2
: OSREQ STORE

3
OSREQ RETRIEVE

4
OSREQ QUERY

5
OSREQ CHANGE

6
OSREQ DELETE

7
OSREQ UNACCESS

8
OSREQ STOREBEG

9
OSREQ STOREPRT

10
OSREQ STOREEND

Table 69 on page 518 shows the format of the subtype data section for all OSREQ functions (subtypes 1–10):

Table 69. OSREQ activity subtypes 1-10 data section format

Offsets	Name	Length	Format	Description
0 0	ST1COLN	44	EBCDIC	Collection name. Valid for subtypes 2, 3, 4, 5, 6, 8, 9, and 10.
44 2C	ST1OBJN	44	EBCDIC	Object name. Valid for subtypes 2, 3, 4, 5, 6, 8, 9, and 10.
88 58	ST1SGN	8	EBCDIC	Storage group name. Valid for subtypes 2, 3, 4, 5, 6, 8, 9, and 10.
96 60	ST1SCN	8	EBCDIC	Storage class name. Valid for subtypes 2, 4, 5, 8, 9, and 10.
104 68	ST1MCN	8	EBCDIC	Management class name. Valid for subtypes 2, 4, 5, 8, 9, and 10.
112 70	ST1OFF	4	binary	Offset for both partial object retrieve (subtype 3), and object store part (subtype 9). Zero for all others.

Table 69. OSREQ activity subtypes 1-10 data section format (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
116 74	ST1LEN	4	binary	Length. Valid for subtypes 2, 3, 4, 6, 8, 9, and 10. **SUBTYPE** **Meaning** **1** Unused, contains binary zero. **2** Length of object stored. **3** Number of bytes retrieved. **4** Number of QEL elements returned to the application program. **5** Unused, contains binary zero. **6** Length of object deleted. **7** Unused, contains binary zero. **8** Total object length in bytes. **9** Length in bytes of the part of the object to be stored. **10** Total object length in bytes to complete storage of the object.
120 78	ST1TTOK	16	binary	OSREQ tracking token supplied with TTOKEN keyword on the OSREQ macro. **Note:** Any application programs that want to use the new TTOKEN keyword interface need to be recompiled with the new OSREQ macro. For more information concerning the TTOKEN keyword, see Table 39 on page 244 and *z/OS DFSMS OAM Application Programmer's Reference*.
136 88	ST1TOK	8	binary	OSREQ token.

Table 69. OSREQ activity subtypes 1-10 data section format (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
144 90	ST1VSN	6	EBCDIC	Volume serial number. Valid for subtypes 2, 3, and 6. For an OSREQ STORE request (subtype 2), this field contains the volume serial number of the tape or optical volume to which the primary copy of the object was stored. Only valid if bit 1 or 2 is on in field ST2FLGS. For an OSREQ RETRIEVE request (subtype 3), this field contains the volume serial number of the tape or optical volume from which the copy of the object was retrieved. Either the first or the second backup copy is retrieved as determined by the VIEW=BACKUP | BACKUP2 option indicated on the RETRIEVE request. Valid if bit 1, 2, 3, 4, 5, or 6 is on in field ST3FLGS. For an OSREQ DELETE request (subtype 6), this field contains the volume serial number of the tape or optical volume from which the primary copy of the object was deleted. Valid if bit 1 or 2 is on in field ST6FLGS.

Table 69. OSREQ activity subtypes 1-10 data section format (continued)

Offsets	Name	Length	Format	Description
150 96	ST1VMT	2	EBCDIC	Volume media type. Valid for subtype 2, 3, and 6. If a volume serial number is contained in the previous field (ST1VSN), this field contains the media type of the volume whose volume serial number is in field ST1VSN as follows: **Value** **Meaning** **01** IBM 3995 5.25-inch 650-MB rewritable optical disk media. **02** IBM 3480 Cartridge System Tape. **03** IBM 3995 5.25-inch 650-MB WORM optical disk media. **04** IBM 3480 Enhanced Capacity Cartridge System Tape. **05** IBM High Performance Cartridge Tape. **06** IBM Extended High Performance Cartridge Tape. **07** IBM Enterprise Tape Cartridge. **08** IBM Enterprise WORM Tape Cartridge. **09** IBM Enterprise Economy Tape Cartridge. **10** IBM Enterprise Economy WORM Tape Cartridge. **11** IBM 3995 5.25-inch 1300-MB rewritable optical disk media. **12** IBM Enterprise Extended Tape Cartridge. **13** IBM 3995 5.25-inch 1300-MB WORM optical disk media. **14** IBM Enterprise Extended WORM Tape Cartridge. **15** IBM 3995 5.25-inch 1300-MB CCW optical disk media. **21** IBM 3995 5.25-inch 2600-MB rewritable optical disk media. **23** IBM 3995 5.25-inch 2600-MB WORM optical disk media. **25** IBM 3995 5.25-inch 2600-MB CCW optical disk media.

Table 69. OSREQ activity subtypes 1-10 data section format (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
152 98	ST1RC	4	binary	OSREQ return code. Value in register 15 following the OSREQ macro invocation.
156 9C	ST1RS	4	binary	OSREQ reason code. Value in register 0 following the OSREQ macro invocation.
160 A0	ST1FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. Unless specified below, all bits are zero and reserved. For subtype 1, the following bit definitions apply: **Bit** **Meaning** **0** When on, the IADDRESS parameter was specified on the OSREQ request. **1-31** Reserved

Offsets	Name	Length	Format	Description
160 A0	ST2FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. Unless specified below, all bits are zero and reserved. For subtype 2, the following bit definitions apply: **Bit** **Meaning** **0** When on, the object is stored to disk. **1** When on, the object is stored to optical. **2** When on, the object is stored to tape. **3** Unused. **4** Unused. **5** When on, the OSREQ STORE request resulted in the mounting of a shelf-resident removable media volume (tape or optical) by an operator. This bit is valid only if bit 1 or 2 is on. **6** When on, the OSREQ STORE request resulted in the mounting of a library-resident removable media volume (tape or optical) inside an automated storage library. This bit is valid only if bit 1 or 2 is on. **7** When on, the OSREQ STORE request was satisfied using an already mounted removable media volume (tape or optical). This bit is valid only if bit 1 or 2 is on. **8** When on, an immediate backup copy is scheduled for this object. **9** When on, the object is stored to LOB storage structure. **10** When on, the object is stored on a sublevel 1. When bit 0 is on, the object is stored in DB2. When bit 2 is on, the object is stored to a tape sublevel 1 volume. **11** When on, the object is stored on a sublevel 2. When bit 0 is on, the object is stored in a file system. When bit 2 is on, the object is stored to a tape sublevel 2 volume. **12** Unused **13** When on, the OSREQ STORE request included a DELHOLD=HOLD parameter. A deletion-hold was in effect for this object when it was initially stored

Table 69. OSREQ activity subtypes 1-10 data section format (continued)

Table 69. OSREQ activity subtypes 1-10 data section format (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
				Bit **Meaning** **14** When on, the object was stored as a retention-protected object. **15** When on, the object was stored as a deletion-protected object. **16** When on, the object was stored as an event-based-retention object. **17** Reserved. **18** When on, a 64-bit virtual storage address was provided for the object data buffer

Table 69. OSREQ activity subtypes 1-10 data section format (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
160 A0	ST3FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. Unless specified below all bits are zero and reserved. For subtype 3, the following bit definitions apply: **Bit** **Meaning** **0** When on, the primary copy of the object was retrieved from disk. **1** When on, the primary copy of the object was retrieved from optical. **2** When on, the primary copy of the object was retrieved from tape. **3** When on, either the first or the second backup copy of the object was retrieved from optical as a result of VIEW=BACKUP or VIEW=BACKUP2 being specified on the OSREQ macro. See bit 10 to indicate which backup copy was retrieved. **4** When on, either the first or the second backup copy of the object was retrieved from tape as a result of VIEW=BACKUP or VIEW=BACKUP2 being specified on the OSREQ macro. See bit 10 to indicate which backup copy was retrieved. **5** When on, either the first or the second backup copy of the object was retrieved from optical as a result of the primary copy of the object being unavailable and the automatic access to backup being active. See bit 10 for indication which backup copy was retrieved. **6** When on, either the first or the second backup copy of the object was retrieved from tape as a result of the primary copy of the object being unavailable and the automatic access to backup being active. See bit 10 for indication which backup copy was retrieved. **7** When on, the OSREQ RETRIEVE request resulted in the mounting of a shelf-resident removable media volume (tape or optical) by an operator. This bit is valid only if bit 1, 2, 3, 4, 5, or 6 is on. **8** When on, the OSREQ RETRIEVE request resulted in the mounting of a library-resident removable media volume (tape or optical) inside an automated storage library. This bit is valid only if bit 1, 2, 3, 4, 5, or 6 is on. **9** When on, the OSREQ RETRIEVE request was satisfied using an already mounted removable

Table 69. OSREQ activity subtypes 1-10 data section format (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
				14 When on, the object is retrieved from a sublevel 1. When bit 0 is on, the object is retrieved from DB2. When bit 2 is on, the object is retrieved from a tape sublevel 1 volume. **15** When on, the object is retrieved from a sublevel 2. When bit 0 is on, the object is retrieved from a file system. When bit 2 is on, the object is retrieved from a tape sublevel 2 volume. **16** Reserved. **17** When on, a 64-bit virtual storage address was provided for the object data buffer
160 A0	ST4FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. Unless specified below, all bits are zero and reserved. For subtype 4, the following bit definitions apply: **Bit** **Meaning** **0** When on, the QUERY BACKUP OPTION has been disabled by specifying QB=N in the IEFSSNxx PARMLIB member. When off, the QUERY BACKUP OPTION is enabled, either by default or by specifying QB=Y in the IEFSSNxx PARMLIB member. **1-31** Reserved.

Table 69. OSREQ activity subtypes 1-10 data section format (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
160 A0	ST5FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. Unless specified below, all bits are zero and reserved. For subtype 5, the following bit definitions apply: **Bit** **Meaning** **0** When on, management class is specified on the OSREQ CHANGE macro. **1** When on, storage class is specified on the OSREQ CHANGE macro. **2** When on, retention period is specified on the OSREQ CHANGE macro. **3** When on, special retention period value of -1 is specified the OSREQ CHANGE macro. **4** When on, special retention period value of -2 is specified the OSREQ CHANGE macro. **5** When on, special retention period value of X'7FFFFFFF' is specified the OSREQ CHANGE macro. **6** When on, event expiration (EVENTEXP) is specified on the OSREQ CHANGE macro. **7** When on, activate deletion hold (DELHOLD=HOLD) is specified on the OSREQ CHANGE macro. **8** When on, release deletion hold (DELHOLD=NOHOLD) is specified on the OSREQ CHANGE macro.

Table 69. OSREQ activity subtypes 1-10 data section format (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
160 A0	ST6FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. Unless specified below, all bits are zero and reserved. For subtype 6, the following bit definitions apply: **Bit** **Meaning** **0** When on, the primary copy of the object is deleted from disk. **1** When on, the primary copy of the object is deleted from optical. **2** When on, the primary copy of the object is deleted from tape. **3** When on, the first backup copy of the object is deleted from optical. **4** When on, the first backup copy of the object is deleted from tape. **5** When on, the second backup copy of the object is deleted from optical. **6** When on, the second backup copy of the object is deleted from tape. **7** When on, the primary copy of the object is deleted from LOB table. **8** When on, the primary copy of the object is deleted from sublevel 1. When bit 0 is on, the object is deleted from DB2. When bit 2 is on, the object is deleted from a tape sublevel 1 volume. **9** When on, the primary copy of object is deleted from sublevel 2. When bit 0 is on, the object is deleted from a file system. When bit 2 is on, the object is deleted from a tape sublevel 2 volume. **10** Reserved.
160 A0	ST7FLGS	4	binary	Processing flags. For subtype 7, all bits contain zero.
160 A0	ST8FLGS	4	binary	Processing flags. For subtype 8, all bits contain zero.
160 A0	ST9FLGS	4	binary	Processing flags. For subtype 9, all bits contain zero.

Offsets	Name	Length	Format	Description
160 A0	ST10FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. Unless specified below, all bits are zero and reserved. For subtype 10, the following bit definitions apply: **Bit** **Meaning** **0** When on, the object is stored to disk. **1** Unused. **2** When on, the object is stored to tape. **3** Unused. **4** Unused. **5** When on, the OSREQ STORE request resulted in the mounting of a shelf-resident tape volume by an operator (this could also be a non-IBM tape library). This bit is valid only if bit 2 is on. **6** When on, the OSREQ STORE request resulted in the mounting of a library-resident tape volume inside an IBM automated tape library. This bit is valid only if bit 2 is on. **7** When on, the OSREQ STORE request was satisfied using an already mounted tape volume. This bit is valid only if bit 2 is on. **8** When on, an immediate backup copy is scheduled for this object. **9** When on, the object is stored to LOB storage structure. **10** When on, the object is stored on a sublevel 1. When bit 0 is on, the object is stored in DB2. When bit 2 is on, the object is stored to a tape sublevel 1 volume. **11** When on, the object is stored on a sublevel 2. When bit 0 is on, the object is stored in a file system. When bit 2 is on, the object is stored to a tape sublevel 2 volume. **12** When on, the CANCEL=YES keyword was specified indicating the store sequence was successfully cancelled. **13** When on, the OSREQ STOREBEG request included a DELHOLD=HOLD parameter. A deletion-hold was in effect for this object when it was initially stored

Table 69. OSREQ activity subtypes 1-10 data section format (continued)

Table 69. OSREQ activity subtypes 1-10 data section format (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
164 A4	ST1STOK	16	binary	OSREQ STOKEN. Valid for subtypes 8, 9, and 10.
180 B4	ST1RC2	4	binary	OSREQ Return Code 2. Valid for subtypes 2, 3, and 10.
184 B8	ST1STOUT	4	binary	STIMEOUT value specified on the STOREBEG request. Specifies the maximum interval in seconds between STOREBEG, STOREPRT, and STOREEND request that OAM should wait before OAM will assume that there will be no more activity for this store sequence. Valid for subtype 8.
188 BC	ST1OLRD	10	EBCDIC	Old Last Reference Date. Value contained in ODLREFDT prior to the OSREQ CHANGE or RETRIEVE request. Valid for subtypes 3 and 5.
198 C6	ST1NLRD	10	EBCDIC	New Last Reference Date. Value contained in ODLREFDT after the OSREQ CHANGE or RETRIEVE request has completed. Valid for subtypes 3 and 5.
208 D0	ST1INST	4	binary	Instance ID. Valid for subtypes 2,3,6, and 10.

Although subtypes 1–10 share a common subtype data section, not all fields are valid for each of the ten subtypes. Table 70 on page 530 identifies which fields in the OAM subtype data section are valid for each of the ten OSREQ subtypes:

Table 70. Valid subtype data section fields for OSREQ functions

Field name	**OSREQ ACCESS Subtype 1**	**OSREQ STORE Subtype 2**	**OSREQ RETRIEVE Subtype 3**	**OSREQ QUERY Subtype 4**	**OSREQ CHANGE Subtype 5**	**OSREQ DELETE Subtype 6**	**OSREQ UNACCESS Subtype 7**	**OSREQ STOREBEG Subtype 8**	**OSREQ STOREPRT Subtype 9**	**OSREQ STOREEND Subtype 10**
ST1COLN		X	X	X	X	X		X	X	X
ST1INST		X	X			X				X
ST1OBJN		X	X	X	X	X		X	X	X
ST1SGN		X	X	X	X	X		X	X	X
ST1SCN		X		X	X			X	X	X
ST1MCN		X		X	X			X	X	X
ST1OFF			X						X	
ST1LEN		X	X	X		X		X	X	X
ST1TTOK										
ST1TOK										
ST1VSN		X	X			X				
ST1VMT		X	X			X				
ST1RC	X	X	X	X	X	X	X			
ST1RS	X	X	X	X	X	X	X			
ST1FLGS	X			X						
ST1STOK								X	X	X
ST1RC2		X	X							X
ST1STOUT								X		
ST1OLRD			X		X					
ST1NLRD			X		X					
ST2FLGS		X								
ST3FLGS			X							

Table 70. Valid subtype data section fields for OSREQ functions (continued)										
Field name	**OSREQ ACCESS Subtype 1**	**OSREQ STORE Subtype 2**	**OSREQ RETRIEVE Subtype 3**	**OSREQ QUERY Subtype 4**	**OSREQ CHANGE Subtype 5**	**OSREQ DELETE Subtype 6**	**OSREQ UNACCESS Subtype 7**	**OSREQ STOREBEG Subtype 8**	**OSREQ STOREPRT Subtype 9**	**OSREQ STOREEND Subtype 10**
ST4FLGS				X						
ST5FLGS					X					
ST6FLGS						X				
ST7FLGS							X			
ST8FLGS								X		
ST9FLGS									X	
ST10FLGS										X

OSMC storage management activity (subtypes 32–35)

Table 71 on page 531 describes the format of the subtype data section for the following OAM SMF record subtypes:

32
OSMC Storage Group Processing

33
OSMC DASD Space Management Processing

34
OSMC Volume Recovery Utility

35
OSMC Move Volume Utility

Table 71. Format of the subtype data section for subtypes 32–35				
Offsets	**Name**	**Length**	**Format**	**Description**
0 0	ST32SGN	8	EBCDIC	Object or Object Backup storage group name.
8 8	ST32VSN0	6	EBCDIC	Volume serial number of a tape or optical volume. Valid for subtypes 34 and 35. This field contains blanks for all other subtypes. If the RECYCLE or DELETE option was specified, this field lists the volume serial number for the volume being recycled or deleted, and field ST32VSN1 lists the volume serial number for the opposite side of the optical volume.
14 E	ST32VSN1	6	EBCDIC	Volume serial number of the opposite side of the optical volume. Valid for subtypes 34 and 35. If the volume serial number contained in field ST32VSN0 is the volume serial number of a tape volume, this field contains blanks. If the volume serial number contained in field ST32VSN0 is the volume serial number of a tape volume, this field contains 'N/A'. If the RECYCLE or DELETE option was specified, this field lists the volume serial number of the opposite side of the optical platter.

Table 71. Format of the subtype data section for subtypes 32–35 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
20 14	ST32old media types	2	EBCDIC	Media type of the volume identified in field ST32VSN0. Valid for subtypes 34 and 35. This field contains blanks for all other subtypes. **Value** **Meaning** **01** IBM 3995 5.25-inch 650-MB rewritable optical disk media. **02** IBM 3480 Cartridge System Tape. **03** IBM 3995 5.25-inch 650-MB WORM optical disk media. **04** IBM 3480 Enhanced Capacity Cartridge System Tape. **05** IBM High Performance Cartridge Tape. **06** IBM Extended High Performance Cartridge Tape. **07** IBM Enterprise Tape Cartridge. **08** IBM Enterprise WORM Tape Cartridge. **09** IBM Enterprise Economy Tape Cartridge. **10** IBM Enterprise Economy WORM Tape Cartridge. **11** IBM 3995 5.25-inch 1300-MB rewritable optical disk media. **12** IBM Enterprise Extended Tape Cartridge. **13** IBM 3995 5.25-inch 1300-MB WORM optical disk media. **14** IBM Enterprise Extended WORM Tape Cartridge. **15** IBM 3995 5.25-inch 1300-MB CCW optical disk media. **21** IBM 3995 5.25-inch 2600-MB rewritable optical disk media. **23** IBM 3995 5.25-inch 2600-MB WORM optical disk media. **25** IBM 3995 5.25-inch 2600-MB CCW optical disk media. **31** IBM 3995 5.25-inch 5.2-GB rewritable optical disk media. **33** IBM 3995 5.25-inch 5.2-GB WORM optical disk media. **35** IBM 3995 5.25-inch 5.2-GB CCW optical disk media. **Note:** CCW = continuous composite WORM media. WORM = write-once-read-many.

Table 71. Format of the subtype data section for subtypes 32–35 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
22 16	*	2	binary	Reserved.
24 18	ST32PDWO	4	binary	Number of primary objects written to disk sublevel 1 (DB2).
28 1C	ST32PDWK	4	binary	Number of kilobytes of primary object data written to disk sublevel 1 (DB2). X'FFFFFFFF' indicates the counter has overflowed.
32 20	ST32PDRO	4	binary	Number of primary objects read from disk sublevel 1 (DB2).
36 24	ST32PDRK	4	binary	Number of kilobytes of primary object data read from disk sublevel 1 (DB2). X'FFFFFFFF' indicates the counter has overflowed.
40 28	ST32PDDO	4	binary	Number of primary objects deleted from disk sublevel 1 (DB2).
44 2C	ST32PDDK	4	binary	Number of kilobytes of primary object data deleted from disk sublevel 1 (DB2). X'FFFFFFFF' indicates the counter has overflowed.
48 30	ST32POWO	4	binary	Number of primary objects written to optical.
52 34	ST32POWK	4	binary	Number of kilobytes of primary object data written to optical. X'FFFFFFFF' indicates the counter has overflowed.
56 38	ST32PORO	4	binary	Number of primary objects read from optical.
60 3C	ST32PORK	4	binary	Number of kilobytes of primary object data read from optical. X'FFFFFFFF' indicates the counter has overflowed.
64 40	ST32PODO	4	binary	Number of primary objects deleted from optical.
68 44	ST32PODK	4	binary	Number of kilobytes of primary object data deleted from optical. X'FFFFFFFF' indicates the counter has overflowed.
72 48	ST32PTWO	4	binary	Number of primary objects written to tape.
76 4C	ST32PTWK	4	binary	Number of kilobytes of primary object data written to tape. X'FFFFFFFF' indicates the counter has overflowed.
80 50	ST32PTRO	4	binary	Number of primary objects read from tape.
84 54	ST32PTRK	4	binary	Number of kilobytes of primary object data read from tape. X'FFFFFFFF' indicates the counter has overflowed.
88 58	ST32PTDO	4	binary	Number of primary objects deleted from tape.
92 5C	ST32PTDK	4	binary	Number of kilobytes of primary object data deleted from tape. X'FFFFFFFF' indicates the counter has overflowed.
96 60	ST32BOWO	4	binary	Number of backup objects written to optical.
100 64	ST32BOWK	4	binary	Number of kilobytes of backup object data written to optical. X'FFFFFFFF' indicates the counter has overflowed.
104 68	ST32BORO	4	binary	Number of backup objects read from optical.
108 6C	ST32BORK	4	binary	Number of kilobytes of backup object data read from optical. X'FFFFFFFF' indicates the counter has overflowed.
112 70	ST32BODO	4	binary	Number of backup objects deleted from optical.
116 74	ST32BODK	4	binary	Number of kilobytes of backup object data deleted from optical. X'FFFFFFFF' indicates the counter has overflowed.
120 78	ST32BTWO	4	binary	Number of backup objects written to tape.

Table 71. Format of the subtype data section for subtypes 32–35 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
124 7C	ST32BTWK	4	binary	Number of kilobytes of backup object data written to tape. X'FFFFFFFF' indicates the counter has overflowed.
128 80	ST32BTRO	4	binary	Number of backup objects read from tape.
132 84	ST32BTRK	4	binary	Number of kilobytes of backup object data read from tape. X'FFFFFFFF' indicates the counter has overflowed.
136 88	ST32BTDO	4	binary	Number of backup objects deleted from tape.
140 8C	ST32BTDK	4	binary	Number of kilobytes of backup object data deleted from tape. X'FFFFFFFF' indicates the counter has overflowed.
144 90	ST32B2OWO	4	binary	Number of BACKUP2 objects written to optical.
148 94	ST32B2OWK	4	binary	Number of kilobytes of BACKUP2 objects written to optical. X'FFFFFFFF' indicates the counter has overflowed.
152 98	ST32B2ORO	4	binary	Number of BACKUP2 objects read from optical.
156 9C	ST32B2ORK	4	binary	Number of rows kilobytes of BACKUP2 objects read from optical. X'FFFFFFFF' indicates the counter has overflowed.
160 A0	ST32B2ODO	4	binary	Number of BACKUP2 objects deleted from optical.
164 A4	ST32B2ODK	4	binary	Number of kilobytes of BACKUP2 objects deleted from optical. X'FFFFFFFF' indicates the counter has overflowed.
168 A8	ST32B2TWO	4	binary	Number of BACKUP2 objects written to tape.
172 AC	ST32B2TWK	4	binary	Number of kilobytes of BACKUP2 objects written to tape. X'FFFFFFFF' indicates the counter has overflowed.
176 B0	ST32B2TRO	4	binary	Number of BACKUP2 objects read from tape.
180 B4	ST32B2TRK	4	binary	Number of kilobytes of BACKUP2 objects read from tape. X'FFFFFFFF' indicates the counter has overflowed.
184 B8	ST32B2TDO	4	binary	Number of BACKUP2 objects logically deleted from tape.
188 BC	ST32B2TDK	4	binary	Number of kilobytes of BACKUP2 objects logically deleted from tape. X'FFFFFFFF' indicates the counter has overflowed.
192 C0	ST32DTUP	4	binary	Number of rows updated in the object directory table.
196 C4	ST32DTDE	4	binary	Number of rows deleted from the object directory table.
200 C8	ST324KIN	4	binary	Number of rows inserted into the 4 KB object storage table.
204 CC	ST324KDE	4	binary	Number of rows deleted from the 4 KB object storage table.
208 D0	ST3232KI	4	binary	Number of rows inserted into the 32 KB object storage table.
212 D4	ST3232KD	4	binary	Number of rows deleted from the 32 KB object storage table.
216 D8	ST32NCE	4	binary	Number of optical cartridges expired. Valid only for Subtype 32.

Offsets	Name	Length	Format	Description
220 DC	ST32FLGS	4	binary	Processing flags. **Bit** **Meaning** **0** When on, the MOVEVOL was invoked automatically under software control as a result of RECYCLE. **1** When on, this process was invoked by a MODIFY OAM,START command issued from an MVS console. **2** When on, this process was invoked using an ISMF line operator. **3** When on, volume recovery was invoked with the BACKUP1 keyword or defaulted to BACKUP1. **Note:** For subtype 34, optional parameters BACKUP1 and BACKUP2 are ignored when recovering a backup volume. **4** When on, volume recovery was invoked with the BACKUP2 keyword. **5** When on, the DELETE option was specified for the RECOVER or MOVEVOL utility. **6** When on, the RECYCLE option was specified for the MOVEVOL utility. **7** When on, the Object storage group was processed. **8** When on, the Object Backup storage group was processed. **9** When on, the storage group cycle ended because the CYCLE END TIME was exceeded. **10** When on, automatic backup deletion was enabled during the storage group cycle. **11–31** Reserved.
224 E0	ST32NTE	4	binary	Number of tape volumes expired. Valid only for subtype 32.
228 E4	ST32RCLD	4	binary	Number of recalled objects processed this storage group cycle. Valid only for subtype 32.
232 E8	ST32RCLK	4	binary	Number of kilobytes of recalled objects processed this storage group cycle. Valid only for subtype 32. X'FFFFFFFF' indicates the counter has overflowed.
236 EC	ST32LOBI	4	binary	Number of rows inserted into the LOB storage structure.
240 F0	ST32LOBD	4	binary	Number of rows deleted from the LOB storage structure.
244 F4	ST32PUWO	4	binary	Number of primary objects written to tape sublevel 2.

Table 71. Format of the subtype data section for subtypes 32–35 (continued)

Table 71. Format of the subtype data section for subtypes 32–35 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
248 F8	ST32PUWK	4	binary	Number of kilobytes of primary objects written to tape sublevel 2. X'FFFFFFFF' indicates the counter has overflowed.
252 FC	ST32PURO	4	binary	Number of primary objects read from tape sublevel 2.
256 100	ST32PURK	4	binary	Number of kilobytes of primary objects read from tape sublevel 2. X'FFFFFFFF' indicates the counter has overflowed.
260 104	ST32PUDO	4	binary	Number of primary objects deleted from tape sublevel 2.
264 108	ST32PUDK	4	binary	Number of kilobytes of primary object data deleted from tape sublevel 2. X'FFFFFFFF' indicates the counter has overflowed.
268 10C	ST32PEWO	4	binary	Number of primary objects written to disk sublevel 2 (file system).
272 110	ST32PERO	4	binary	Number of primary objects read from disk sublevel 2 (file system).
276 114	ST32PEDO	4	binary	Number of primary objects deleted from disk sublevel 2 (file system).
280 118	ST32PDWB	8	binary	Number of bytes of primary object data written to disk sublevel 1 (DB2).
288 120	ST32PDRB	8	binary	Number of bytes of primary object data read from disk sublevel 1 (DB2).
296 128	ST32PDDB	8	binary	Number of bytes of primary object data deleted from disk sublevel 1 (DB2).
304 130	ST32POWB	8	binary	Number of bytes of primary object data written to optical.
312 138	ST32PORB	8	binary	Number of bytes of primary object data read from optical.
320 140	ST32PODB	8	binary	Number of bytes of primary object data deleted from optical.
328 148	ST32PTWB	8	binary	Number of bytes of primary object data written to tape.
336 150	ST32PTRB	8	binary	Number of bytes of primary object data read from tape.
344 158	ST32PTDB	8	binary	Number of bytes of primary object data deleted from tape.
352 160	ST32BOWB	8	binary	Number of bytes of backup object data written to optical.
360 168	ST32BORB	8	binary	Number of bytes of backup object data read from optical.
368 170	ST32BODB	8	binary	Number of bytes of backup object data deleted from optical.
376 178	ST32BTWB	8	binary	Number of bytes of backup object data written to tape.
384 180	ST32PTRB	8	binary	Number of bytes of backup object data read from tape.
392 188	ST32BTDB	8	binary	Number of bytes of backup object data deleted from tape.
400 190	ST32B2OWB	8	binary	Number of bytes of BACKUP2 objects written to optical.
408 198	ST32B2ORB	8	binary	Number of bytes of BACKUP2 objects read from optical.
416 1A0	ST32B2ODB	8	binary	Number of bytes of BACKUP2 objects deleted from optical.
424 1A8	ST32B2TWB	8	binary	Number of bytes of BACKUP2 objects written to tape.
432 1B0	ST32B2TRB	8	binary	Number of bytes of BACKUP2 objects read from tape.
440 1B8	ST32B2TDB	8	binary	Number of bytes of BACKUP2 objects logically deleted from tape.

Table 71. Format of the subtype data section for subtypes 32–35 (continued)

Offsets	Name	Length	Format	Description
448 1C0	ST32RCLB	8	binary	Number of bytes of recalled objects processed this storage group cycle. Valid only for subtype 32.
456 1C8	ST32PUWB	8	binary	Number of bytes of primary objects written to tape sublevel 2.
464 1D0	ST32PURB	8	binary	Number of bytes of primary objects read from tape sublevel 2.
472 1D8	ST32PUDB	8	binary	Number of bytes of objects deleted from tape sublevel 2.
480 1E0	ST32PEWB	8	binary	Number of bytes of primary objects written to disk sublevel 2 (file system).
488 1E8	ST32PERB	8	binary	Number of bytes of primary objects read from disk sublevel 2 (file system).
496 1F0	ST32PEDB	8	binary	Number of bytes of primary objects deleted from disk sublevel 2 (file system).
504 1F8	ST32BOAO	4	Binary	Number of unneeded backup 1 copies deleted from optical. Valid for subtype 32 only. Note: These backup copies are also included in the count in the ST32BODO field.
508 1FC	ST32B2OAO	4	Binary	Number of unneeded backup 2 copies deleted from optical. Valid for subtype 32 only. Note: These backup copies are also included in the count in the ST32B2ODO field.
512 200	ST32BTAO	4	Binary	Number of unneeded backup 1 copies deleted from tape. Valid for subtype 32 only. Note: These backup copies are also included in the count in the ST32BTDO field.
516 204	ST32B2TAO	4	Binary	Number of unneeded backup 2 copies deleted from tape. Valid for subtype 32 only. Note: These backup copies are also included in the count in the ST32B2TDO field.
520 208	ST32BOAB	8	Binary	Number of bytes of unneeded backup 1 copies deleted from optical. Valid for subtype 32 only. Note: These bytes are also included in the count in the ST32BODB field.
528 210	ST32B2OAB	8	Binary	Number of bytes of unneeded backup 2 copies deleted from optical. Valid for subtype 32 only. Note: These bytes are also included in the count in the ST32B2ODB field.
536 218	ST32BTAB	8	Binary	Number of bytes of unneeded backup 1 copies deleted from tape. Valid for subtype 32 only. Note: These bytes are also included in the count in the ST32BTDB field.
544 220	ST32B2TAB	8	Binary	Number of bytes of unneeded backup 2 copies deleted from tape. Valid for subtype 32 only. Note: These bytes are also included in the count in the ST32B2TDB field.
Note: For subtypes 32-35, the total value in fields that refer to 'number of kilobytes' or 'number of bytes' will be rounded up.				

OSMC single object recovery utility (subtype 36)

Table 72 on page 538 describes the format of the subtype data section for a subtype 36 OAM SMF record for the single object recovery utility.

Offsets	Name	Length	Format	Description
0 0	ST36COLN	44	EBCDIC	Collection name.
44 2C	ST36CNID	4	binary	Collection ID.
48 30	ST36OBJN	44	EBCDIC	Object name.
92 5C	ST36SGN	8	EBCDIC	OBJECT storage group name.
100 64	ST36OLEN	4	binary	Object length.
104 68	ST36BVSN	6	EBCDIC	Volume serial number of the optical or tape volume from which the backup copy of the object was read. The backup copy can be either the first or the second backup copy as determined by options specified on the F OAM,START,OBJRECV command. The options are: BACKUP1 | BACKUP2.

Table 72. Format of the subtype data section for subtype 36

Table 72. Format of the subtype data section for subtype 36 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
110 6E	ST36BMT	2	EBCDIC	Media type of volume from which the backup copy of the object was read: **Value** **Meaning** **01** IBM 3995 5.25-inch 650-MB rewritable optical disk media. **02** IBM 3480 cartridge System Tape. **03** IBM 3995 5.25-inch 650-MB WORM optical disk media. **04** IBM 3480 Enhanced Capacity Cartridge System Tape. **05** IBM High Performance Cartridge Tape. **06** IBM Extended High Performance Cartridge Tape. **07** IBM Enterprise Tape Cartridge. **08** IBM Enterprise WORM Tape Cartridge. **09** IBM Enterprise Economy Tape Cartridge. **10** IBM Enterprise Economy WORM Tape Cartridge. **11** IBM 3995 5.25-inch 1300-MB rewritable optical disk media. **12** IBM Enterprise Extended Tape Cartridge. **13** IBM 3995 5.25-inch 1300-MB WORM optical disk media. **14** IBM Enterprise Extended WORM Tape Cartridge. **15** IBM 3995 5.25-inch 1300-MB CCW optical disk media. **21** IBM 3995 5.25-inch 2600-MB rewritable optical disk media. **23** IBM 3995 5.25-inch 2600-MB WORM optical disk media. **25** IBM 3995 5.25-inch 2600-MB CCW optical disk media. **31** IBM 3995 5.25-inch 5.2-GB rewritable optical disk media. **33** IBM 3995 5.25-inch 5.2-GB WORM optical disk media. **35** IBM 3995 5.25-inch 5.2-GB CCW optical disk media. **Note:** CCW = continuous composite WORM media. WORM = write-once-read-many.

Offsets	Name	Length	Format	Description
112 70	ST36BTKN	4	binary	Volume location token associated with the backup copy of the object on the volume specified in the ST36BVSN field.
116 78	ST36TVSN	6	EBCDIC	Volume serial number of the target optical or tape volume to which the new primary copy of the object was written. This field contains blanks if the new location is on a disk sublevel.

Table 72. Format of the subtype data section for subtype 36 (continued)

Table 72. Format of the subtype data section for subtype 36 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
122 7E	ST36TMT	2	EBCDIC	Media type of target optical or tape volume to which the new primary copy of the object was written. This field contains blanks if the new primary copy of the object was written to a disk sublevel: **Value** **Meaning** **01** IBM 3995 5.25-inch 650-MB rewritable optical disk media. **02** IBM 3480 Cartridge System Tape. **03** IBM 3995 5.25-inch 650-MB WORM optical disk media. **04** IBM 3480 Enhanced Capacity Cartridge System Tape. **05** IBM High Performance Cartridge Tape. **06** IBM Extended High Performance Cartridge Tape. **07** IBM Enterprise Tape Cartridge. **08** IBM Enterprise WORM Tape Cartridge. **09** IBM Enterprise Economy Tape Cartridge. **10** IBM Enterprise Economy WORM Tape Cartridge. **11** IBM 3995 5.25-inch 1300-MB rewritable optical disk media. **12** IBM Enterprise Extended Tape Cartridge. **13** IBM 3995 5.25-inch 1300-MB WORM optical disk media. **14** IBM Enterprise Extended WORM Tape Cartridge. **15** IBM 3995 5.25-inch 1300-MB CCW optical disk media. **21** IBM 3995 5.25-inch 2600-MB rewritable optical disk media. **23** IBM 3995 5.25-inch 2600-MB WORM optical disk media. **25** IBM 3995 5.25-inch 2600-MB CCW optical disk media. **31** IBM 3995 5.25-inch 5.2-GB rewritable optical disk media. **33** IBM 3995 5.25-inch 5.2-GB WORM optical disk media. **35** IBM 3995 5.25-inch 5.2-GB CCW optical disk media. **Note:** CCW = continuous composite WORM media. WORM = write-once-read-many.

Offsets	Name	Length	Format	Description
124 80	ST36OVSN	6	EBCDIC	Volume serial number of the original optical or tape volume on which the primary copy of the object resided prior to the start of the single object recovery utility. This field contains blanks if the original location was on a disk sublevel.

Table 72. Format of the subtype data section for subtype 36 (continued)

Table 72. Format of the subtype data section for subtype 36 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
130 82	ST36OMT	2	EBCDIC	Media type of the original optical or tape volume on which the primary copy of the object resided prior to the start of the single object recovery utility. This field contains blanks if the primary copy of the object resides on a disk sublevel: **Value** **Meaning** **01** IBM 3995 5.25-inch 650-MB rewritable optical disk media. **02** IBM 3480 Cartridge System Tape. **03** IBM 3995 5.25-inch 650-MB WORM optical disk media. **04** IBM 3480 Enhanced Capacity Cartridge System Tape. **05** IBM High Performance Cartridge Tape. **06** IBM Extended High Performance Cartridge Tape. **07** IBM Enterprise Tape Cartridge. **08** IBM Enterprise WORM Tape Cartridge. **09** IBM Enterprise Economy Tape Cartridge. **10** IBM Enterprise Economy WORM Tape Cartridge. **11** IBM 3995 5.25-inch 1300-MB rewritable optical disk media. **12** IBM Enterprise Extended Tape Cartridge. **13** IBM 3995 5.25-inch 1300-MB WORM optical disk media. **14** IBM Enterprise Extended WORM Tape Cartridge. **15** IBM 3995 5.25-inch 1300-MB CCW optical disk media. **21** IBM 3995 5.25-inch 2600-MB rewritable optical disk media. **23** IBM 3995 5.25-inch 2600-MB WORM optical disk media. **25** IBM 3995 5.25-inch 2600-MB CCW optical disk media. **31** IBM 3995 5.25-inch 5.2-GB rewritable optical disk media. **33** IBM 3995 5.25-inch 5.2-GB WORM optical disk media. **35** IBM 3995 5.25-inch 5.2-GB CCW optical disk media. **Note:** CCW = continuous composite WORM media. WORM = write-once-read-many.

Table 72. Format of the subtype data section for subtype 36 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
132 84	ST36FLGS	4	binary	Processing flags. **Bit** **Meaning** **0** When on, object recovery was invoked with the BACKUP1 keyword or defaulted to BACKUP1. **1** When on, object recovery was invoked with the BACKUP2 keyword. **2–31** Reserved.
136 88	ST36DSL	2	EBCDIC	Disk sublevel associated with the recovered object. This field is valid only when the ST36TVSN field contains blanks. **Value** **Meaning** **01** Recovered object on disk sublevel 1 (DB2). **02** Recovered object on disk sublevel 2 (file system).

OSMC library space management (subtype 37)

Table 73 on page 544 describes the format of the subtype data section for a subtype 37 OAM SMF record for OSMC library space management.

Table 73. Format of the subtype data section for subtype 37				
Offsets	**Name**	**Length**	**Format**	**Description**
0 0	ST37LIBN	8	EBCDIC	Library name.
8 8	ST37LIBD	8	EBCDIC	Library device type.
16 10	ST37NOCE	4	binary	Number of optical disk cartridges ejected.
20 14	ST37FLGS	4	binary	Processing flags. **Bit** **Meaning** **0** When on, library space management is invoked automatically under software control due to a storage group out-of-space condition in the specified library. **1** When on, library space management is invoked by a **F OAM,START,LIBMGT** command issued from an MVS console. **2–31** Reserved.

OSMC RECALL to the disk sublevel (subtype 38)

Table 74 on page 545 describes the format of the subtype data section for a subtype 38 OAM SMF record for OSMC RECALL to the disk sublevel.

Table 74. Format of the subtype data section for subtype 38

Offsets	Name	Length	Format	Description
0 0	ST38COLN	44	EBCDIC	Collection name.
44 2C	ST38CNID	4	binary	Collection ID.
48 30	ST38OBJN	44	EBCDIC	Object name.
92 5C	ST38SGN	8	EBCDIC	Object storage group name.
100 64	ST38OLEN	4	binary	Object length.
104 68	ST38VSN	6	EBCDIC	Volume serial number of the optical or tape volume from which the copy of the object was read.

Table 74. Format of the subtype data section for subtype 38 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
110 6E	ST38MT	2	EBCDIC	Media type of volume from which the copy of the object was read:. **Value** **Meaning** **01** IBM 3995 5.25–inch 650–MB rewritable optical disk media. **02** IBM 3480 Cartridge System Tape. **03** IBM 3995 5.25–inch 650–MB WORM optical disk media. **04** IBM 3480 Enhanced Capacity Cartridge System Tape. **05** IBM High Performance Cartridge Tape. **06** IBM Extended High Performance Cartridge Tape. **07** IBM Enterprise Tape Cartridge. **08** IBM Enterprise WORM Tape Cartridge. **09** IBM Enterprise Economy Tape Cartridge. **10** IBM Enterprise Economy WORM Tape Cartridge. **11** IBM 3995 5.25–inch 1300–MB rewritable optical disk media. **12** IBM Enterprise Extended Tape Cartridge. **13** IBM 3995 5.25-inch 1300-MB WORM optical disk media. **14** IBM Enterprise Extended WORM Tape Cartridge. **15** IBM 3995 5.25–inch 1300–MB CCW optical disk media. **21** IBM 3995 5.25–inch 2600–MB rewritable optical disk media. **23** IBM 3995 5.25–inch 2600–MB WORM optical disk media. **25** IBM 3995 5.25–inch 2600–MB CCW optical disk media. **31** IBM 3995 5.25–inch 5.2–GB rewritable optical disk media. **33** IBM 3995 5.25–inch 5.2–GB WORM optical disk media. **Note:** WORM=write once read many CCW=continuous composite WORM media

Table 74. Format of the subtype data section for subtype 38 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
112 70	ST38TKN	4	binary	Volume location token associated with the copy of the object on the volume specified in the ST36BVSN field.
116 74	ST38RCLD	4	binary	Number of days specified for object recall.
120 78	ST38VT	1	EBCDIC	Volume type:. **Value** **Meaning** **G** Volume is a grouped volume belonging to an OBJECT storage group. **B** Volume is a backup volume belonging to an OBJECT BACKUP storage group.
121 79	ST38BT	1	EBCDIC	Backup type:. **Value** **Meaning** **1** Volume belonging to a backup one OBJECT BACKUP storage group. **2** Volume belonging to a backup two OBJECT BACKUP storage group.
122 7A	ST38DSL	2	EBCDIC	Disk sublevel associated with the recalled object. **Value** **Meaning** **01** Recalled object on disk sublevel 1 (DB2). **02** Recalled object on disk sublevel 2 (file system).
124 7C	ST38FLGS	4	binary	Processing flags:. **Bit** **Meaning** **0** When on, Object Recall was successful. **1–31** Reserved.

Immediate backup copy (subtype 39)

Table 75 on page 547 describes the format of the subtype data section for a subtype 39 OAM SMF record for Immediate Backup Copy.

Table 75. Format of the subtype data section for subtype 39				
Offsets	**Name**	**Length**	**Format**	**Description**
0 0	ST39COLN	44	EBCDIC	Collection name.
44 2C	ST39CNID	4	binary	Collection ID.

Table 75. Format of the subtype data section for subtype 39 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
48 30	ST39OBJN	44	EBCDIC	Object name.
92 5C	ST39SGN	8	EBCDIC	OBJECT storage group name.
100 64	ST39MCN	8	EBCDIC	Management Class name.
108 6C	ST39OLEN	4	binary	Object length.
112 70	ST39SVSN	6	EBCDIC	Source Volume serial number of the optical or tape volume on which the primary object was read. Only valid if the bit 1 or 2 is ON in field ST39FLGS.

Offsets	Name	Length	Format	Description
118 76	ST39SMT	2	EBCDIC	Source volume Media type. Only valid if the bit 1 or 2 is ON in field ST39FLGS. **Value** **Meaning** **01** IBM 3995 5.25–inch 650–MB rewritable optical disk media. **02** IBM 3480 Cartridge System Tape. **03** IBM 3995 5.25–inch 650–MB WORM optical disk media. **04** IBM 3480 Enhanced Capacity Cartridge System Tape. **05** IBM High Performance Cartridge Tape. **06** IBM Extended High Performance Cartridge Tape. **07** IBM Enterprise Tape Cartridge. **08** IBM Enterprise WORM Tape Cartridge. **09** IBM Enterprise Economy Tape Cartridge. **10** IBM Enterprise Economy WORM Tape Cartridge. **11** IBM 3995 5.25–inch 1300–MB rewritable optical disk media. **12** IBM Enterprise Extended Tape Cartridge. **13** IBM 3995 5.25-inch 1300-MB WORM optical disk media. **14** IBM Enterprise Extended WORM Tape Cartridge. **15** IBM 3995 5.25–inch 1300–MB CCW optical disk media. **21** IBM 3995 5.25–inch 2600–MB rewritable optical disk media. **23** IBM 3995 5.25–inch 2600–MB WORM optical disk media. **25** IBM 3995 5.25–inch 2600–MB CCW optical disk media. **31** IBM 3995 5.25–inch 5.2–GB rewritable optical disk media. **33** IBM 3995 5.25–inch 5.2–GB WORM optical disk media. **35** IBM 3995 5.25–inch 5.2–GB CCW optical disk media.

Table 75. Format of the subtype data section for subtype 39 (continued)

Table 75. Format of the subtype data section for subtype 39 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
120 78	ST39TVSN	6	EBCDIC	Target Volume serial number of the optical or tape volume on which the backup copy of the object was written.
126 7E	ST39TMT	2	EBCDIC	Target Volume Media type: Refer to ST39SMT for the values.
128 80	ST39BTKN	4	binary	Volume location token on the ST39TVSN.
132 84	ST39FLGS	4	binary	Processing flags: **Bit** **Meaning** **0** When on, the primary copy is stored to disk. **1** When on, the primary copy is stored to optical. **2** When on, the primary copy is stored to tape. **3** Reserved. **4** Reserved. **5** When on, the backup copy is stored to optical. **6** When on, the backup copy is stored to tape. **7** Reserved. **8** When on, write to backup was successful. **9** Reserved. **10** When on, the primary copy is stored on sublevel 1. When bit 0 is on, the primary copy is stored in DB2. When bit 2 is on, the primary copy is stored to a tape sublevel 1 volume. **11** When on, the primary copy is stored on sublevel 2. When bit 0 is on, the primary copy is stored in a file system. When bit 2 is on, the primary copy is stored to a tape sublevel 2 volume.

OSMC tape recycle (subtype 40)

Table 76 on page 550 describes the format of the subtype data section for a subtype 40 OAM SMF record for Tape Recycle.

Table 76. Format of the subtype data section for subtype 40				
Offsets	**Name**	**Length**	**Format**	**Description**
0 0	ST40STRD	10	EBCDIC	DATE RECYCLE CMD STARTED.
10 A	ST40ENDD	10	EBCDIC	DATE RECYCLE CMD ENDED.

Table 76. Format of the subtype data section for subtype 40 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
20 14	ST40VOLN	2	BINARY	NUMBER OF VOLS COMPLETED.
22 16	ST40PCTV	2	BINARY	PERCENTVALID .
24 18	ST40LIM	2	BINARY	LIMIT.
26 1A	ST40SUBL	1	EBCDIC	TSL-TAPE SUBLEVEL.
27 1B		1	BINARY	RESERVED.
28 1C	ST40VSN	240	EBCDIC	ARRAY of up to 40 volume serials that are completed.

LCS optical library/drive VARY online/offline (subtypes 64–67)

Table 77 on page 551 describes the format of the subtype data section for the following subtypes:

64
LCS optical drive vary online.

65
LCS optical drive vary offline.

66
LCS optical library vary online.

67
LCS optical library vary offline.

Table 77. Format of the subtype data section for subtypes 64–67				
Offsets	**Name**	**Length**	**Format**	**Description**
0 0	ST64OLN	8	EBCDIC	Contains the real optical library name for the operator-accessible drive.
8 8	ST64OLDT	8	EBCDIC	Optical library device type.
16 10	ST64OLDN	4	EBCDIC	MVS device number corresponding to the optical library.
20 14	ST64ODN	8	EBCDIC	Optical drive name. Valid for subtypes 64 and 65. This field contains blanks for other subtypes.
28 1C	ST64ODDT	8	EBCDIC	Optical drive device type. Valid for subtypes 64 and 65. This field contains blanks for other subtypes.
36 24	ST64ODDN	4	EBCDIC	MVS device number corresponding to the optical drive. Valid for subtypes 64 and 65. This field contains blanks for other subtypes.
40 28	ST64VSN0	6	EBCDIC	Volume serial number of the currently mounted volume. Valid for subtypes 64 and 65. This field contains blanks for other subtypes.
46 2E	ST64VSN1	6	EBCDIC	Volume serial number of the opposite side of the currently mounted volume. Valid for subtypes 64 and 65. This field contains blanks for other subtypes.

Table 77. Format of the subtype data section for subtypes 64–67 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
52 34	ST64OMT	2	EBCDIC	Optical media type. Valid for subtypes 64 and 65. This field contains blanks for other subtypes. **Value** **Meaning** **01** IBM 3995 5.25-inch 650-MB rewritable optical disk media. **03** IBM 3995 5.25-inch 650-MB WORM optical disk media. **11** IBM 3995 5.25-inch 1300-MB rewritable optical disk media. **13** IBM 3995 5.25-inch 1300-MB WORM optical disk media. **15** IBM 3995 5.25-inch 1300-MB CCW optical disk media. **21** IBM 3995 5.25-inch 2600-MB rewritable optical disk media. **23** IBM 3995 5.25-inch 2600-MB WORM optical disk media. **25** IBM 3995 5.25-inch 2600-MB CCW optical disk media. **31** IBM 3995 5.25-inch 5.2-GB rewritable optical disk media. **33** IBM 3995 5.25-inch 5.2-GB WORM optical disk media. **35** IBM 3995 5.25-inch 5.2-GB CCW optical disk media. **Note:** CCW = continuous composite WORM media. WORM = write-once-read-many.
54 36	ST64ODT	1	EBCDIC	Optical drive type. Valid for subtypes 64 and 65. This field contains blanks for other subtypes. **Value** **Meaning** **L** Optical drive is a library-resident drive. **S** Optical drive is a stand-alone or operator-accessible drive.

Table 77. Format of the subtype data section for subtypes 64–67 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
55 37	ST64OVT	1	EBCDIC	Optical volume type. Valid for subtypes 64 and 65. This field contains blanks for other subtypes. **Value** **Meaning** **B** Optical volume is a backup volume belonging to an OBJECT BACKUP storage group. **G** Optical volume is a grouped volume belonging to an OBJECT storage group. **S** Optical volume is a scratch volume.
56 38	ST64SGN	8	EBCDIC	Storage group name. Valid for subtypes 64 and 65. This field contains blanks for other subtypes.
64 40	ST64LIQT	4	binary	LCS input-work-queue time. The amount of time in milliseconds this request has spent on the LCS input-work-queue waiting to be processed.
68 44	ST64LDQT	4	binary	LCS dispatcher-queued time. The amount of time in milliseconds this request has spent on the LCS dispatcher-queue waiting to be processed.
72 48	ST64LEQT	4	binary	LCS execution-queue time. The amount of time in milliseconds this request has spent on the LCS execution-queue being processed.
76 4C	ST64LTQT	4	binary	LCS library task queue time. The amount of time in milliseconds that this request has spent on the LCS library queue waiting to be processed. Normally, this field represents the cartridge transport mechanism wait time. That is, the time spent waiting for the cartridge transport mechanism within the automated optical disk library to become available. Valid for subtypes 66 and 67.
80 50	ST64LTPT	4	binary	LCS library task processing time. The amount of time in milliseconds that this request took to be processed by the library task. Normally, this field represents the cartridge transport mechanism service time. This is, the time spent by the cartridge transport mechanism within the automated optical disk library performing mechanical motion to move cartridges within the optical disk library. Valid for subtypes 66 and 67.
84 54	ST64RC	4	binary	LCS return code
88 58	ST64RS	4	binary	LCS reason code.
92 5C	ST64FLGS	4	binary	Processing flags. **Bit** **Meaning** **0–31** Reserved

Although subtypes 64–67 share a common subtype data section, not all fields are valid for each of the four subtypes. Table 78 on page 554 identifies which fields in the OAM subtype data section are valid for each of the four subtypes.

Table 78. Valid subtype data section fields for subtypes 64–67				
Field name	**Drive online subtype 64**	**Drive offline subtype 65**	**Library online subtype 66**	**Library offline subtype 67**
ST64OLN	X	X	X	X
ST64OLDT	X	X	X	X
ST64OLDN	X	X	X	X
ST64ODN	X	X		
ST64ODDT	X	X		
ST64ODDN	X	X		
ST64VSN0	See table note 1.	See table note 2.		
ST64VSN1	See table note 1.	See table note 2.		
ST64OMT	See table note 1.	See table note 2.		
ST64ODT	X	X		
ST64OVT	See table note 1.	See table note 2.		
ST64SGN	See table note 1.	See table note 2.		
ST64LIQT				
ST64LDQT	X	X	X	X
ST64LTQT			X	X
ST64LTPT			X	X
ST64RC	X	X	X	X
ST64RS	X	X	X	X
ST64FLGS	X	X	X	X

Note:

1. This field is valid only if there is an optical disk cartridge mounted in the drive at the time the **VARY SMS,DRIVE(***drive_name***),ONLINE** is issued.
2. This field is valid only if there is an optical disk cartridge mounted in the drive at the time the **VARY SMS,DRIVE(***drive_name***),OFFLINE** is issued.

LCS optical cartridge entry, eject, label, audit, mount, and demount (subtypes 68–73)

Table 79 on page 555 describes the format of the subtype data section for the following OAM SMF record subtypes:

68
: LCS optical cartridge entry

69
: LCS optical cartridge eject

70
: LCS optical cartridge label

71
: LCS optical volume audit

72
: LCS optical volume mount

73
LCS optical volume demount

Table 79. Format of the subtype data section for subtypes 68–73

Offsets	Name	Length	Format	Description
0 0	ST68OLN	8	EBCDIC	Optical library name. This field contains the real library name of an operator-accessible drive.
8 8	ST68OLDT	8	EBCDIC	Optical library device type.
16 10	ST68OLDN	4	EBCDIC	MVS device number that corresponds to the optical library.
20 14	ST68ODN	8	EBCDIC	Optical drive name.
28 1C	ST68ODDT	8	EBCDIC	Optical drive device type.
36 24	ST68ODDN	4	EBCDIC	MVS device number that corresponds to the optical drive.
40 28	ST68VSN0	6	EBCDIC	Volume serial number.
46 2E	ST68VSN1	6	EBCDIC	Volume serial number of the opposite side of the optical disk.
52 34	ST68OMT	2	EBCDIC	Optical media type **Value** **Meaning** **01** IBM 3995 5.25-inch 650-MB rewritable optical disk media. **03** IBM 3995 5.25-inch 650-MB WORM optical disk media. **11** IBM 3995 5.25-inch 1300-MB rewritable optical disk media. **13** IBM 3995 5.25-inch 1300-MB WORM optical disk media. **15** IBM 3995 5.25-inch 1300-MB CCW optical disk media. **21** IBM 3995 5.25-inch 2600-MB rewritable optical disk media. **23** IBM 3995 5.25-inch 2600-MB WORM optical disk media. **25** IBM 3995 5.25-inch 2600-MB CCW optical disk media. **31** IBM 3995 5.25-inch 5.2-GB rewritable optical disk media. **33** IBM 3995 5.25-inch 5.2-GB WORM optical disk media. **35** IBM 3995 5.25-inch 5.2-GB CCW optical disk media. **Note:** CCW = continuous composite WORM media. WORM = write-once-read-many.

Table 79. Format of the subtype data section for subtypes 68–73 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
54 36	ST68ODT	1	EBCDIC	Optical drive type: **Value** **Meaning** **L** Optical drive is a library-resident drive. **S** Optical drive is a stand-alone or operator-accessible drive.
55 37	ST68OVT	1	EBCDIC	Optical volume type: **Value** **Meaning** **B** Optical volume is a backup volume belonging to an OBJECT BACKUP storage group. **G** Optical volume is a grouped volume belonging to an OBJECT storage group. **S** Optical volume is a scratch volume.
56 38	ST68SGN	8	EBCDIC	Storage group name.
64 40	ST68LIQT	4	binary	LCS input-work-queue time. The amount of time in milliseconds this request has spent on the LCS input-work-queue waiting to be processed.
68 44	ST68LDQT	4	binary	LCS dispatcher-queue time. The amount of time in milliseconds this request has spent on the LCS dispatcher-queue waiting to be processed.
72 48	ST68LEQT	4	binary	LCS execution-queue time. The amount of time in milliseconds this request has spent on the LCS execution-queue being processed.
76 4C	ST68LTQT	4	binary	LCS library task queue time. The amount of time in milliseconds this request has spent on the LCS library queue waiting to be processed. Normally, this field represents the cartridge transport mechanism wait time. That is, the time spent waiting for the cartridge transport mechanism within the automated optical disk library to become available.
80 50	ST68LTPT	4	binary	LCS library task processing time. The amount of time in milliseconds this request took to be processed by the library task. Normally, this field represents the cartridge transport mechanism service time. That is, the time spent by the cartridge transport mechanism within the automated optical disk library performing mechanical motion to move cartridges within the optical disk library.
84 54	ST68RC	4	binary	LCS return code.
88 58	ST68RS	4	binary	LCS reason code.

Table 79. Format of the subtype data section for subtypes 68–73 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
92 5C	ST68FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. The following bit definitions apply for record subtype 68: **Bit** **Meaning** **0** When on, the volume serial number described by field ST68VSN0 required formatting as part of the optical cartridge entry processing. **1** When on, the volume serial number described by field ST68VSN1 required formatting as part of optical cartridge entry processing. **2** When on, the volume serial number described by field ST68VSN0 was not known to OAM at the time of being entered into the optical library. There was no row for this optical disk volume in the Volume table in the OCDB. **3** When on, the volume serial number described by field ST68VSN1 was not known to OAM at the time of being entered into the optical library. There was no row for this optical disk volume in the Volume table in the OCDB. **4–31** Reserved.
92 5C	ST69FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. The following bit definitions apply for record subtype 69: **Bit** **Meaning** **0** When on, this optical cartridge was automatically ejected by the system due to an error condition known as a system-initiated eject request. **1–31** Reserved.
92 5C	ST70FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. The following bit definitions apply for record subtype 70: **Bit** **Meaning** **0–31** Reserved.
92 5C	ST71FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. The following bit definitions apply for record subtype 71: **Bit** **Meaning** **0–31** Reserved.

Offsets	Name	Length	Format	Description
92 5C	ST72FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. The following bit definitions apply for record subtype 72: **Bit** **Meaning** **0** When on, the volume serial number described by field ST68VSN0 required formatting as part of the optical volume mount processing. **1** When on, the volume serial number described by field ST68VSN1 required formatting as part of the optical volume mount processing. **2–31** Reserved.
92 5C	ST73FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. The following bit definitions apply for record subtype 73: **Bit** **Meaning** **0–31** Reserved.
96 60	ST68TMNT	4	binary	Elapsed time in milliseconds that the optical disk volume was mounted. Valid for subtypes 69 and 73.
100 64	ST68NOW	4	binary	Number of objects written to this optical disk volume while it was mounted. Valid for subtypes 69 and 73.
104 68	ST68NKBW	4	binary	Number of kilobytes of object data written to this optical disk volume while it was mounted. Valid for subtypes 69 and 73.
108 6C	ST68NOR	4	binary	Number of objects read from this optical disk volume while it was mounted. Valid for subtypes 69 and 73.
112 70	ST68NKBR	4	binary	Number of kilobytes of object data read from this optical disk volume while it was mounted. Valid for subtypes 69 and 73.
116 74	ST68NOD	4	binary	Number of objects deleted from this optical disk volume while it was mounted. Valid for subtypes 69 and 73.
120 78	ST68NKBD	4	binary	Number of kilobytes of object data deleted from this optical disk volume while it was mounted. Valid for subtypes 69 and 73.
Note: For subtypes 69 and 73, each object size is rounded up to the next whole KB before being added to the total.				

Table 79. Format of the subtype data section for subtypes 68–73 (continued)

Although subtypes 68–73 share a common subtype data section, not all fields are valid for each of the six subtypes. Table 80 on page 559 identifies which fields in the OAM subtype data section are valid for each of the six subtypes.

Table 80. Valid subtype data section fields for subtypes 68–73						
Field name	**Optical cartridge entry subtype 68**	**Optical cartridge eject subtype 69**	**Optical cartridge label subtype 70**	**Optical volume audit subtype 71**	**Optical volume mount subtype 72**	**Optical volume demount subtype 73**
ST68OLN	X	X	X	X	X	X
ST68OLDT	X	X	X	X	X	X
ST68OLDN	X	X	X	X	X	X
ST68ODN	X	See table note.	X	X	X	X
ST68ODDT	X	See table note.	X	X	X	X
ST68ODDN	X	See table note.	X	X	X	X
ST68VSN0	X	X	X	X	X	X
ST68VSN1	X	X	X	X	X	X
ST68OMT	X	X	X	X	X	X
ST68ODT	X	See table note.	X	X	X	X
ST68OVT	X	X	X	X	X	X
ST68SGN	X	X	X	X	X	X
ST68LIQT		X	X	X		
ST68LDQT	X	X	X	X		
ST668LEQT	X	X	X	X		
ST68LTQT		X		X		
ST68LTPT		X		X		
ST68RC	X	X	X	X	X	X
ST68RS	X	X	X	X	X	X
ST68FLGS	X	X	X	X	X	X
ST68TMNT		See table note.				X
ST68NOW		See table note.				X
ST68NKBW		See table note.				X
ST68NOR		See table note.				X
ST68NKBR		See table note.				X
ST68NOD		See table note.				X
ST68NKBD		See table note.				X
Note: This field contains valid data if the optical disk volume being ejected is mounted in an optical disk drive at the time that the **LIBRARY,EJECT,***volser* command or **F OAM,EJECT,*volser*** command is received to eject the volume.						

LCS optical write, read, logical delete, physical delete (subtypes 74–77)

Table 81 on page 560 describes the format of the subtype data section for the following OAM SMF record subtypes:

74
LCS optical write request

75
: LCS optical read request

76
: LCS optical delete request (logical)

77
: LCS optical delete request (physical)

Table 81. Format of the subtype data section for subtypes 74–77				
Offsets	**Name**	**Length**	**Format**	**Description**
0 0	ST74ORMN	16	EBCDIC	OAM request member name. Valid for subtypes 74 and 75.
16 10	ST74OTMN	16	EBCDIC	OAM target member name. Valid for subtypes 74 and 75.
32 20	ST74OLN	8	EBCDIC	Optical library name.
40 28	ST74OLDT	8	EBCDIC	Optical library device type.
48 30	ST74OLDN	4	EBCDIC	MVS device number that corresponds to the optical library.
52 34	ST74ODN	8	EBCDIC	Optical drive name.
60 3C	ST74ODDT	8	EBCDIC	Optical drive device type.
68 44	ST74ODDN	4	EBCDIC	MVS device number that corresponds to the optical drive.
72 48	ST74ODT	1	EBCDIC	Optical drive type: **Value** **Meaning** **L** Optical drive is a library-resident drive. **S** Optical drive is a stand-alone or operator-accessible drive.
73 49	ST74OVT	1	EBCDIC	Optical volume type **Value** **Meaning** **B** Optical volume is a backup volume belonging to an OBJECT BACKUP storage group. **G** Optical volume is a grouped volume belonging to an OBJECT storage group.
74 4A	ST74SGN	8	EBCDIC	Storage group name.
82 52	ST74LIQT	4	binary	LCS input-work-queue time. The amount of time in milliseconds that this request has spent on the LCS input-work-queue waiting for processing.
86 56	ST74LDQT	4	binary	LCS dispatcher-queue time. The amount of time in milliseconds that this request has spent on the LCS dispatcher-queue waiting for processing.
90 5A	ST74LEQT	4	binary	LCS execution-queue time. The amount of time in milliseconds that this request has spent on the LCS execution-queue processing.

Table 81. Format of the subtype data section for subtypes 74–77 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
94 5E	ST74LXQT	4	binary	XCF cross system processing-queue time. The amount of time in milliseconds that this request spent being processed on the XCF cross system queue. For subtypes 76 and 77, this field contains binary zeros.
98 62	ST74OVMT	4	binary	Optical volume mount time. The amount of time in milliseconds that it took to mount the optical disk volume required by this request. This field is valid if bit 1, 2, or 3 in field ST74FLGS is on. This field is valid for subtypes 74, 75, and 77.
102 66	ST74OVDT	4	binary	Optical volume demount time. This is the amount of time in milliseconds that it took to demount the optical disk volume that was mounted prior to mounting the optical disk volume required by this request. The field is valid if bit 3 in field ST74FLGS is on. This field is valid for subtypes 74, 75, and 77.
106 6A	ST74FLGS	4	binary	Processing flags **Bit** **Meaning** **0** This request was processed using a mounted optical disk volume and did not require an unmounted optical disk volume to be mounted. Valid for subtypes 74, 75, and 77. Not valid for subtype 76. **1** This request was processed using the opposite side of a mounted optical disk volume. Therefore, this request required the optical disk volume to be turned over in order to access the volume on the opposite side of the mounted volume. Valid for subtypes 74, 75, and 77. Not valid for subtype 76. **2** This request required an unmounted optical disk volume to be mounted and the optical disk drive that was used to process this request was empty at the time of the request. Therefore, this request did not require a mounted optical disk volume to be demounted prior to mounting the required optical disk volume. Valid for subtypes 74, 75, and 77. Not valid for subtype 76. **3** This request required an unmounted optical disk volume to be mounted and the optical disk drive that was selected for this request was full. Therefore, this request required a mounted optical disk volume to be demounted prior to mounting the required optical disk volume. Valid for subtypes 74, 75, and 77. Not valid for subtype 76. **4–31** Reserved.
110 6E	ST74NOBJ	4	binary	Total number of objects in this request. The maximum possible for this field is 280. With 280 object entries, the maximum SMF record size is 32 744 bytes.
114 72	ST74NKBP	4	binary	Total number of kilobytes of object data in this request.
118 76	ST74SOBJ	4	binary	Total number of objects in this request that processed successfully.

Table 81. Format of the subtype data section for subtypes 74–77 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
122 7A	ST74SKBP	4	binary	Total number of kilobytes of object data in this request that processed successfully.
126 7E	*	14	binary	Reserved.
Note: The following fields (comprising 116 bytes) are repeated for each object in the chained request, that is for the number of objects specified in the ST74NOBJ field. The maximum number of times that the following fields will be repeated is 280. With 280 object entries, the maximum SMF record size is 32 744 bytes.				
140 8C	ST74COLN	44	EBCDIC	Collection name.
184 B8	ST74OBJN	44	EBCDIC	Object name.
228 E4	ST74OLEN	4	binary	Object length.
232 E8	ST74OOFF	4	binary	Object offset. Valid for a subtype 75 partial object read.
236 EC	ST74VSN	6	EBCDIC	Volume serial number.
242 F2	ST74OMT	2	EBCDIC	Optical media type: **Value** **Meaning** **01** IBM 3995 5.25-inch 650-MB rewritable optical disk media. **03** IBM 3995 5.25-inch 650-MB WORM optical disk media. **11** IBM 3995 5.25-inch 1300-MB rewritable optical disk media. **13** IBM 3995 5.25-inch 1300-MB WORM optical disk media. **15** IBM 3995 5.25-inch 1300-MB CCW optical disk media. **21** IBM 3995 5.25-inch 2600-MB rewritable optical disk media. **23** IBM 3995 5.25-inch 2600-MB WORM optical disk media. **25** IBM 3995 5.25-inch 2600-MB CCW optical disk media. **31** IBM 3995 5.25-inch 5.2-GB rewritable optical disk media. **33** IBM 3995 5.25-inch 5.2-GB WORM optical disk media. **35** IBM 3995 5.25-inch 5.2-GB CCW optical disk media. **Note:** CCW = continuous composite WORM media. WORM = write-once-read-many.
244 F4	ST74OTKN	4	binary	Object volume location token.
248 F8	ST74RC	4	binary	LCS return code.
252 FC	ST74RS	4	binary	LCS reason code.

Although subtypes 74–77 share a common subtype data section, not all fields are valid for each of the four subtypes. Table 82 on page 563 identifies which fields in the OAM subtype data section are valid for each of the four subtypes.

Table 82. Valid subtype data section fields for subtypes 74–77

Field name	**Optical write request subtype 74**	**Optical read request subtype 75**	**Optical delete request (logical) subtype 76**	**Optical delete request (physical) subtype 77**
ST74ORMN	X	X		
ST74OTMN	X	X		
ST74OLN	X	X	X	X
ST74OLDT	X	X	X	X
ST74OLDN	X	X	X	X
ST74ODN	X	X		X
ST74ODDT	X	X		X
ST74ODDN	X	X		X
ST74ODT	X	X		X
ST74OVT	X	X	X	X
ST74SGN	X	X	X	X
ST74LIQT	X	X	X	
ST74LDQT	X	X		
ST74LEQT	X	X		X
ST74LXQT	X	X		
ST74LVMT	X	X		X
ST74LVDT	X	X		X
ST74NOBJ	X	X	X	X
ST74NKBP	X	X	X	X
ST74FLGS	X	X	X	X
ST74SOBJ	X	X	X	X
ST74SKBP	X	X	X	X
ST74COLN	X	X	X	X
ST74OBJN	X	X	X	X
ST74OLEN	X	X	X	X
ST74OOFF		X		
ST74VSN	X	X	X	X
ST74OMT	X	X	X	X
ST74OTKN	X	X	X	X
ST74RC	X	X	X	X
ST74RS	X	X	X	X

LCS tape write and read request (subtypes 78–79, and 88)

Table 83 on page 564 describes the format of the subtype data section for the following OAM SMF record subtypes:

78
: LCS tape write request

79
: LCS tape read request

88
: LCS object tape logical delete request

Table 83. Format of subtype data section for subtypes 78–79				
Offsets	**Name**	**Length**	**Format**	**Description**
0 0	ST78ORMN	16	EBCDIC	OAM request member name. Valid for subtype 79 only.
16 10	ST78OTMN	16	EBCDIC	OAM target member name. Valid for subtype 79 only.
32 20	ST78TDUN	8	EBCDIC	Tape drive unit name.
40 28	ST78TDDN	4	EBCDIC	MVS device number of the tape drive.
44 2C	ST78TVT	1	EBCDIC	OAM tape volume type. **Value** **Meaning** **B** Tape volume is a backup volume belonging to an OBJECT BACKUP storage group. **G** Tape volume is a grouped volume belonging to an OBJECT storage group.
45 2D	*	3	binary	Reserved.
48 30	ST78SGN	8	EBCDIC	Name of the OBJECT or OBJECT BACKUP storage group to which the tape volume belongs.
56 38	ST78LIQT	4	binary	LCS input-work-queue time. The amount of time in milliseconds this request has spent on the LCS input-work-queue waiting to be processed. For subtype 88, this field contains binary zeros.
60 3C	ST78LDQT	4	binary	LCS dispatcher-queue time. The amount of time in milliseconds this request has spent on the LCS dispatcher-queue waiting to be processed. For subtype 88, this field contains binary zeros.
64 40	ST78LEQT	4	binary	LCS execution-queue time. The amount of time in milliseconds that this request has spent on the LCS execution-queue being processed. For subtype 88, this field contains binary zeros.
68 44	ST78LXQT	4	binary	XCF cross system processing-queue time. The amount of time in milliseconds that this request has spent being processed on the XCF cross system queue. For subtypes 78 and 88, this field contains binary zeros.

Table 83. Format of subtype data section for subtypes 78–79 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
72 48	ST78LMAT	4	binary	MVS dynamic allocation time. This is the amount of time in milliseconds that was required by MVS dynamic allocation (SVC 99) to dynamically allocate the tape drive. For subtypes 78 and 79, this field is valid only if bit 1 in field ST78FLGS is on. For subtype 88, this field contains binary zeros.
76 4C	ST78LMDT	4	binary	MVS dynamic deallocation time. This is the amount of time in milliseconds that was required by MVS dynamic deallocation (SVC99) to dynamically deallocate the tape drive. For subtypes 78 and 79, this field is valid only if bit 2 in field ST78FLGS is on. For subtype 88, this field contains binary zeros.
80 50	ST78LDCT	4	binary	DFP CLOSE time. This is the amount of time in milliseconds that was required by DFP CLOSE processing to close an already-opened tape data set. For subtypes 78 and 79, this field is valid only if bit 2 in field ST78FLGS is on. For subtype 88, this field contains binary zeros.
84 54	ST78LDOT	4	binary	DFP OPEN time. This is the amount of time in milliseconds that was required by DFP OPEN processing to open the tape data set. For subtypes 78 and 79, this field is valid only if bit 1 or 2 in field ST78FLGS is on. For subtype 88, this field contains binary zeros.
88 58	ST78LDPT	4	binary	DFP POINT time. This is the amount of time in milliseconds that was required by DFP POINT processing to position to the correct block-ID on the tape media. For subtype 88, this field contains binary zeros.
92 5C	ST78LBRT	4	binary	BSAM READ time. This is the amount of time in milliseconds that OAM spent in BSAM READ processing reading data from the tape volume. Valid for subtype 79.
96 60	ST78LBWT	4	binary	BSAM WRITE time. This is the amount of time in milliseconds that OAM spent in BSAM WRITE processing writing data to the tape volume. Valid for subtype 78.
100 64	ST78LBCT	4	binary	BSAM CHECK time. This is the amount of time in milliseconds that OAM spent in BSAM CHECK processing waiting for I/O operations to the tape volume to complete. Valid for subtype 78 and 79.

Table 83. Format of subtype data section for subtypes 78–79 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
104 68	ST78FLGS	4	binary	Processing flags. For subtype 88, this field contains binary zeros. **Bit** **Meaning** **0** This request was processed using a mounted tape volume and did not require an unmounted tape volume to be mounted. **1** This request required an unmounted tape volume to be mounted and the tape drive that was used to process this request was empty at the start of processing this request. Therefore, this request did not require a mounted tape volume to be demounted prior to mounting the required tape volume. **2** This request required an unmounted tape volume to be mounted and the tape drive that was used to process this request was full at the start of processing this request. Therefore, this request required a mounted tape volume to be demounted prior to mounting the required tape volume. **3** This request was processed using a tape drive inside an automated tape library dataserver. **4** This request was processed using a tape volume associated with TAPE SUBLEVEL 1. **5** This request was processed using a tape volume associated with TAPE SUBLEVEL 2. **6–31** Reserved.
108 6C	ST78NOBJ	4	binary	Total number of objects in this request. The maximum value for this field is 280. With 280 object entries in this record, the maximum SMF record size for subtype 78 is 32 744 bytes.
112 70	ST78NKBP	4	binary	Total number of kilobytes of object data in this request.
116 74	ST78SOBJ	4	binary	Total number of objects in this request that processed successfully.
120 78	ST78SKBP	4	binary	Total number of kilobytes of object data in this request that processed successfully.
124 7C	*	16	binary	Reserved.
Note: The following fields (comprising 116 bytes) are repeated for each object in the chained request, that is for the number of objects specified in the ST78NOBJ field. The maximum number of times that the following fields will be repeated is 280. With 280 object entries, the maximum SMF record size for subtype 78 is 32 744 bytes.				
140 8C	ST78COLN	44	EBCDIC	Collection name.
184 B8	ST78OBJN	44	EBCDIC	Object name.
228 E4	ST78OLEN	4	binary	Object length.
232 E8	ST78OOFF	4	binary	Object offset. Valid for a subtype 79 partial object read.

Table 83. Format of subtype data section for subtypes 78–79 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
236 EC	ST78VSN	6	EBCDIC	Volume serial number.
242 F2	ST78TMT	2	EBCDIC	Tape media type: **Value** **Meaning** **02** IBM 3480 Cartridge System Tape. **04** IBM 3480 Enhanced Capacity Cartridge System Tape. **05** IBM High Performance Cartridge Tape. **06** IBM Extended High Performance Cartridge Tape. **07** IBM Enterprise Tape Cartridge. **08** IBM Enterprise WORM Tape Cartridge. **09** IBM Enterprise Economy Tape Cartridge. **10** IBM Enterprise Economy WORM Tape Cartridge. **12** IBM Enterprise Extended Tape Cartridge. **14** IBM Enterprise Extended WORM Tape Cartridge.
244 F4	ST78OTKN	4	binary	Object volume location token.
248 F8	ST78RC	4	binary	LCS return code.
252 FC	ST78RS	4	binary	LCS reason code.

OAM tape volume demount (subtype 87)

Table 84 on page 567 describes the format of the subtype 87 data section.

Table 84. Format of subtype data section for subtype 87				
Offsets	**Name**	**Length**	**Format**	**Description**
0 0	ST87TDDN	4	EBCDIC	MVS device number that corresponds to the tape drive on which the volume was mounted.
4 4	ST87TDDT	4	EBCDIC	MVS UCB device type associated with the tape drive on which the volume was mounted.
8 8	ST87TVUN	8	EBCDIC	Unit name associated with the tape volume and used to allocate the tape drive.
16 10	ST87VSN	6	EBCDIC	Volume serial number of the tape volume.

Table 84. Format of subtype data section for subtype 87 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
22 16	ST87TMT	2	EBCDIC	Tape media type. **Value** **Meaning** **02** IBM 3480 Cartridge System Tape. **04** IBM 3480 Enhanced Capacity Cartridge System Tape. **05** IBM High Performance Cartridge Tape. **06** IBM Extended High Performance Cartridge Tape. **07** IBM Enterprise Tape Cartridge **08** IBM Enterprise WORM Tape Cartridge **09** IBM Enterprise Economy Tape Cartridge **10** IBM Enterprise Economy WORM Tape Cartridge **12** IBM Enterprise Extended Tape Cartridge **14** IBM Enterprise Extended WORM Tape Cartridge
24 18	ST87TVT	1	EBCDIC	OAM tape volume type. **Value** **Meaning** **B** Tape volume is a backup volume belonging to an OBJECT BACKUP storage group. **G** Tape volume is a grouped volume belonging to an OBJECT storage group.
25 19	*	3	binary	Reserved.
28 1C	ST87SGN	8	EBCDIC	Name of the OBJECT or OBJECT BACKUP storage group.
36 24	ST87RC	4	binary	LCS return code.
40 28	ST87RS	4	binary	LCS reason code.
44 2C	ST87FLGS	4	binary	Processing flags. **Bit** **Meaning** **0** This request was processed using a tape volume associated with TAPE SUBLEVEL 1. **1** This request was processed using a tape volume associated with TAPE SUBLEVEL 2. **2–31** Reserved.

Table 84. Format of subtype data section for subtype 87 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
48 30	ST87TMNT	4	binary	Elapsed time in milliseconds that the tape volume was mounted, measured from the time that the first DFP OPEN macro completed to the time the tape volume was deallocated by an SVC 99 dynamic deallocation request.
52 34	ST87NOW	4	binary	Number of objects written to this tape volume while it was mounted.
56 38	ST87NKBW	4	binary	Number of logical kilobytes of object data written to this tape volume while it was mounted. X'FFFFFFFF' indicates the counter has overflowed.
60 3C	ST87NOR	4	binary	Number of objects read from this tape volume while it was mounted.
64 40	ST87NKBR	4	binary	Number of kilobytes of object data read from this tape volume while it was mounted. X'FFFFFFFF' indicates the counter has overflowed.
68 44	ST87NBW	8	binary	Number of logical bytes of object data written to this tape volume while it was mounted.
76 4C	ST87NBR	8	binary	Number of bytes of object data read from this tape volume while it was mounted
Note: For subtype 87, the total value in fields that refer to "number of kilobytes" or "number of bytes" will be rounded up.				

LCS file system write, read, delete (subtypes 90-93)

Table 85 on page 569 describes the format of the subtype data section for the following OAM SMF record subtypes:

90
LCS File System Write Request

91
LCS File System Read Request

92
LCS File System Physical Delete Request

93
LCS File System Physical Delete Request (Uncommitted Application Store Cleanup)

Table 85. Format of subtype data section for subtypes 90-93				
Offsets	**Name**	**Length**	**Format**	**Description**
0 0	ST90DIR	30	EBCDIC	File system directory.
30 1E	ST90SGN	8	EBCDIC	OBJECT storage group name.
38 26	ST90COLN	44	EBCDIC	Collection name.
82 52	ST90OBJN	44	EBCDIC	Object name.

Table 85. Format of subtype data section for subtypes 90-93 (continued)				
Offsets	**Name**	**Length**	**Format**	**Description**
126 7E	ST90FST	2	EBCDIC	File system type. Only valid for subtypes 90 and 91: **Value** **Meaning** **00** ZFS **01** NFS
128 80	ST90INST	4	binary	Instance ID.
132 84	ST90FLGS	4	binary	Processing Flags. **Bit** **Meaning** **0** When on, request originated within OAM address space. **1** When on, request originated outside OAM address space.
2-31	Reserved			
136 88	ST90OLEN	4	binary	Object length. Only valid for subtypes 90 and 91.
140 8C	ST90OOFF	4	binary	Object offset. Valid for a subtype 91 partial object read.
144 90	ST90LIQT	4	binary	LCS input-work-queue time. The amount of time in milliseconds that this request has spent on the LCS inputwork- queue waiting for processing. Only valid for subtypes 90 and 91.
148 94	ST90LDQT	4	binary	LCS dispatcher-queue time. The amount of time in milliseconds that this request has spent on the LCS dispatcher-queue waiting for processing. Only valid for subtypes 90 and 91.
152 98	ST90LEQT	4	binary	LCS execution-queue time. The amount of time in milliseconds that this request has spent on the LCS execution-queue processing. Only valid for subtypes 90 and 91.
156 9C	ST90RC	4	binary	LCS return code.
160 A0	ST90RS	4	binary	LCS reason code.

Invoking the SMF PARMLIB member

The MVS operator can dynamically change which SMF records and record subtypes are being recorded by one of two methods:

- Issue a **SET SMF** = *xx* command at an MVS console to activate a new SMF PARMLIB member. The *xx* identifies the **SMFPRM***xx* member of the SYS1.PARMLIB that is to be activated by SMF.

- Issue a **SETSMF** command at an MVS system console to add a SUBPARM parameter or to replace any previously specified parameter in the active SMF PARMLIB member except for the ACTIVE, PROMPT, SID, or EXITS parameters.

Related reading: For more information regarding the SET SMF, SETSMF, and the SMF PARMLIB member, see the following documents:

- *z/OS MVS System Management Facilities (SMF)*
- *z/OS MVS System Commands*
- *z/OS MVS Initialization and Tuning Reference*

Changing SMF recording

The MVS system operator or MVS system programmer can dynamically change OAM SMF recording using one of the following two methods:

- Update the SMF PARMLIB member (SMFPRM*xx*) to include the OAM SMF record subtypes:

  SYS(TYPE(85(2:3)))

  and activate the SMF PARMLIB member (SMFPRM*xx*) by entering the following MVS operator SET command:

  SET SMF=*xx*

  Note: The above example activates the OAM SMF recording for subtypes 2 and 3.
- Update the SMF options dynamically by entering the following MVS operator SETSMF command:

  SETSMF SYS(TYPE(85(4:6)))

  Note: The above example activates the OAM SMF recording for subtypes 4, 5, and 6.

Below are several examples of the format of the SETSMF command to activate various OAM SMF record subtypes from an MVS console:

- To exclude collecting all OAM SMF records, enter the following command:

  SETSMF SYS(NOTYPE(85)))
- To activate all OAM SMF record subtypes for the OSREQ macro application programming interface (subtypes 1–10), enter the following command:

  SETSMF SYS(TYPE(85(1:10)))
- To activate the OAM SMF record subtype for the OSMC storage group processing (subtype 32), enter the following command:

  SETSMF SYS(TYPE(85(32)))
- To activate the OAM SMF record subtype for the OSREQ RETRIEVE (subtype 3), LCS optical volume mount (subtype 72), and LCS optical read request (subtype 75), enter the following command:

  SETSMF SYS(TYPE(85(3,72,75)))
- To activate the OAM SMF record subtypes to track all optical library subsystem activity: optical cartridge entry (subtype 68), optical cartridge eject (subtype 69), optical volume audit (subtype 71), optical volume mount (subtype 72) and optical volume demount (subtype 73), enter the following command:

  SETSMF SYS(TYPE(85(68,69,71,72,73)))
- To activate the OAM SMF record subtypes to track all objects being retrieved from tape: OSREQ RETRIEVE (subtype 3) and LCS tape read requests (subtype 79), enter the following command:

  SETSMF SYS(TYPE(85(3,79)))

Related reading: For more information regarding the SET SMF and SETSMF commands, see *z/OS MVS System Commands*.

DASD space allocation

The number of OAM SMF records written to the SMF data sets is dependent on two major factors:

- The amount of OAM activity that occurs on the processor complex, and
- The OAM SMF record subtypes that the system programmer has selected to be recorded in the SMF data sets.

Depending on the number of SMF records being recorded in the SMF data sets, the system programmer should perform the following activities:

- Determine which OAM SMF record subtypes should be captured.
- Make a preliminary determination of the number of each OAM SMF record subtype that will occur each hour or day.
- Calculate the additional DASD space requirements needed for SMF data sets based on the above two factors. The OAM SMF record subtype sizes are provided in the Table 32 on page 215. Variables and formulas for determining DASD requirements can be found in Table 8 on page 73, Table 9 on page 73, and Table 10 on page 74.
- Determine if the DASD space allocation quantities for the existing SMF data sets will be satisfactory given the additional space required by the OAM SMF records. If the DASD space allocation quantity for the existing SMF data sets is insufficient based on how frequently the system programmer wishes SMF to switch data sets, perform one or both of the following:
  - Reallocate the SMF data set with a larger primary space allocation quantity
  - Increase the number of SMF data sets
- Determine the adequacy of the DASD subsystem containing the SMF data sets to determine if the additional I/O activity caused by OAM recording the selected SMF records is going to introduce unacceptable levels of utilization and I/O contention on the DASD subsystem components, including:
  - DASD device level I/O contention
  - Control unit I/O contention
  - Channel path contention and utilization

Appendix F. Auto-delete installation exit

This appendix contains product-sensitive programming interface and associated guidance information that describes how to tailor the OAM auto-delete installation exit to suit your needs.

Related reading: See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* for information on the following tape library-related exits:

- Cartridge Entry Installation Exit (CBRUXENT)
- Change Use Attribute Installation Exit (CBRUXCUA)
- Cartridge Eject Installation Exit (CBRUXEJC)
- Volume Not in Library Installation Exit (CBRUXVNL)

Auto-delete installation exit (CBRHADUX)

You can use the auto-delete installation exit (CBRHADUX) to confirm or bypass automatic deletion of objects during an OSMC management cycle in your OAM system.

This installation exit executes as part of the OAM Storage Management Component (OSMC) mainline processing. OSMC can delete an object when its lifetime expires in accordance with the definition of the management class assigned to the object. An object can also expire through an explicit expiration date. If the object has an explicit expiration date, that takes precedence over the defined management class for the object. Before any object is deleted, OSMC calls the auto-delete installation exit to approve or deny the request for object deletion. This approval or disapproval for object deletion is dependent upon the return code returned by the installation exit. The installation exit also records the deletion of the object so that other applications are kept in synchronization with the OAM directory.

Exception: The installation exit is not called when a user requests deletion of an object during an OSREQ macro call.

OSMC performs the storage management cycle using a separate task for each storage group. The auto-delete installation exit can execute concurrently; therefore, it must be reentrant. A *reentrant* program is serially reusable. Each time the user enters the program, a fresh copy of working storage is provided. If any values must be saved, the user must save them in other storage areas or files.

The installation exit is called for every object that is to be deleted by OSMC and one last time when there are no more objects to delete. This last call does not include object deletion information.

Related reading: See *z/OS DFSMS OAM Application Programmer's Reference* for information on how to use the OSREQ macro in your application.

Installing and replacing the CBRHADUX installation exit

You can use and modify the sample auto-delete installation exit that is provided in SAMPLIB (see “Sample auto-delete installation exit” on page 577), or you can write a new installation exit. If you modify or write your own auto-delete installation exit, you must reassemble the data set, and link-edit the control section (CSECT) into SYS1.LINKLIB. The name of the load module must be CBRHADUX.

In an OAMplex, when a new CBRHADUX load module is installed, make sure that changes are consistent on all OAMs within the OAMplex.

Writing the CBRHADUX exit

In general, the routines you code for the auto-delete installation exit should follow these criteria:

- Written in Assembler H or High Level Assembler

- Handle multiple requests (reentrant)
- Reside in SYS1.LINKLIB
- Include any valid combination of AMODE and RMODE
- Return to the caller using the BSM instruction

The sample auto-delete installation exit has a "Sample auto-delete installation exit" on page 577 that prevents objects from being deleted. It also contains code to allow automatic deletion of objects which can be activated with a simple code modification. Once the three lines of code are commented out, the sample auto-delete installation exit reads the VERIFY data set. This is specified at the label, VRFYDSN and needs to be modified based upon your installation. The VERIFY data set should contain objects, along with their collections, that should never be deleted. Try to minimize the number of records in this file because this module is called frequently. From this file the user exit builds an internal table used for each subsequent call. When an object and collection are matched a return code of X'4' and a reason code of X'2', which signifies this object should not be expired.

Attention: Take care to avoid processing overhead because it can affect the time it takes to process the OSMC storage management cycle. For example, when the two lines of bypass code are removed, the sample exit, "Sample auto-delete installation exit" on page 577, reads the names of the objects once per cycle. It maintains the verify table to avoid I/O for each object, and it issues I/O to a data set instead of a teleprocessing link to allow for notification of deleted objects.

If the object name is *not* found in the verification table, the exit approves deletion. When the object is deleted, the name is written to a sequential data set called the notify data set, HLQ.XXXXXXXX.OBJECT.DELETE.NOTIFY. Fully qualified data set names contain both the object name and the collection name, allowing uniqueness across OSMC and concurrent I/O from the different tasks. A concatenation of these data sets (one for each group) provides input to other applications that need to synchronize their directories with the OAM object directories.

Input

The sample uses the auto-delete installation field (ADUUFLD) to store a pointer to a dynamic area. Dynamic areas contain: save areas, system services parameter lists, WTO macro definitions, and data control blocks (DCBs) that require updating.

The input data set HLQ.OBJECT.DELETE.VERIFY, known as the VERIFY data set, is defined in the DCBI in the dynamic working storage and also defined in local storage in the DCB with the DDNAME of CBRADUXI.

The verify data set format:

```
          ORGANIZATION:   PS
          RECORD FORMAT:  FB
          RECORD LENGTH:  89
          BLOCK SIZE:     8900
```

The verify data set record format is as follows:

```
          0-43  :   Object Name       CHAR(44)
          44    :   Blank             CHAR(1)
          45-89 :   Collection Name   CHAR(44)
```

Output

The output data set HLQ.XXXXXXXX.OBJECT.DELETE.NOTIFY, known as the NOTIFY data set, is defined in the DCBO in the dynamic working storage and also defined in local storage in the DCB with the DDNAME of CBRADUXO.

The notify data set format:

```
          ORGANIZATION:   PS
          RECORD FORMAT:  FB
          RECORD LENGTH:  89
          BLOCK SIZE:     8900
```

The notify data set record format is as follows:

```
0-43  :   Object Name        CHAR(44)
44    :   Blank              CHAR(1)
45-89 :   Collection Name    CHAR(44)
```

Coding restrictions

The sample executes in 24-bit addressing mode. It can execute in 31-bit mode if you modify it to remove 24-bit dependencies. Input to the routine is always addressable in 24-bit mode. The following are 24-bit dependencies:

- The first GETMAIN should have LOC=BELOW because it has DCBs and OPEN and CLOSE short lists. The OPEN and CLOSE parameter lists can be in storage above the line if MODE=31 is coded on the list and execute forms. LOC=ABOVE can be added to other GETMAINs.
- SYNAD and EODAD must reside below the line by having RMODE 24 for the CSECTs or there must be a DCBE with SYNAD and EODAD.
- Optionally add RMODE 31=BUFF to a DCBE to get QSAM buffers above the line. If so, you can delete the FREEPOOL macro.

Registers on entry to the auto-delete exit routine

The following information is found in the registers on entry to the auto-delete exit routine.

Register
  Contents

0
  Unpredictable, must be saved and restored

1
  Address of the auto-delete parameter list (mapped by macro CBRADUP)

2-12
  Unpredictable, must be saved and restored

13
  Address of a standard 18-word save area

14
  Addressing mode and return address

15
  Address of the auto-delete installation exit

Auto-delete installation exit parameter list

The auto-delete installation exit parameter list contains a user field (ADUUFLD). This field is binary zero on the first call to the installation exit. It is not changed by the system on any subsequent calls; therefore, the auto-delete installation exit can use it to save pertinent information. For example, the reentrant requirement makes it necessary to obtain (GETMAIN) virtual storage for save areas and temporary values. If your exit needs a dynamic work area, you can use GETMAIN and store the address of the work area in ADUUFLD. This allows the exit to use the same storage area on subsequent calls in the cycle. Function code 2 (ADUDONE) is placed in ADUFUNC on the last call in the cycle. Your exit should check ADUFUNC and at end-of-cycle free up the dynamic work area space and any other resources it used.

Register 1 contains the address of the input parameter list for the auto-delete installation exit. The CBRADUP macro maps the parameter list, which contains the fields in Table 86 on page 575.

Table 86. Auto-delete parameter list, CBRADUP

Offset	Type	Length or bit pattern	Name	Description
00 (X'00')	CHARACTER	264	CBRADUP	Class selection parm list

Table 86. Auto-delete parameter list, CBRADUP (continued)

Offset	Type	Length or bit pattern	Name	Description
00 (X'00')	CHARACTER	16	ADUHDR	Header section
00 (X'00')	CHARACTER	4	ADUID	Block ID 'ADU'
	CHARACTER ADU		ADUIDV	Control block ID
04 (X'04')	FIXED	4	ADULEN	Length of parameter list
08 (X'08')	FIXED	1	ADUVER	Version number
	DECIMAL 1		ADUVERV	Version number
09 (X'09')	FIXED	1	ADUREV	Revision number
	DECIMAL 0		ADUREVV	Revision number
10 (X'0A')	FIXED	1	ADUSP	Subpool number
	DECIMAL 0		ADUSPV	Subpool number
11 (X'0B')	FIXED	1	*	Reserved
12 (X'0C')	FIXED	4	*	Reserved
16 (X'10')	CHARACTER	8	ADULINK	Queue linkage section
16 (X'10')	ADDRESS	4	*	Reserved for coexistence must contain zero
20 (X'14')	ADDRESS	4	*	Reserved for coexistence. Must contain zero
24 (X'18')	FIXED	1	ADUFUNC	Function code
	DECIMAL 1		ADUNOTFY	Notify delete call
	DECIMAL 2		ADUDONE	End of auto delete cycle call
25 (X'19')	BITSTRING	3	*	Reserved
28 (X'1C')	CHARACTER	44	ADUONAME	The object name
72 (X'48')	CHARACTER	1	ADUBLANK	Separator blank
73 (X'49')	CHARACTER	44	ADUCLNAM	Collection name
117 (X'75')	CHARACTER	10	ADUCDAT	Object creation date
127 (X'7F')	CHARACTER	1	*	Reserved
128 (X'80')	DECIMAL	2	ADUSCLEN	Length of storage class name
130 (X'82')	CHARACTER	30	ADUSCNAM	Name of the storage class to which object belongs
160 (X'A0')	DECIMAL	2	ADUMCLEN	Length of management class name
162 (X'A2')	CHARACTER	30	ADUMCNAM	Name of the management class to which object belongs
192 (X'C0')	DECIMAL	4	ADUSGLEN	Length of storage group name
196 (X'C4')	CHARACTER	8	ADUSGNAM	Name of the storage group to which object belongs
204 (X'CC')	CHARACTER	8	ADUSGDB2	DB2 storage group name/high level qualifier
212 (X'D4')	CHARACTER	4	ADUDSSID	Name of the DB2 SSID - In a classic OAM configuration, this is what was specified in IGDSMSxx in PARMLIB for DB2SSID(x). - In a multiple OAM configuration, this is the value that is specified for D= when the OAM address space was started.
216 (X'D8')	CHARACTER	32	ADUUFLD	User field
248 (X'F8')	CHARACTER	16	*	Reserved

Registers on return from the CBRHADUX installation exit

The primary output from the installation exit is a return code in register 15 upon return to OSMC (see return code definitions listed in the input parameter list).

Other output includes information necessary to notify other applications when objects are deleted. Table 87 on page 577 describes the return codes returned from the auto-delete Installation exit.

Table 87. Auto-delete return codes, CBRADUP		
Return code	**Name**	**Description**
DECIMAL 0	ADUDELOK	Exit OKs object deletion.
DECIMAL 4	ADUNODEL	Exit rejects object deletion.
DECIMAL 8	ADUFAILC	Exit fails and should not be called again for this cycle deletions continue.
DECIMAL 12	ADUFAILN	Exit fails and should not be called again for this cycle. Deletions do not continue.

CBRHADUX return code and description

X'00'
Allows for deletion of the object to continue and to continue calling the user exit.

X'04'
Do not allow deletion of this object for this instance of OSMC for this storage group.

X'08'
Delete all objects in this storage group for this instance of OSMC but do not call the user exit again.

X'12'
Do not allow for deletion of objects for this OSMC instance and this storage group and call the user exit again.

CBRHADUX messages

The CBRHADUX has implemented WTO messages for diagnostic purposes to ease the implementation of this exit. A WTO has been added to the default bypass code:

```
CBRHADUX USER EXIT: RC=12 OSMC bypassed object expiration
```

This message identifies that the three lines of bypass code have not been either commented or removed from the exit. This message will be displayed for each storage group that is ran during OSMC.

```
CBRHADUX USER EXIT: SG(xxxxxxxx) completed RC=YYYYYYYY RSN=ZZZZZZZZ
```

The subroutine ADUXOUT has been implemented to display errors that have occurred in the CBRHADUX. As shown in the message above, the storage group, return code and reason code are included for diagnostic purposes. Explanation of the return and reason codes can be found in the comment section REASON CODES FOR DIAGNOSTIC REASONS in the sample CBRHADUX. By default the user exit will only display a message for errors with a return code of 8 or higher but this can be modified by altering the RETMIN DC statement.

Sample auto-delete installation exit

SYS1.SAMPLIB member CBRHADUX, as shown here, is the sample OSMC auto-delete installation exit.

```
**********************************************************************
*                                                                    *
* $MOD(CBRHADUX),COMP(OSMC),PROD(OAM) :                              *
*                                                                    *
* MODULE NAME:  CBRHADUX                                             *
*                                                                    *
* DESCRIPTIVE NAME:  OSMC Sample AUTO-DELETE Installation Exit       *
*                                                                    *
**********************************************************************
*PROPRIETARY V3 STATEMENT                                            *
*Licensed Material - Property of IBM                                 *
*"Restricted Materials of IBM"
*5650-ZOS                                                            *
*COPYRIGHT IBM CORP. 1989, 2013                                      *
*END PROPRIETARY V3 STATEMENT                                        *
**********************************************************************
*                                                                    *
*  NOTE: Read before implementing, this exit is used during OSMC     *
```

```
*        and does not perform an OSREQ functions during its           *
*        processing.                                                  *
*                                                                     *
*  This sample program, if installed as is, will prevent objects     *
*  from being deleted/                                                *
*                                                                     *
*  RETURNS:       12 - Do noy delete any objects within this storage *
*                      group and do not call this exit again for this*
*                      instance of OSMC for this storage group.      *
*                                                                     *
*  This sample exit is provided as an example of different coding    *
*  techniques that may be used.  It should be modified to suit the   *
*  specific needs of each user.  The minimum modification that will  *
*  enable any of the function provided is to delete or comment out   *
*  the three lines of "bypass" code that return the "12" described   *
*  above.  (right after "-- THREE LINES OF BYPASS CODE --"  )        *
*                                                                     *
*  SUBROUTINE MCVERIFY is provided as an example of a technique that *
*  could be used to check management class, but is not currently     *
*  invoked.  This technique involves coding the management class     *
*  names in the data storage area for the CBRHADUX.                  *
*                                                                     *
*  Search for keyword MMM to find the Minimum Mandatory              *
*  Modification/Review Points in the code.                           *
*                                                                     *
***********************************************************************
*  FUNCTION:                                                          *
*    Module CBRHADUX is used to verify whether or not an object      *
*    should be deleted.  This module is called during OSMC           *
*    processing when an object has been selected for deletion.       *
*                                                                     *
*  OPERATION:                                                         *
*  - Perform standard entry linkage.                                  *
*  - Set CBRHADUX return code to 12 indicating to OSMC that objects  *
*    are not to be deleted for this storage group and that the       *
*    CBRHADUX exit is not to be called again for this storage group  *
*    ]default behavior(                                               *
*    (Read NOTE: READ BEFORE IMPLEMENTING EXIT for explanation of    *
*     default behavior)                                               *
*  - ]NOTE( Write message to the operator stating that objects       *
*     will NOT be expired for this storage group ]default behavior(  *
*  - If return code not set to 12 then                                *
*     | If this is the first call then                                *
*     | | - Initialize storage                                        *
*     | | - Set up Data Area to allow for reentrancy.                 *
*     | | - Prepare Data Area for next call for same Storage Group   *
*     | | - Establish register save area in Data Area.                *
*     | Endif                                                         *
*     | If current Storage Group is GROUPXX ]see note in code(       *
*     | | - Set CBRHADUX return code to 8 indicating to OSMC that    *
*     | | - objects are to be deleted for this storage group and that*
*     | |   the CBRHADUX exit is not to be called again for this     *
*     | |   storage group.                                            *
*     | |   (as an example of bypassing processing by the CBRHADUX   *
*     | |    exit for a given storage group)                          *
*     | Else                                                          *
*     | | If this invocation is not the end of a storage group       *
*     | | (indicated by the ADUDONE flag)                             *
*     | | | If this is the first invocation (i.e. verify table has   *
*     | | | not yet been built) then                                  *
*     | | | | - Open input dataset and build a local table in storage*
*     | | | |  if needed                                              *
*     | | | | - Open output dataset                                   *
*     | | | Endif                                                     *
*     | | | If no errors encountered so far then                      *
*     | | | | If object name and collection name of current object   *
*     | | | |  do not exist in the in the Verify Table Then          *
*     | | | | | Add a record to the notify dataset                    *
*     | | | | Endif                                                   *
*     | | | Endif                                                     *
*     | | Endif                                                       *
*     | Endif                                                         *
*     | If not the end of storage group (indicated by ADUDONE flag)  *
*     | and an error return code of 8 or greater has been set then   *
*     | | Write message to the operator indicating the storage group *
*     | | reason code and return code                                 *
*     | Endif                                                         *
*     | If return code is 8 (indicating that CBRHADUX will not be    *
*     | called again)                                                 *
*     | | - Free the verify table (if it exists)                      *
*     | | - Close the output dataset                                  *
*     | | - Free data area                                            *
```

```
*      | Endif                                                        *
*      Endif                                                          *
*   - Perform standard exit linkage                                   *
*                                                                     *
*   NOTES:                                                            *
*                                                                     *
*      DEPENDENCIES: This routine is called on a group basis.  That   *
*                    is, the first and every call of a single         *
*                    instance of this routine is for the same storage *
*                    group.  This allows a return code indicating the *
*                    routine is not interested in a certain group     *
*                    (i.e. GROUPXX) and that the output data          *
*                    set can be segregated by group allowing for      *
*                    processing overlap (i.e. more than one task      *
*                    can have an instance of this routine because     *
*                    each task is writing to a seperate data set).    *
*                                                                     *
*                    The end of cycle call does not include the name  *
*                    of an object to be deleted.                      *
*                                                                     *
*                                                                     *
*      CHARACTER-CODE-DEPENDENCIES: EBCDIC Character Set              *
*                                                                     *
*      RESTRICTIONS: None                                             *
*                                                                     *
*      REGISTER CONVENTIONS:                                          *
*                                                                     *
*                Standard Entry Linkage                               *
*                                                                     *
*                R0 and R1 used for system service invocation         *
*                R2 and R3 are work registers                         *
*                R4  contains the ADUP address                        *
*                R5 through R10 are work registers                    *
*                R11 contains the address of the dynamic area         *
*                R12 is the base register                             *
*                                                                     *
*   MODULE TYPE: User Exit                                            *
*                                                                     *
*     PROCESSOR: Assembler                                        @P2C*
*                                                                     *
*     ATTRIBUTES:                                                     *
*      LOCATION: LINKLIB                                              *
*      STATE:    PROBLEM                                              *
*      TYPE:     REENTRANT                                            *
*                                                                     *
*   INPUT:                                                            *
*       SYMBOLIC NAME: CBRADUP                                        *
*       DESCRIPTION:   Control block containing the parameters for    *
*                      object deletion. (Contained in MACRO CBRADUP)  *
*                                                                     *
*       SYMOBLIC NAME: Verify Data Set ]OPTIONAL(                     *
*       DSN NAME:      HLQ.OBJECT.DELETE.VERIFY                       *
*       DESCRIPTION:   This is the input data set to the CBRHADUX     *
*                      which is used to determine that an object      *
*                      should not be expired.  The criteria           *
*                      is based on Object Name and Collection Name    *
*                      which is based upon the input to this module.  *
*                      This Data set also has a table created in      *
*                      storage for which comparisons shall be made.   *
*       ATTRIBUTES:                                                   *
*          ORGANIZATION:   PS                                         *
*          RECORD FORMAT:  FB                                         *
*          RECORD LENGTH:  89                                         *
*          BLOCK SIZE:     8900                                       *
*       RECORD FORMAT:                                                *
*          0-43  :   Object Name        CHAR(44)                      *
*          44    :   Blank                                            *
*          45-89 :   Collection Name    CHAR(44)                      *
*                                                                     *
*       NOTES:         Be sure to modify the (HLQ) high level         *
*                      qualifier and adjust accordingly to the        *
*                      private catalog that is accessible. An         *
*                      in-storage table will be built for these names *
*                      and will be formatted as follows:              *
*                                                                     *
*           VERIFY TABLE:                                             *
*                0-3    :  Number of entries in table                 *
*                4-7    :  Pointer to next table                      *
*                9-4095 :  Records from Input data set                *
*                                                                     *
*           NOTE: For last Table in Chain, the pointer to the next    *
*                  table is 00000000 (signifies end of chain)         *
```

```
*                                                                    *
*  OUTPUT:                                                           *
*      SYMBOLIC NAME: ADUUFLD                                        *
*      DESCRIPTION:   CBRADUP parameter list is updated to hold a    *
*                     pointer to the dynamic data ares for reentrant *
*                     calls for the same storage group.              *
*                                                                    *
*      SYMBOLIC NAME: Notify Data Set                                *
*      DSN NAME:      HLQ.XXXXXXXX.OBJECT.DELETE.NOTIFY              *
*      DESCRIPTION:   This is the output data set that contains the  *
*                     Object Name along with the Collection Name of  *
*                     each object deleted for the Storage group,     *
*                     which replaces the XXXXXXXX in the data set    *
*                     name, padded with the character X.  The few    *
*                     sections marked with MMM correlate to the      *
*                     output data set name.                          *
*      ATTRIBUTES:                                                   *
*         ORGANIZATION:   PS                                         *
*         RECORD FORMAT:  FB                                         *
*         RECORD LENGTH:  89                                         *
*         BLOCK SIZE:     8900                                       *
*      RECORD FORMAT:                                                *
*         0-43  :   Object Name        CHAR(44)                      *
*         44    :   Blank                                            *
*         45-89 :   Collection Name    CHAR(44)                      *
*                                                                    *
*   NOTE:  If this dataset is preallocated prior to the invocation   *
*          of this routine, the DCB declares of the dataset should   *
*          not be altered or removed.  Otherwise, modification to    *
*          the DCB statements can cause unexpected results.          *
*                                                                    *
*   NOTE:  Be sure to modify the VERIFY and NOTIFY data set names    *
*          prior to assembling and linking this module.  The         *
*          following labels need to be modified to reflect the High  *
*          Level Qualifier (HLQ) you would like to be used.          *
*                 1)  D1                                             *
*                 2)  NTFYDSN                                        *
*                 3)  VRFYDSN                                        *
*                                                                    *
*    RETURN CODES = 0  Allows for expiration of the object to        *
*                      continue and to continue calling the user     *
*                      exit for OSMC processing.                     *
*                   4  Do not allow expiration of this object for    *
*                      this instance of OSMC for this storage group. *
*                   8  Expire all objects in this storage group for  *
*                      this instance of OSMC but do not call the     *
*                      user exit again.                              *
*                   12 Do not allow for expiration of objects for    *
*                      this OSMC instance and this storage group and *
*                      call the user exit again.                     *
*                                                                    *
* CHANGE ACTIVITY:                                                   *
*                                                                    *
*      $L0=OAM,110,082687,TUCWV:  INITIAL RELEASE                    *
*      $L1=JDP3227,320,890523,TUCHTT:  RELEASE 1                     *
*      $D1=JDP3227,320,890523,TUCLJS:  COLLECTION NAMES              *
*      $L2=PRESCOTT,331,901112,TUCLJS:  PRESCOTT SUPPORT             *
*      $L3=OAMR1C,R1C,090311,TUCDMS:  RAS Improvements               *
*      $L4=OAMR21,R21,110215,TUCBJF:  RAS Improvements               *
*      $P1=K210268,R21,110714,TUCBJF: RAS Improvements               *
*      $P2=130050,R23,160817,TUCAED: Typo in prolog                  *
**********************************************************************
         TITLE 'CBRHADUX WORKING STORAGE'
*--------------------------------------------------------------------*
*                                                                    *
*        CBRHADUX WORKING STORAGE                                    *
*                                                                    *
*--------------------------------------------------------------------*
DATAREA  DSECT ,
SAVE1    DS    18F                 SAVE AREA
SAVE14   DS    1F                  R14 SAVE AREA FOR SUBROUTINES
FLAGS    DS    X                   FLAG AREA
TABUILT  EQU   X'80'               VERIFY TABLE HAS BEEN BUILT
ODCBO    EQU   X'40'               NOTIFY DATA SET IS OPEN
VRECSPTR DS    F                   ADDRESS OF THE IST 4K BLOCK OF
RETTXT   DS    AL2
         DS    AL2
         DS    AL2
         DS    CL8
*                                  OBJECT NAMES READ FROM THE VERIFY
*                                  DATA SET
RETCODE  DS    F                   INTERNAL RETURN CODE
```

```
REASCODE DS    F
S99RBPTR DS    F                   ADDRESS OF SVC99 RB
RENTAREA DS    0F                  REENTRANT COPY OF STATIC DEFINED
*                                  CONTROL BLOCKS
OPENL    OPEN   (,),MF=L
DCBS     DS    0F
DCBO     DCB  DDNAME=CBRADUXO,MACRF=(PM),OPTCD=W,LRECL=89,BLKSIZE=8900,+
               DSORG=PS,RECFM=FB,SYNAD=ADUXSYN
DCBI     DCB  DDNAME=CBRADUXI,MACRF=(GM),LRECL=89,BLKSIZE=8900,        +
               DSORG=PS,RECFM=FB,SYNAD=ADUXSYN,EODAD=REODAD
DYNRB    DS    0F
         DC    AL1(20)             RB LENGTH
         DC    AL1(01)             DSNAME ALLOCATION
         DC    X'C0'               FLAGS1 - NO EXIST ALLOC
         DC    X'0' FLAGS1
         DC    F'0'                ERROR CODES
TXTPTR   DC    A(TXTPTRV)          TEXT UNIT POINTERS
         DC    F'0'                RESERVED
*        DC    X'D1'               WAIT FOR VOLS,UNITS,DSNS AND MOUNTS
         DC    X'00'
         DC    AL3(0)
TXTPTRV  DS    0F
TXTDSN   DC    A(DSNTXT)
TXTDDN   DC    A(DDNMTXT)
TXTSTAT  DC    A(SHRTXT)
TXTDISP  DC    A(KEEPTXT)
TXTTRK   DC    A(0)
TXTPRIM  DC    A(0)
TXTSEC   DC    A(0)
TXTRET   DC    A(0)
TXTUNIT  DC    A(UNITTXT)
         DC    A(CLOSTXT)
         DC    X'80'
         DC    AL3(0)
DSNTXTN  DC    AL2(2)
         DC    AL2(1)
         DC    AL2(ENDDSNN-NTFYDSN)
*                                                                      *
*  Please note that modifying NTFYDS2 can cause unexpected results,   *
*  the XXXXXXXX is overlayed by the current storage group name.       *
*                                                                      *
NTFYDSN  DC    C'HLQ.'                                               MMM
NTFYDS2  DC    C'XXXXXXXX.OBJECT.DELETE.NOTIFY'
ENDDSNN  EQU   *
*
CONVERT  DS    D                   CONVERSION AREA FOR WTD
         DS    D                   EXTENSION OF WORK AREA
HEXWORK  DS    D                   CHARACTER CONVERSION FOR WTD
*---------------------------------------------------------------------*
*                                                                     *
*        SINGLE LINE WTO PARAMETER LIST                               *
*                                                                     *
*---------------------------------------------------------------------*
WTOLIST  WTO   TEXT=((,DE)),                                           +
               MF=L
WTOLISTL EQU   *-WTOLIST           SIZE OF WTO MACRO EXPANDED
         SPACE 2
*---------------------------------------------------------------------*
*                                                                     *
*        SINGLE LINE WTO TEXT LINE                                    *
*                                                                     *
*---------------------------------------------------------------------*
STATDISP EQU   *-DATAREA           DISPLACEMENT IN DATA AREA
STATLINE DS    0F                  STATUS CONTROL LINE MODEL
         DS    AL2                 LENGTH OF MESSAGE
         DS    C'CBRHADUX USER EXIT: SG ('  START OF WTO MSG
STATSG   DS    CL8                 STORAGE GROUP NAME
         DS    C') completed RC='  LABEL FOR RETURN CODE
STATRET  DS    CL8                 RETURN CODE
         DS    C' RSN='            LABEL FOR REASON CODE
STATREAS DS    CL8                 REASON CODE
STATLEN  EQU   *-STATLINE          LENGTH OF STATUS LINE
*
DATALEN  EQU   *-DATAREA
         EJECT
         TITLE 'CBRHADUX DSECT DEFINITIONS'
*--------------------------------------------------------------------*
*                                                                    *
*        CBRADUP - PARAMETER LIST DSECT DEFINITION                   *
*                                                                    *
*--------------------------------------------------------------------*
         CBRADUP
```

```
         SPACE 2
*---------------------------------------------------------------------*
*                                                                     *
*        MANAGEMENT CLASS DSECT TO MAP TO MCTAB                       *
*                                                                     *
*---------------------------------------------------------------------*
TAB      DSECT ,
MCLEN    DS    H
MCNAME   DS    CL30
TABLEN   EQU   *-TAB
         EJECT
         TITLE 'CBRHADUX - VERIFY FOR OBJECT DELETION EXIT'
***********************************************************************
*                                                                     *
*        CBRHADUX ENTRY POINT - MAINLINE PROCESSING                   *
*                                                                     *
***********************************************************************
CBRHADUX CSECT                      BEGINNING OF CBRHADUX
* CBRHADUX AMODE 31                   FOR 31 BIT ADDRESSING
* CBRHADUX RMODE ANY
ADUX     DS    0H                   ENTRY POINT
         USING *,R15                ALLOW FOR IMPLICIT ADDRESSING
         B     PASTID               BRANCH AROUND ID
         DC    CL8'CBRHADUX'        EYECATCHER FOR EXIT
         DC    CL8'&SYSDATE'        DATE IN THE EYECATCHER
PASTID   DS    0H                   BRANCH AROUND LABEL
         STM   R14,R12,12(R13)      SAVE CALLERS REGS
         LR    R12,R15              SET BASE REG
         DROP  R15
         USING ADUX,R12             IMPLICIT ADDRESSABILITY
         LR    R4,R1                GET INPUT PARAMETER LIST
         USING CBRADUP,R4           ADDRESSABILITY ON PARM LIST
         SYSSTATE AMODE64=NO ACSENV=P
**********************************************************************
*                                                                    *
*    Comment the following lines to allow for object deletion, by    *
*      default objects will not be deleted and a WTO is made.        *
*                  -- THREE LINES OF BYPASS CODE --                  *
*                                                                    *
**********************************************************************
*                                                                    *
         LA    R2,ADUFAILN          LOAD RETURN CODE 12 INTO R2
       WTO 'CBRHADUX USER EXIT: RC=12 OSMC bypassed object expiration'
         B     NOR13                BYPASS ALL CODE EXCEPT EXIT CODE
*                                                                    *
**********************************************************************
         L     R11,ADUUFLD          GET ADDRESS OF DYNAMIC AREA
         LTR   R11,R11              IS THIS THE FIRST CALL
         BNZ   NOGMAIN              NO, DO NOT GETMAIN
         GETMAIN RU,LV=DATALEN,SP=0  GET DYNAMIC AREA
         LR    R11,R1               GET STORAGE ADDRESS FOR DYNAMIC AREA
         USING DATAREA,R11          SET DYNAMIC AREA ADDRESSABILITY FOR
*                                   REENTRANCY
         SLR   R1,R1                GET A ZERO
         ST    R1,RETCODE           ZERO THE RETURN CODE
         ST    R1,REASCODE          ZERO THE REASON CODE
         ST    R1,VRECSPTR          ZERO THE IN-STORAGE NAMES FIELD
         STC   R1,FLAGS             ZERO THE FLAGS BYTE
         ST    R11,ADUUFLD          SAVE DYNAMIC AREA FOR NEXT CALL
NOGMAIN  DS    0H                   BRANCH AROUND FOR NO GETMAIN
         ST    R13,SAVE1+4          BACKCHAIN SAVE AREAS
         LA    R2,SAVE1             GET SAVE AREA ADDRESS
         ST    R2,8(,R13)           FORWARDCHAIN SAVE AREAS
         LR    R13,R2               ESTABLISH SAVE AREA ADDRESS
         SR    R2,R2                CLEAR FOR ZERO
         ST    R2,RETCODE           ZERO THE INTERNAL RETURN CODE
         ST    R2,REASCODE          ZERO THE INTERNAL REASON CODE
**********************************************************************
*  The following comparison is made to provide an example, of how to *
*  allow deletions for a particular storage group.  The following    *
*  four lines can be removed if this does not pertain to your needs. *
**********************************************************************
         CLC   ADUSGNAM,=CL8'GROUPXX' DO NOT PROCESS FURTHER IF    MMM
         BNE   NOTGRPXX             GROUPXX FOUND DO NOT PROCESS
         LA    R2,ADU8RS01          LOAD REASON CODE FOR GROUPXX
         ST    R2,REASCODE          STORE REASON CODE
         LA    R2,ADUFAILC          INDICATE DELETE IS OKAY BUT
         ST    R2,RETCODE           DO NOT CALL ADUX AGAIN
         B     ENDADUX              FOR THIS CYCLE
NOTGRPXX DS    0H
         CLI   ADUFUNC,ADUDONE      IS THIS AN END OF CYCLE CALL
         BE    DOFREE               YES, GO CLEANUP
```

```
         TM    FLAGS,TABUILT       HAS VERIFY TABLE BEEN BUILT
         BO    DOVERIFY            YES, DO NOT RE-OPEN DATA SET
         BAL   R14,OPENDS          FIRST CALL OPEN INPUT AND OUTPUT
         OI    FLAGS,TABUILT       INDICATE TABLE BUILT FOR NEXT CALL
*                                  VRECSPTR WILL BE ZERO IF NO
*                                  ENTRIES IN TABLE
DOVERIFY DS    0H
         L     R2,RETCODE          GET RETURN CODE
         LTR   R2,R2               DO NOT CONTINUE IF NON-ZERO
         BNZ   ENDADUX             END ADUX IF NON-ZERO CODE
         L     R10,VRECSPTR        GET ADDRESS OF IN-CORE VERIFY RECS
         LTR   R10,R10             IS THERE ANY VERIFY TABLE
         BZ    NOVERIFY            IF ZERO NO VERIFY NEEDED
         BAL   R14,VERIFY          CALL VERIFY ROUTINE
NOVERIFY DS    0H                  FREEMAIN TABLE
         L     R2,RETCODE          GET THE RETURN CODE
         LTR   R2,R2               IF NOT ZERO VERIFY FAILED
         BNZ   NONOTIFY            NO NOTIFY IF VERIFY FAILED
         BAL   R14,NOTIFY          ADD OBJECT TO NOTIFY DATA SET
NONOTIFY DS    0H                  DO NOT NOTIFY IF NON-ZERO CODE
ENDADUX  DS    0H                  END OF CBRADUX
**********************************************************************
*  If you would not like the CBRHADUX module to display return code  *
*  messages, the please comment out the following statement.         *
**********************************************************************
         BAL   R14,ADUXOUT         CALL WTO FOR ADUX STATUS
         LA    R3,ADUFAILC         RETURN CODE INDICATING NO
*                                  RECALL OF ADUX FOR
*                                  THIS CYCLE
         C     R3,RETCODE          IF ADUX IS TO BE CALLED AGAIN
         BH    NOFREE              DO NOT FREE AND CLOSE
DOFREE   DS    0H                  FREE THE VERIFY TABLE AND CLOSE
*                                  THE OUTPUT DATA SET
         BAL   R14,FREETAB         FREEMAIN THE VERIFY TABLE
         BAL   R14,CLOSEDS         CLOSE AND FREE OUTPUT DATA SET
         L     R13,SAVE1+4         GET CALLERS SAVE AREA ADDRESS
         L     R2,RETCODE          GET RETURN CODE
         SLR   R1,R1               GET A ZERO
         ST    R1,ADUUFLD          CLEAR THE USER FIELD TO AVOID
*                                  INADVERTENT USE OF FREEMAINED
*                                  STORAGE
         FREEMAIN RU,LV=DATALEN,SP=0,A=(R11)  FREE DYNAMIC AREA
         B     NOR13               BYPASS R13 RESTORE
NOFREE   DS    0H                  BYPASS FREEMAIN
         L     R13,4(R13)          RESTORE CALLERS SAVE AREA
NOR13    DS    0H                  BYPASS R13 RESTORE IF FREEMAIN PATH
         L     R14,12(R13)         GET RETURN ADDRESS
         LR    R15,R2              GET RETURN CODE
         LM    R0,R12,20(R13)      RESTORE CALLERS REGS
         BSM   0,R14               RETURN TO CALLER
         EJECT
         TITLE 'MCVERIFY - TEST MANAGEMENT CLASS FOR DELETION'
MCVERIFY DS    0H                  MC VERIFY SUBROUTINE
***********************************************************************
*                                                                     *
* SUBROUTINE:  MCVERIFY                                               *
*                                                                     *
* FUNCTION:    Search the in storage management class table to       *
*              determine if this object can be deleted. if the input  *
*              management class is listed in the table then no        *
*              objects with this management class are to be deleted. *
*                                                                     *
* OPERATION:                                                          *
*     1.  Load total number of Management Class Name Entries.         *
*     2.  Load address of the MC table.                               *
*     3.  If ADUMCNAM matches current MC table entry then set a       *
*         return code of 4 and a reason code of 1, do not delete      *
*         object but continue to process storage group.               *
*     4.  Return to calling module.                                   *
*                                                                     *
* CALLED BY:   Add a call from main line if desired                   *
*                                                                     *
* CALLS:       None                                                   *
*                                                                     *
* NOTE: This subroutine is provided as an example only and is not     *
*       called from the main line code.  Modify the main line code    *
*       as needed to invoke this routine if desired.                  *
*                                                                     *
***********************************************************************
         SPACE 1
         ST    R14,SAVE14          SAVE RETURN ADDRESS
         LA    R11,MCCNT           LOAD NUMBER OF TABLE ENTRIES
```

```
         LA    R10,MCTAB            LOAD ADDRESS OF TABLE IN R10
         USING TAB,R10              USE DSECT
COMPMC   CLC   MCNAME,ADUMCNAM      COMPARE MC NAMES
         BE    VMCMATCH
         LA    R10,TABLEN(,R10)
         BCT   R11,COMPMC
VMCMATCH DS    0H                   MC NAME IS IN VERIFY TABLE
         LA    R1,ADU4RS01          LOAD REASON CODE OF 1
         ST    R1,REASCODE          STORE REASON CODE
         LA    R1,ADUNODEL          GOOD COMPARE, DO NOT DELETE
         ST    R1,RETCODE           SET DO NOT DELETE RETURN CODE
ENDMC    DS    0H                   END MC VERIFY SUBROUTINE
         L     R14,SAVE14           GET SAVED R14
         BR    R14                  RETURN TO MAINLINE
         EJECT
         TITLE 'OPENDS - OPEN INPUT/OUTPUT DATASETS'
OPENDS   DS    0H                   OPENDS SUBROUTINE
***********************************************************************
*                                                                     *
* SUBROUTINE:  OPENDS                                                 *
*                                                                     *
* FUNCTION:    Open the input data set and create the verify table.   *
*              Then open the output data set for object deletion      *
*              notification.                                          *
*                                                                     *
* OPERATION:                                                          *
*     1.  Prepare Data Area for Input and Output control blocks,      *
*         along with the necessary SVC 99 RB.                         *
*.    2.  Begin input (Verify) data set processing.                   *
*     3.  Verify that the input data set was allocated successfully,  *
*         on failure set RC=12 and REAS=1. (ADUX not called again)    *
*     4.  Verify that the input data set is catalogued, on failure    *
*         set RC=12 and REAS=2. (ADUX not called again)               *
*     5.  Perform an OPEN on the Verify Data Set using OPEN MACRO.    *
*     6.  Verify input data set was opened successfully, on failure   *
*         set RC=12 and REAS=3. (ADUX not called again)               *
*     7.  Obtain storage to build the Verify table, using GETMAIN     *
*         MACRO using Subpool=0.                                      *
*     8.  Verify GETMAIN succeeded, on failure set RC=12, REAS=4      *
*         (ADUX not called again)                                     *
*     9.  Initialize the initial verify table and store a pointer to  *
*         the table in the Data Area.                                 *
*     10. Read a record from input data set using GET MACRO, on       *
*         failure call ADUXSYN and end input processing and read      *
*         until end of file.                                          *
*     11. Increment and store counter for the number of records.      *
*     12. If counter equals MAXNAMES then branch to obtain a new      *
*         block of storage for Verify Table, otherwise go to step 10. *
*         A. Obtain storage to get a new block for Verify table using *
*            GETMAIN MACRO with subpool=0.                            *
*         B. Verify GETMAIN succeeded, on failure RC=12 and REAS=5    *
*            (ADUX not called again)                                  *
*         C. Initialize the new block of storage for Verify Table.    *
*     13. Upon end of file condition close the dataset using CLOSE    *
*         MACRO.                                                      *
*     14. Verify no errors have occured while processing input before *
*         beggining output data set processing, on failure branch to  *
*         end of input/output data set processing.                    *
*     15. Begin output data set (NOTIFY) processing by updating the   *
*         output data set name with Storage Group name.               *
*     16. Update Text Unit pointers for the Notify data set.          *
*     17. Perform DYNALLOC for output data set and verify no errors   *
*         occured, on failure set RC=12 and REAS=6                    *
*     18. Attempt to open the output data set using the OPEN MACRO,   *
*         on failure set RC=12 and REAS=7 (ADUX not called again)     *
*     19. Return to main line code.                                   *
*                                                                     *
* REGISTER CONVENTIONS:                                               *
*     R8  - DCB Pointer                                               *
*     R11 - Data Area Pointer                                         *
*                                                                     *
***********************************************************************
         SPACE 2
         ST    R14,SAVE14           SAVE RETURN ADDRESS
         LA    R2,RENTAREA          GET TARGET ADDRESS FOR MOVE
         LA    R3,MOVELN            GET THE LENGTH OF THE MOVE
         LA    R6,STATAREA          GET SOURCE ADDRESS OF THE MOVE
         LR    R7,R3                GET LENGTH OF THE MOVE AND
*                                   PAD WITH ZEROS
         MVCL  R2,R6                COPY CONTROL BLOCKS TO
*                                   DYNAMIC AREA FROM STATIC AREA
         LA    R2,DYNRB             GET DYNAMIC RB ADDRESS
```

```
         ST    R2,S99RBPTR         STORE IN DYNAMIC RB POINTER FIELD
         OI    S99RBPTR,X'80'      SET END OF LIST FLAG
         LA    R2,TXTPTRV          GET DYNAMIC VERSION OF TXTPTR
         ST    R2,TXTPTR           STORE IN SVC 99 RB DYNAMIC VERSION
         LA    R2,0                GET ZERO FOR DDNAME TEXT UNIT PTR
         ST    R2,TXTDDN           PUT IT IN THE TXT UNIT PTR
         MVC   RETTXT(6),RETSTAT   MOVE FROM STATIC TO DYNAMIC
         LA    R2,RETTXT           LET SYSTEM DETERMINE DDNAME
         ST    R2,TXTRET
         LA    R1,S99RBPTR         SETUP REG 1 FOR DYNAMIC ALLOC.
         DYNALLOC                  ALLOCATE INPUT DATA SET
         LA    R2,4                GET DYNAMIC ENVIRONMENT ERROR
         CR    R15,R2              WAS DYNAMIC ALLOC. OKAY
         BL    DOOPENI             YES, OPEN VERIFY DATA SET
         BE    CHKNODS             IF ERROR CHECK IF NO DATA SET
         LA    R15,ADUCRS01        SET REASON CODE OF 1
         ST    R15,REASCODE        STORE REASON CODE
         LA    R15,ADUFAILN        ERROR CAUSES NO RETURN TO ADUX
         ST    R15,RETCODE         SET RETURN CODE
         B     ENDOPEN             END ADUX
CHKNODS  DS    0H                  ENVIRONMENT ERROR
         LH    R15,=X'1708'        GET NO CATALOGED DS FAIL CODE
         CH    R15,DYNRB+4         FAILED BECAUSE DS NOT CATALOGED
         BE    NOOPENI             SO PROCEED WITHOUT VERIFY
         LA    R15,ADUCRS02        SET REASON CODE OF 2
         ST    R15,REASCODE        STORE REASON CODE
         LA    R15,ADUFAILN        ERROR CAUSES NO RETURN TO ADUX
         ST    R15,RETCODE         SET RETURN CODE
         B     ENDOPEN             END ADUX
DOOPENI  DS    0H                  ALLOCATED, DO OPEN
         LA    R1,OPENL            GET OPEN LIST ADDRESS
         LA    R8,DCBI             GET INPUT DCB ADDRESS
         USING IHADCB,R8
         MVC   DCBDDNAM,RETTXT+6   MOVE SYSTEM GENERATED DDN INTO DCB
         OPEN  ((R8),INPUT),MF=(E,(1)) OPEN INPUT DCB
         TM    DCBOFLGS,DCBOFOPN   WAS DS OPENED PROPERLY
         BO    OPENI               BUILD VERIFY TABLE
         LA    R15,ADUCRS03        LOAD REASON CODE OF 3
         ST    R15,REASCODE        STORE REASON CODE
         LA    R15,ADUFAILN        ERROR CAUSES NO RETURN TO ADUX
         ST    R15,RETCODE         SET RETURN CODE
         B     ENDOPEN             END ADUX
*
*        Build Verify Table
*
OPENI    DS    0H
         GETMAIN RU,LV=4096,SP=0   GET BLOCK FROM SUBPOOL ZERO
*
         LTR   R15,R15             VERIFY GETMAIN WAS SUCCESSFUL
         BZ    BTABBEG             BRANCH TO BUILD INITIAL TABLE
         LA    R15,ADUCRS04        LOAD REASON CODE OF 4
         ST    R15,REASCODE        STORE REASON CODE
         LA    R15,ADUFAILN        ERROR CAUSES NO RETURN TO ADUX
         ST    R15,RETCODE         SET RETURN CODE
         B     ENDOPEN             END OPEN DATA SET
*
BTABBEG  LR    R5,R1               GET THE VERIFY TABLE ADDRESS
         SLR   R6,R6               GET A ZERO
*                                  ALSO, USE AS A RECORD COUNTER
         ST    R6,0(,R5)           INDICATE IN-CORE TABLE NOT YET SET
         ST    R6,4(,R5)           INDICATE IN-CORE TABLE NOT YET SET
         LA    R9,8(,R5)           GET PAST RECORD COUNTER
         ST    R5,VRECSPTR         SAVE TABLE ADDRESS
GETINPUT DS    0H
         LA    R1,DCBI             GET INPUT DCB ADDRESS
         LR    R0,R9               ADDRESS OF RECORD
         GET   (1),(0)             READ A RECORD
         L     R2,RETCODE          CHECK FOR SYNAD ENTRY
         LTR   R2,R2               IF NON-ZERO SYNAD ENTERED
         BNZ   REODAD              CLOSE DATA SET AND END
         LA    R9,89(R9)           GET TO NEXT SLOT IN VERIFY TBL
         LA    R6,1(0,R6)          INCREMENT NUMBER OF RECORDS
         ST    R6,0(0,R5)          UPDATE COUNTER IN BLOCK
         C     R6,MAXNAMES         SEE IF MAX NAMES IN TABLE YET
         BE    GNEXTBLK            IF SO, GET A NEW BLOCK
         B     GETINPUT            READ UNTIL EODOD
GNEXTBLK DS    0H
         GETMAIN RU,LV=4096,SP=0   GET BLOCK FROM SUBPOOL ZERO
*
         LTR   R15,R15             VERIFY GETMAIN WAS SUCCESSFUL
         BZ    BTABNEXT            BRANCH TO BUILD INITIAL TABLE
         LA    R15,ADUCRS05        LOAD REASON CODE OF 5
```

```
         ST    R15,REASCODE        STORE REASON CODE
         LA    R15,ADUFAILN        ERROR CAUSES NO RETURN TO ADUX
         ST    R15,RETCODE         SET RETURN CODE
         B     ENDOPEN             END OPEN DATA SET
*
BTABNEXT ST    R1,4(0,R5)          CHAIN TO CURRENT TABLE
         LR    R5,R1               GET THE VERIFY TABLE ADDRESS
         SLR   R6,R6               CLEAR TO ZERO
         ST    R6,0(,R5)           INDICATE IN-CORE TABLE NOT YET SET
         ST    R6,4(,R5)           INDICATE IN-CORE TABLE NOT YET SET
         LA    R9,8(,R5)           GET PAST RECORD COUNTER
         B     GETINPUT            READ UNTIL EODOD
REODAD   DS    0H                  END OF DATA ON READ
         LA    R2,DCBI             GET INPUT DCB POINTER
         LA    R1,OPENL            GET AREA FOR CLOSE LIST
         CLOSE ((R2)),MF=(E,(1))   CLOSE INPUT DCB
         FREEPOOL DCBI             RELEASE BUFFER POOL
NOOPENI  DS    0H                  NO VERIFY DATA SET
         L     R2,RETCODE          CHECK FOR ERROR
         LTR   R2,R2               IF NON-ZERO ERROR OCCURRED
         BNE   ENDOPEN             END PROCESSING
*
* BEGIN OUTPUT DATASET PROCCESSING
*
         LA    R6,NTFYDS2          LOAD ADDRESS OF SG NAME IN VERIFY
*                                  DATA SET NAME
         L     R7,ADUSGLEN         LOAD LENGTH OF SG NAME
         LA    R8,ADUSGNAM         LOAD ADDRESS OF SG NAME FROM
*                                  CBRADUP
         LR    R9,R7               PREPARE FOR MVCL
         MVCL  R6,R8               MOVE SG NAME INTO THE VERIFY
*                                  DATA SET NAME
*
* UPDATE TEXT UNIT POINTERS FOR THE NOTIFY DATA SET
*
         LA    R3,0                GET ZERO FOR DDNAME TEXT UNIT PTR
         ST    R3,TXTDDN           PUT IT IN THE TXT UNIT PTR
         LA    R3,DSNTXTN          GET NOTIFY DSNAME TEXT UNIT PTR
         ST    R3,TXTDSN           PUT IT IN THE TXT UNIT PTR
*                                  INSTEAD OF VERIFY DSNAME
         LA    R3,MODTXT           GET STAT=SHR TEXT UNIT PTR
         ST    R3,TXTSTAT          STORE NEW TEXT UNIT PTR
         LA    R3,CTLGTXT          GET DISP=,KEEP TEXT UNIT PTR
         ST    R3,TXTDISP          STORE NEW TEXT UNIT PTR
         LA    R3,TRKTXT           GET TRACK ALLOATION TU PTR
         ST    R3,TXTTRK           STORE NEW TEXT UNIT PTR
         LA    R3,PRIMTXT          GET PRIMARY AMOUNT TU PTR
         ST    R3,TXTPRIM          STORE NEW TEXT UNIT PTR
         LA    R3,SECTXT           GETSECONDARY AMOUNT TU PTR
         ST    R3,TXTSEC           STORE NEW TEXT UNIT PTR
         MVC   RETTXT(6),RETSTAT   MOVE FROM STATIC TO DYNAMIC
         LA    R3,RETTXT           LET SYSTEM DETERMINE DDNAME
         ST    R3,TXTRET
         LA    R1,S99RBPTR         SET UP FOR SVC 99
         DYNALLOC                  FREE THE DATA SET
         LTR   R15,R15             IF DYNALLOC OKAY
         BZ    OPENO               OPEN NOTIFY DATA SET
         LA    R15,ADUCRS06        LOAD REASON CODE OF 6
         ST    R15,REASCODE        STORE REASON CODE
         LA    R15,ADUFAILN        ERROR CAUSES NO RETURN TO ADUX
         ST    R15,RETCODE         SET RETURN CODE
         B     ENDOPEN             END ADUX
OPENO    DS    0H                  OPEN OUTPUT DATA SET
         LA    R1,OPENL            GET OPEN LIST ADDRESS
         LA    R8,DCBO             GET INPUT DCB ADDRESS
         MVC   DCBDDNAM,RETTXT+6   MOVE SYSTEM GENERATED DDN INTO DCB
         OPEN  ((R8),OUTPUT),MF=(E,(1)) OPEN OUTPUT DCB
         USING IHADCB,R8
         TM    DCBOFLGS,DCBOFOPN   WAS DS OPENED PROPERLY
         BNO   NOOPENO             SET NO NOTIFY TABLE
         OI    FLAGS,ODCBO         INDICATE NOTIFY DATA SET IS OPEN
         B     ENDOPEN             END DATA SET OPENING
NOOPENO  DS    0H                  OUTPUT DATA SET DID NOT OPEN
         LA    R2,ADUCRS07         LOAD REASON CODE OF 7
         ST    R2,REASCODE         STORE REASON CODE
         LA    R2,ADUFAILN         SET FAILING RETURN CODE
         ST    R2,RETCODE          DO NOT RECALL ADUX
ENDOPEN  DS    0H                  END OF OPENDS SUBROUTINE
         L     R14,SAVE14          GET RETURN ADDRESS
         BR    R14                 RETURN TO MAINLINE
         EJECT
VERIFY   DS    0H                  VERIFY SUBROUTINE
```

```
***********************************************************************
*                                                                     *
* SUBROUTINE:  VERIFY                                                 *
*                                                                     *
* FUNCTION:    Search the Verify Table, built by OPENDS.  If the      *
*              object name is in the table then send a return code    *
*              indicating that auto-deletion should not occur.        *
*                                                                     *
* OPERATION:                                                          *
*     1.  Load address of the table pointer from Data Area.           *
*     2.  Retrieve the number of entries from Verify Table.           *
*     3.  If number of entries = 0 then end VERIFY.                   *
*     4.  While Names and collections to verify                       *
*         A. If ADUONAME = row in Verify Table, then check Collection *
*         B. Read next record from Verify Table.                      *
*         C. If entries = 0 then move to next Verify Table.           *
*         D. If ADUCLNAM = row in Verify Table, then set RC=4 and     *
*            REAS=2 and do not delete object.                         *
*     5.  Branch back to main line code.                              *
*                                                                     *
***********************************************************************
         SPACE 1
         ST    R14,SAVE14          SAVE RETURN ADDRESS
         L     R5,VRECSPTR         GET FIRST TABLE ADDRESS
COMPLOOP DS    0H                  COMPARE LOOP
         L     R3,0(,R5)           GET COUNT OF ENTRIES
         LA    R2,8(,R5)           GET ADDRESS OF FIRST NAME
         LTR   R3,R3               IF EMPTY FILE
         BZ    ENDVERF             GO TO ENDVERF
COMPNAME DS    0H                  NAME COMPARE LOOP
         CLC   0(44,R2),ADUONAME   COMPARE NAME IN VERIFY TABLE
*                                  AGAINST OBJECT BEING DELETED
         BE    VCOLNAM             IF MATCH, CHECK COLLECTION NAME
DIFCOLNM LA    R2,89(,R2)          GET ADDRESS OF NEXT NAME
         BCT   R3,COMPNAME         DECREMENT COUNTER BY 1
         L     R5,4(,R5)           GET NEXT TABLE SECTION ADDRESS
         LTR   R5,R5               IF THERE IS A NEXT SECTION
         BZ    ENDVERF             END VERIFY IF NOT
         B     COMPLOOP            COMPARE NEXT NAME
VCOLNAM  DS    0H                  OBJ MATCH, CHECK COL NAME
         CLC   45(44,R2),ADUCLNAM  OBJ MATCH, SAME COL NAME?
         BE    VMATCH              IF MATCH, DO NOT DELETE
         B     DIFCOLNM            NO MATCH, GET NEXT OBJ NAME
VMATCH   DS    0H                  OBJECT NAME IS IN VERIFY TABLE
         LA    R1,ADU4RS02         LOAD REASON CODE OF 2
         ST    R1,REASCODE         STORE REASON CODE
         LA    R1,ADUNODEL         GOOD COMPARE, DO NOT DELETE
         ST    R1,RETCODE          SET DO NOT DELETE RETURN CODE
ENDVERF  DS    0H                  END VERIFY SUBROUTINE
         L     R14,SAVE14          GET SAVED R14
         BR    R14                 RETURN TO MAINLINE
         EJECT
NOTIFY   DS    0H                  NOTIFY SUBROUTINE
***********************************************************************
*                                                                     *
* SUBROUTINE:  NOTIFY                                                 *
*                                                                     *
* FUNCTION:    Write the object and collection names to the notify    *
*              data set.  This data set can be read by another        *
*              application for removal of object information.        *
*                                                                     *
* OPERATION:                                                          *
*     1.  Load address of the output DCB.                             *
*     2.  Load address of Object name and collection name output line *
*         for output to NOTIFY data set.                              *
*     3.  Write record to data set using PUT MACRO.                   *
*     4.  Branch back to main line processing.                        *
*                                                                     *
***********************************************************************
         SPACE 1
         ST    R14,SAVE14          SAVE RETURN ADDRESS
         LA    R1,DCBO             GET OUTPUT DCB ADDRESS
         LA    R0,ADUOBJCL         GET OBJECT NAME/COLL. ADDRESS
         PUT   (1),(0)             WRITE NAME TO NOTIFY DATA SET
         L     R14,SAVE14          GET SAVED R14
         BR    R14                 RETURN TO MAINLINE
         EJECT
FREETAB  DS    0H                  FREEMAIN VERIFY TABLE SUBROUTINE
***********************************************************************
*                                                                     *
* SUBROUTINE:  FREETAB                                                *
*                                                                     *
```

```
* FUNCTION:    Free the in-storage Verify Table.                      *
*                                                                     *
* OPERATION:                                                          *
*     1.  Load address of the Verify Table Head.(VRECSPTR)            *
*     2.  While (tables exist ]Pointer(R5) != 0()                     *
*         A.  Free the table storage by using the FREEMAIN macro.     *
*         B.  Load the next table address from current table.         *
*     3. Return to main line code.                                    *
*                                                                     *
***********************************************************************
         SPACE 1
         ST    R14,SAVE14          SAVE RETURN ADDRESS
         L     R5,VRECSPTR         GET FIRST TABLE ADDRESS
FREELOOP DS    0H                  LOOP THROUGH CHAINED TABLES
         LTR   R5,R5               IS THERE A TABLE ADDRESS
         BZ    ENDFREE             IF NOT, END FREEMAIN LOOP
         L     R3,4(,R5)           GET NEXT TABLE ADDRESS
         FREEMAIN RU,LV=4096,A=(R5) FREE TABLE SECTION
         LR    R5,R3               ADDRESS NEXT SECTION TO FREE
         B     FREELOOP            FREEMAIN NEXT SECTION
ENDFREE  DS    0H                  END OF TABLE FREEMAIN LOOP
         L     R14,SAVE14          GET SAVED R14
         BR    R14                 RETURN TO MAINLINE
         EJECT
CLOSEDS  DS    0H                  CLOSE DATA SET SUBROUTINE
***********************************************************************
*                                                                     *
* SUBROUTINE:  CLOSEDS                                                *
*                                                                     *
* FUNCTION:    Closes the Notify data set.                            *
*                                                                     *
* OPERATION:                                                          *
*     1.  Check to see if Notify data set is currently open.          *
*     2.  Prepare for close macro.                                    *
*         A. Load address of the Open List. (R1)                      *
*         B. Load address of the Notify DCB. (R2)                     *
*     3.  Perform a close on the Notify data set using the CLOSE      *
*         macro.                                                      *
*     4.  Free the DCB buffer pool using the FREEPOOL macro.          *
*     5.  Return to main line code.                                   *
*                                                                     *
***********************************************************************
         SPACE 1
         ST    R14,SAVE14          SAVE RETURN ADDRESS
         TM    FLAGS,ODCBO         CHECK TO SEE IF OUTPUT DS IS OPEN
         BNO   NOCLOSE             DO NOT ISSUE CLOSE IF NOT OPEN
         LA    R1,OPENL            GET AREA FOR CLOSE LIST
         LA    R2,DCBO             GET DCB ADDRESS
         CLOSE ((R2)),MF=(E,(1))   CLOSE OUTPUT DCB NOTIFY DSN
         FREEPOOL DCBO             RELEASE OUTPUT BUFFER POOL
NOCLOSE  DS    0H                  DO NOT CLOSE
         L     R14,SAVE14          GET SAVED R14
         BR    R14                 RETURN TO MAINLINE
         EJECT
ADUXSYN  DS    0H                  BEGINNING OF WTO SUBROUTINE
***********************************************************************
*                                                                     *
* SUBROUTINE:  ADUXSYN                                                *
*                                                                     *
* FUNCTION:    Handles input and output errors during file operations.*
*                                                                     *
* OPERATION:                                                          *
*     1.  Set a RC=12 and REAS=8 (ADUX not called again) due to I/O   *
*         error.                                                      *
*     2.  Store the return code for ADUX.                             *
*     3.  Return to calling module.                                   *
*                                                                     *
***********************************************************************
         SPACE 1
         LA    R2,ADUCRS08         LOAD REASON CODE OF 8
         ST    R2,REASCODE         STORE REASON CODE
         LA    R2,ADUFAILN         SET FAIL CODE TO NO DELETE AND
*                                  DO NOT CALL ADUX FOR THIS SG
         ST    R2,RETCODE          STORE RETURN CODE
         BR    R14                 RETURN TO MAINLINE
         EJECT
ADUXOUT  DS    0H
***********************************************************************
*                                                                     *
* SUBROUTINE:  ADUXOUT                                                *
*                                                                     *
* FUNCTION:    Writes to the console to show if an error occurred     *
```

```
*              while the CBRHADUX was processing.                   *
*                                                                   *
* OPERATION:                                                        *
*     1.  Determine whether the ADUXOUT should be performed based on *
*         ADUFUNC and return code severity.                         *
*     2.  Copy from static to dynamic memory, the WTO macro list    *
*         definition along with the print line label.               *
*     3.  Move current Storage group name to WTO message line.      *
*     4.  Convert the current return code and move to the WTO message *
*         line.                                                     *
*     5.  Convert the current reason code and move to the WTO message *
*         line.                                                     *
*     6.  Prepare to perform the WTO macro.                         *
*         A. Set the address of the message. (R5)                   *
*         B. Clear register 0 to zero.                              *
*     7.  Write to operator using the WTO macro.                    *
*     8.  Return to main line code.                                 *
*                                                                   *
* NOTE: If you change the return code severity to allow anything    *
*       below 8, can cause unexpected results, such as the WTO      *
*       could become full.  Modify the value of RETMIN to adjust the *
*       severity of messages to be displayed.                       *
*                                                                   *
*********************************************************************
         SPACE 1
         ST    R14,SAVE14          SAVE RETURN ADDRESS
*
         CLI   ADUFUNC,ADUDONE     IS THIS AN END OF CYCLE CALL
         BE    OUTDONE             DO NOT PERFORM WTO ON EQUAL
         L     R5,RETCODE          LOAD CURRENT RETURN CODE
         C     R5,RETMIN           IF RETURN CODE < RETMIN THEN
         BL    OUTDONE              ENDADUX OUT
         MVC   WTOLIST(WTOLISTL),WTOMODEL  COPY FROM MODULE STORAGE
*                                          TO WORKING STORAGE
         MVC   STATLINE(STATLEN),STATMODL  COPY FROM MODULE STORAGE
*                                          TO WORKING STORAGE
         MVC   STATSG(8),ADUSGNAM  MOVE SG TO STATUS LINE
         L     R5,RETCODE          LOAD CURRENT RETURN CODE
         CVD   R5,CONVERT          CONVERT TO PACK FORMAT
         UNPK  HEXWORK(8),CONVERT+4(4)  UNPACK FOR CONVERSION
         OI    HEXWORK+7,X'F0'     FORMAT LAST BYTE TO EBCDIC
         MVC   STATRET(8),HEXWORK  MOVE TO STATUS LINE
         L     R5,REASCODE         LOAD CURRENT REASON CODE
         CVD   R5,CONVERT          CONVERT TO PACK FORMAT
         UNPK  HEXWORK(8),CONVERT+4(4)  UNPACK FOR CONVERSION
         OI    HEXWORK+7,X'F0'     FORMAT LAST BYTE TO EBCDIC
         MVC   STATREAS(8),HEXWORK MOVE TO STATUS LINE
*
*        PERFORM WRITE TO OPERATOR REQUEST
*
         LR    R5,R11              LOAD WORKING STORAGE AREA
         LA    R6,STATDISP         LOAD OFFSET OF STATUS LINE
         AR    R5,R6               FIND DISPLACEMENT TO STATUS
*                                  LINE FROM WORKING STORAGE
         SR    R0,R0               CLEAR REGISTER 0
         WTO   TEXT=(R5),                                            +
               MF=(E,WTOLIST)      PERFORM WTO TO CONSOLE
OUTDONE  L     R14,SAVE14          RESTORE RETURN ADDRESS
         BR    R14                 RETURN TO MAINLINE
         EJECT
*********************************************************************
*                                                                   *
*        LITERAL DEFINITIONS                                        *
*                                                                   *
*********************************************************************
         LTORG ,                   ORGANIZE LITERALS
R0       EQU   0
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
```

```
*-------------------------------------------------------------------*
*                                                                   *
*        MAX RETURN CODE TO DETERMINE WHAT SEVERITY OF MESSAGE      *
*         SHOULD BE REPORTED                                        *
*                                                                   *
*-------------------------------------------------------------------*
RETMIN   DC    F'8'                HIGHEST RETURN CODE THAT THE     MMM
*                                  ADUXOUT SHOULD WTO MESSAGES FOR
*-------------------------------------------------------------------*
*                                                                   *
*        REASON CODES FOR DIAGNOSTIC REASONS                        *
*                                                                   *
*-------------------------------------------------------------------*
*
* RETURN CODE ADUNODEL = RETCODE = 4
*
ADU4RS01 EQU   1    MANAGEMENT CLASS REJECTED OBJECT TO BE DELETED
ADU4RS02 EQU   2    OBJECT RESIDES IN VERIFY DATASET, NO DELETION
*
* RETURN CODE ADUFAILC = RETCODE = 8
*
ADU8RS01 EQU   1    DELETE ALL OBJECTS FOR THIS STORAGE GROUP AND
*                   DO NOT CALL EXIT AGAIN
*
* RETURN CODE ADUFAILN = RETCODE = 12
*
ADUCRS01 EQU   1    DYNALLOC FAILED FOR VERIFY DATASET
ADUCRS02 EQU   2    VERIFY DATASET NOT CATOLOGED
ADUCRS03 EQU   3    VERIFY DATASET FAILED TO OPEN
ADUCRS04 EQU   4    GETMAIN FAILED FOR INITIAL ON VERIFY TABLE
ADUCRS05 EQU   5    GETMAIN FAILED FOR NEXT ON VERIFY TABLE
ADUCRS06 EQU   6    DYNALLOC FAILED FOR NOTIFY DATASET
ADUCRS07 EQU   7    NOTIFY DATASET FAILED TO OPEN
ADUCRS08 EQU   8    I/O ERROR OCCURED DURING GETMAIN
*-------------------------------------------------------------------*
*                                                                   *
*        SINGLE LINE WTO PARAMETER LIST                             *
*                                                                   *
*-------------------------------------------------------------------*
WTOMODEL WTO   TEXT=((,DE)),                                         +
               MF=L
         SPACE 2
*-------------------------------------------------------------------*
*                                                                   *
*        SINGLE LINE WTO MODEL LINE                                 *
*                                                                   *
*-------------------------------------------------------------------*
STATMODL DC    0F'0'               STATUS LINE MODEL FOR WORKING
*                                  STORAGE
         DC    AL2(STATMLEN-2)     LENGTH OF MESSAGE FOR WTO MACRO
         DC    C'CBRHADUX USER EXIT: SG ('  START OF WTO MSG
         DC    C'        '         PLACE HOLDER FOR STORAGE GROUP
         DC    C') completed RC='  LABEL FOR RETURN CODE
         DC    C'        '         PLACE HOLDER FOR RETURN CODE
         DC    C' RSN='            LABEL FOR REASON CODE
         DC    C'        '         PLACE HOLDER FOR REASON CODE
STATMLEN EQU   *-STATMODL          LENGTH OF STATUS LINE
*
*-------------------------------------------------------------------*
*                                                                   *
*        MANAGEMENT CLASS VERIFICATION                              *
*                                                                   *
*-------------------------------------------------------------------*
MCTAB    DC    C'MCNODEL1'         MANAGEMENT CLASS TABLE
         DC    C'MCNODEL2'
MCCNT    EQU   (*-MCTAB)/8
MAXNAMES DC    F'44'               MAXIMUM NUMBER OF NAMES IN
*                                  A 4K BLOCK OF VERIFY TABLE
*-------------------------------------------------------------------*
*                                                                   *
*        DD STATEMENT DECLARATIONS                                  *
*                                                                   *
*-------------------------------------------------------------------*
RETSTAT  DC    AL2(85)
         DC    AL2(1)
         DC    AL2(8)
DDNMTXT  DS    0F
         DC    AL2(1)
         DC    AL2(1)
         DC    AL2(ENDDDN-STRTDDN)
STRTDDN  DC    C'CBRADUXI'
ENDDDN   EQU *
```

```
MODTXT   DC    AL2(4)              STATUS=MOD
         DC    AL2(1)
         DC    AL2(1)
         DC    AL1(2)
SHRTXT   DC    AL2(4)              STATUS=SHR
         DC    AL2(1)
         DC    AL2(1)
         DC    AL1(8)
TRKTXT   DC    AL2(7)              TRACK ALLOCATION IF NOT OLD
         DC    AL2(0)
PRIMTXT  DC    AL2(10)
         DC    AL2(1)
         DC    AL2(3)
         DC    AL3(10)             10 TRACKS PRIMARY ALLOCATION
SECTXT   DC    AL2(11)
         DC    AL2(1)
         DC    AL2(3)
         DC    AL3(10)             10 TRACKS SECONDARY ALLOCATION
CTLGTXT  DC    AL2(5)              DISP=CATLG
         DC    AL2(1)
         DC    AL2(1)
         DC    AL1(2)
KEEPTXT  DC    AL2(5)              DISP=KEEP
         DC    AL2(1)
         DC    AL2(1)
         DC    AL1(8)
UNITTXT  DC    AL2(21)
         DC    AL2(1)
         DC    AL2(5)
         DC    C'SYSDA'
CLOSTXT  DC    AL2(28)             FREE DATA SET WHEN CLOSED
         DC    AL2(0)
DSNTXT   DC    AL2(2)
         DC    AL2(1)
         DC    AL2(ENDDSN-VRFYDSN)
VRFYDSN  DC    C'HLQ.OBJECT.DELETE.VERIFY'
ENDDSN   EQU   *
STATAREA DS    0F                  STATIC CBS TO BE MOVED
         OPEN   (,),MF=L
DCBSTAT  DS    0F
*   NOTE:  IF THIS DATASET IS PREALLOCATED PRIOR TO THE INVOCATION   *
*          OF THIS ROUTINE, THE DCB DECLARES OF THE DATASET SHOULD   *
*          BE REMOVED FROM THIS ROUTINE.  OTHERWISE, THE ALLOCATION  *
*          HERE WILL OVERRIDE THE PREALLOCATION OF THE DATASET,      *
*          CAUSING UNEXPECTED OUTPUT.                                *
*                                                                    *
         DCB  DDNAME=CBRADUXO,MACRF=(PM),OPTCD=W,LRECL=89,BLKSIZE=8900,+
               DSORG=PS,RECFM=FB,SYNAD=ADUXSYN
         DCB  DDNAME=CBRADUXI,MACRF=(GM),LRECL=89,BLKSIZE=8900,        +
               DSORG=PS,RECFM=FB,SYNAD=ADUXSYN,EODAD=REODAD
DYNRBC   DS    0F
         DC    AL1(20)             RB LENGTH
         DC    AL1(01)             DSNAME ALLOCATION
         DC    X'C0'               FLAGS1 - NO EXIST ALLOC
         DC    X'0' FLAGS1
         DC    F'0'                ERROR CODES
         DC    A(0)                TEXT UNIT POINTERS
         DC    F'0'                RESERVED
         DC    X'00'               WAIT FOR VOLS,UNITS,DSNS AND MOUNTS
*        DC    X'D1'               WAIT FOR VOLS,UNITS,DSNS AND MOUNTS
         DC    AL3(0)
TXTPTRC  DS    0F
         DC    A(DSNTXT)
         DC    A(DDNMTXT)
         DC    A(SHRTXT)
         DC    A(KEEPTXT)
         DC    A(0)                TRACK TEXT UNIT FOR OUTPUT DS
         DC    A(0)                PRIMARY TEXT UNIT FOR OUTPUT DS
         DC    A(0)                SECONDARY TEXT UNIT FOR OUTPUT DS
         DC    A(0)                RETURN DDNAME FOR OUTPUT DS
         DC    A(UNITTXT)          UNIT TEXT UNIT FOR OUTOUT DS
         DC    A(CLOSTXT)          FREE (UN-ALLOCATE) AT CLOSE
         DC    X'80'
         DC    AL3(0)
         DC    AL2(2)
         DC    AL2(1)
         DC    AL2(ENDD1-D1)
D1       DC    C'HLQ.XXXXXXXX.OBJECT.DELETE.NOTIFY'                 MMM
ENDD1    EQU   *
MOVELN   EQU   *-STATAREA
         EJECT
         DCBD    DSORG=QS
```

```
         EJECT
         IEFZB4D2
         END CBRHADUX
```

Appendix G. 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 document helps you plan, install, and administer the object access method (OAM), a component of DFSMSdfp. This document primarily documents intended programming interfaces that allow the customer to write programs to obtain the services of the OAM component of DFSMSdfp.

This document also documents information that is NOT intended to be used as Programming Interfaces of the OAM component of DFSMSdfp. This information is identified where it occurs, either by an introductory statement to a topic or section or by the following marking:

[NOT Programming Interface Information] [End NOT Programming Interface Information]

Trademarks

Glossary

This glossary defines technical terms and abbreviations used in DFSMS documentation. If you do not find the term you are looking for, refer to the index of the appropriate DFSMS manual.

This glossary includes terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published part of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.
- The *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

The following cross-reference is used in this glossary:

See:
: This refers the reader to (a) a related term, (b) a term that is the expanded form of an abbreviation or acronym, or (c) a synonym or more preferred term.

3480
: IBM 3480 Magnetic Tape Subsystem. A group of magnetic tape controllers and drives supporting cartridge system tape (as opposed to reel tape). There are two controller models, A11 and A22, and two drive models, B11 and B22.

3490
: IBM 3490 Magnetic Tape Subsystem. A group of magnetic tape controllers and drives supporting cartridge system tape (as opposed to reel tape).

3490E
: IBM 3490E Magnetic Tape Subsystem. A group of enhanced capability tape controllers and drives supporting cartridge system tape (as opposed to reel tape).

3590
: IBM TotalStorage Enterprise Tape System 3590.

3590B1*x*
: IBM TotalStorage Enterprise Tape Drive 3590 Model B1*x*.

3590E1*x*
: IBM TotalStorage Enterprise Tape Drive 3590 Model E1*xx*.

3590H1*x*
: IBM TotalStorage Enterprise Tape Drive 3590 Model H1*xx*.

3592J1A
: IBM TotalStorage Enterprise Tape Drive 3592.

3592E05
: IBM Enterprise Tape Drive 3592 Model E05.

3592E06
: IBM Enterprise Tape Drive 3592 Model E06.

3592E07
: IBM Enterprise Tape Drive 3592 Model E07.

access path
The path DB2 uses to get to data specified in SQL statements. An access path can involve an index, a sequential search, or a combination of both.

ACDS
See *active control data set.*

ACS
See *automatic class selection.*

active configuration
The configuration currently used by SMS to control the managed storage in the installation.

active control data set (ACDS)
A VSAM linear data set that contains an SCDS that has been activated to control the storage management policy for the installation. When activating an SCDS, you determine which ACDS will hold the active configuration (if you have defined more than one ACDS). The ACDS is shared by each system that is using the same SMS configuration to manage storage. See *source control data set, communications data set*.

address space
One or more unique identifiers assigned to OAM and OTIS sessions. Also the complete range of addresses in memory available to a computer program.

aggregate group
A collection of related data sets and control information that have been pooled to meet a defined backup or recovery strategy. Aggregate group is used with the storage of DASD data, not within an OAM environment.

alphanumeric
The set of characters that contains only 0–9 and uppercase A–Z.

application plan
The control structure produced during the bind process and used by DB2 to process SQL statements during application execution.

ATLDS
See *automated tape library dataserver*.

attribute
A named property of an entity.

automated tape library dataserver (ATLDS)
A device consisting of robotic components, cartridge storage areas, tape subsystems, and controlling hardware and software, together with the set of tape volumes that reside in the library and can be mounted on the library tape drives. Contrast with *manual tape library*. See *tape library*.

automatic class selection (ACS)
A mechanism for assigning Storage Management Subsystem classes and storage groups to data sets.

automatic class selection (ACS) routine
A procedural set of ACS language statements. Based on a set of input variables, the ACS language statements generate the name of a predefined SMS class, or a list of names of predefined storage groups, for a data set.

backup
The first or second backup copy of a primary object in an Object Backup storage group.

base configuration
The part of an SMS configuration that contains general storage management attributes, such as the default management class, default unit, and default device geometry. It also identifies the systems or system groups that an SMS configuration manages.

bind
The process by which the output from the DB2 precompiler is converted to a usable control structure called an application plan. This process is the one during which access paths to the data are selected and some authorization checking is performed.

block
A string of data elements recorded, processed, or transmitted as a unit. The elements can be characters, words, or physical records.

byte stream
A simple sequence of bytes stored in a stream file.

CAF
Call attachment facility. A DB2 attachment facility that allows application programs to connect to and use DB2, used to establish the OAM address space as a user of DB2.

cartridge
See *optical disk cartridge*.

Cartridge System Tape
The base tape cartridge media used with 3480 or 3490 Magnetic Tape Subsystems.

cartridge eject
The act of physically removing a tape cartridge, usually under robot control, by placing it in an output station. The software logically removes the cartridge by deleting or updating the tape volume record in the tape configuration database. For a manual or virtual tape library, the act of logically removing a tape cartridge from the tape library by deleting or updating the tape volume record in the tape configuration database.

cartridge entry
The process of logically adding a tape cartridge to the library by creating or updating the tape volume record in the tape configuration database. The cartridge entry process includes the assignment of the cartridge to a scratch or private category in the library.

CCW
Continuous composite WORM.

CDS
See *Control data set.*

CDS base
Control data set base.

central processing unit (CPU)
The circuitry of a computer that controls the interpretation and execution of instructions. Traditionally, the complete unit was often regarded as the CPU, whereas today the CPU is often a microchip. In either case, the centrality of a processor or processing unit depends on the configuration of the system or network in which it is used.

CFRM
Coupling facility resource management.

channel-to-channel
A method of connecting two computing devices.

CICS
See *Customer Information Control System*.

class transition
A change in an object's management class, storage class, or both when an event occurs that brings about a change in an object's service level or management criteria. Class transition occurs during a storage management cycle.

Classic OAM configuration
An OAM configuration that supports a single OAM instance per system and can be used for both object related processing and tape library related processing. This OAM instance can also be used with other OAM instances on other systems in an OAMplex. This is the original OAM configuration support prior to the introduction of the "multiple OAM configuration" support.

collection
In OAM, a group of objects that typically have similar performance, availability, backup, retention, and class transition characteristics.

commit
: In DB2, to cause all changes that have been made to the database file since the last commitment operation to become permanent, and the records to be unlocked so they are available to other users.

communications data set (COMMDS)
: The primary means of communication among systems governed by a single SMS configuration. The COMMDS is a VSAM linear data set that contains the name of the ACDS and current utilization statistics for each system-managed volume, which helps balance space among systems running SMS. See *active control data set* and *source control data set*.

compaction
: See *improved data recording capability*.

compatibility mode
: The mode of running SMS in which no more than eight names—representing systems, system groups, or both—are supported in the SMS configuration.

complex
: See *SMS complex*.

configuration
: The arrangement of a computer system as defined by the characteristics of its functional units. See *SMS configuration*.

connectivity
: Relationship establishing the eligibility of a given system in an SMS complex to access a VIO, pool, object, object backup, or tape storage group, or the individual volumes within a pool storage group. The relationship can be NOTCON (not connected), indicating eligibility, or any of the following, all of which imply some level of eligibility: ENABLE, QUIALL (quiesce all), QUINEW (quiesce new), DISALL (disable all), DISNEW (disable new).

construct
: One of the following: data class, storage class, management class, storage group, aggregate group, base configuration.

control data set (CDS)
: With respect to SMS, a VSAM linear data set containing configurational, operational, or communication information. SMS uses three types of control data sets: the source control data set (SCDS), the active control data set (ACDS), and the communication data set (COMMDS).

controlling library
: A 3995 model 131, 132, 133, or C3A optical library model containing the control unit for the 3995 Optical Library Dataserver.

coupling facility (CF)
: The hardware that provides high-speed caching, list processing, and locking functions in a Parallel Sysplex.

CPU
: See *central processing unit*.

cross-system coupling facility (XCF)
: A component of z/OS that provides functions to support cooperation between authorized programs running within a sysplex.

CTC
: See *channel-to-channel*.

Customer Information Control System (CICS)
: An IBM licensed program that provides online transaction processing services and management for critical business applications. CICS runs on many IBM and non-IBM platforms (from the desktop to the mainframe) and is used in various types of networks that range in size from a few terminals to many thousands of terminals. The CICS application programming interface (API) enables programmers to port applications among the hardware and software platforms on which CICS is available. Each product in the CICS family can interface with the other products in the CICS family, thus enabling interproduct communication.

DASD
See *Direct access storage device.*

DATABASE 2 (DB2)
A relational database management system.

DATABASE 2 interactive (DB2I)
An interactive relational database management program.

database request module
A data set member created by the DB2 precompiler that contains information about SQL statements. DBRMs are input into the bind process.

data class
A collection of allocation and space attributes, defined by the storage administrator, that are used to create a data set.

data compaction
See *improved data recording capability*.

Data Facility Storage Management Subsystem (DFSMS)
An operating environment that helps automate and centralize the management of storage. To manage storage, SMS provides the storage administrator with control over data class, storage class, management class, storage group, and automatic class selection routine definitions.

Data Facility Storage Management Subsystem data facility product (DFSMSdfp)
A DFSMS functional component and a base element of z/OS that provides functions for storage management, data management, program management, device management, and distributed data access.

DB2
DATABASE 2.

DB2 data sharing
The ability of concurrent DB2 subsystems or application programs to directly access and change the same data while maintaining data integrity.

DB2 data sharing group
A collection of one or more concurrent DB2 subsystems that directly access and change the same data while maintaining data integrity.

DB2I
See *DATABASE 2 interactive*.

DBRM
See *database request module*.

device
This term is used interchangeably with unit. For a disk or tape, a unit on which a volume may be mounted. For example, a tape drive is a device; a tape cartridge is a volume. Device also applies to other types of equipment, such as a card reader or a channel-to-channel (CTC) adapter.

device group
A group of devices that are interchangeable as far as z/OS allocation is concerned. Unless a request is for a specific device name, if one device in a given device group can satisfy a request, any other can also satisfy that request.

device name
This term is used interchangeably with device number, unit number, and unit name. It is the number by which a specific device is known. For example, and installation with two tape drives might assign them device names 181 and 182.

DFSMS
See *Data Facility Storage Management Subsystem.*

DFSMSdfp
A DFSMS functional component or base element of z/OS, that provides functions for storage management, data management, program management, device management, and distributed data access.

DFSMShsm
A DFSMS functional component or base element of z/OS, used for backing up and recovering data, and managing space on volumes in the storage hierarchy.

DFSMSrmm
A DFSMS functional component or base element of z/OS, that manages removable media.

DFSMSdss
A DFSMS functional component or base element of z/OS, used to copy, move, dump, and restore data sets and volumes.

direct access storage device (DASD)
A device in which time is effectively independent of the location of the data.

DISALL (disable all)
Relationship preventing a system from allocating new data sets in a VIO, pool, object, object backup, or tape storage group, or on individual volumes within a pool storage group.

disk
See *optical disk*.

disk sublevel 1
The sublevel in the OAM storage hierarchy containing DB2 object storage tables.

disk sublevel 2
The sublevel in the OAM storage hierarchy containing a file system (NFS or zFS)

drive definition
A set of attributes used to define an optical disk drive as a member of a real optical library or pseudo optical library.

EFMT1
Enterprise Recording Format 1 recording technology.

EFMT2
Enterprise Recording Format 2 recording technology.

EFMT3
Enterprise Recording Format 3 recording technology.

EFMT4
Enterprise Recording Format 4 recording technology.

EEFMT2
Enterprise Encrypted Format 2 recording technology.

EEFMT3
Enterprise Encrypted Format 3 recording technology.

EEFMT4
Enterprise Encrypted Format 4 recording technology.

Environmental Record Editing and Printing program (EREP)
The program that formats and prepares reports from the data contained within the environmental recording data set (ERDS).

EPI
ERDS Physical Identifier.

EREP
See *Environmental Record Editing and Printing program*.

ESCON
Enterprise System Connection.

esoteric unit name
A name used to define a group of devices having similar hardware characteristics, such as TAPE or SYSDA.

expiration
The process by which data sets or objects are identified for deletion because their expiration date or retention period has passed. On DASD, data sets and objects are deleted. On tape, when all data sets have reached their expiration date, the tape volume is available for reuse.

In OAM, when all of the objects on the volume have expired, you can expire tape and optical volumes.

expiration date
The date at which a file is no longer protected against automatic deletion by the system.

expiration processing
The process of inventory management that ensures expired volumes are released and carries out required release actions on those volumes.

free space
The total amount of unused space in a page. The space that is not used to store records or control information is free space.

GB
See *gigabyte*.

generalized trace facility (GTF)
An optional OS/VS service program that records significant system events, such as supervisor calls and start I/O operations, for the purpose of problem determination.

generic unit name
A name assigned to a class of devices with the same geometry (such as 3390).

gigabyte
1 073 741 824 bytes.

GMT
Greenwich Mean Time.

grant
A DB2 process that authorizes users to access data.

GTF
See *generalized trace facility.*

hardware configuration definition (HCD)
An interactive interface in z/OS that enables an installation to define hardware configurations from a single point of control.

HCD
See *hardware configuration definition*.

IARS
See *Initial Access Response Seconds.*

IBM Enterprise Tape Cartridge
A cartridge system tape with increased capacity that can only be used with 3592 Magnetic Tape Subsystems (MEDIA5, MEDIA6, MEDIA7, MEDIA8, MEDIA9, MEDIA10, MEDIA11, MEDIA12, and MEDIA13).

ICF
See *Integrated catalog facility.*

ID
Identification.

identification
In computer security, the process that allows a system to recognize an entity by means of personal, equipment, or organizational characteristics or codes.

IDRC
See *Improved data recording capability.*

image copy
An exact reproduction of all or part of a table space. DB2 provides utilities to make full image copies or incremental image copies.

improved data recording capability (IDRC)
A form of compression used when storing data on tape. This can increase the effective cartridge data capacity and the effective data transfer rate.

IMS
Information Management System.

index
A set of pointers that are logically ordered by the values of a key. Indexes provide quick access to data and can enforce uniqueness on the rows in a DB2 storage table.

Information Management System (IMS)
A transaction and hierarchical database management system that organizes the data in different structures, depending on data type, to optimize storage and retrieval, and to ensure integrity and ease of recovery.

initial access response seconds (IARS)
A parameter specified in the definition of an SMS storage class indicating the desired response time to locate, mount, and prepare a piece of media for data transfer.

initial program load (IPL)
The initialization procedure that causes an operating system to commence operation.

The process by which a configuration image is loaded into storage at the beginning of a work day or after a system malfunction.

The process of leading system programs and preparing a system to run jobs.

Synonymous with system restart, system startup.

installation exit
The means specifically described in an IBM software product's documentation by which an IBM software product may be modified by a customer's system programmers to change or extend the functions of the IBM software product. Such modifications consist of exit routines written to replace one or more existing modules of an IBM software product, or to add one or more modules or subroutines to an IBM software product, for the purpose of modifying (including extending) the functions of the IBM software product.

integrated catalog facility (ICF)
In the Data Facility Product (DFP), a facility that provides for integrated catalog facility catalogs.

Interactive Storage Management Facility (ISMF)
The interactive interface of DFSMS that allows users and storage administrators access to the storage management functions.

Interactive System Productivity Facility (ISPF)
An interactive base for ISMF.

IPL
See *initial program load.*

ISMF
See *Interactive Storage Management Facility*.

ISO
International Organization for Standardization.

ISPF
See *Interactive System Productivity Facility*.

KB
Kilobyte.

kilobyte (KB)
A unit of measure for storage capacity. One kilobyte equals 1024 bytes.

LCS
See *Library Control System*.

Library Control System (LCS)
The component of OAM used in the support of tape libraries that writes and reads objects on optical disk storage, and manipulates the optical volumes on which the objects reside.

library expansion unit
A 3995 model 111, 112, 113, C12, C16, C18, C32, C34, C36, or C38 that connects to a controlling library to expand the capacities of the 3995 Optical Library Dataserver.

magneto-optic (MO) recording
A method of storing information magnetically on optical media, which is sensitive only at high temperatures, while stable at normal temperatures. A laser is used to heat a small spot on the medium for recording. The ability to focus the laser tightly greatly increases the data density over standard magnetic media. MO media are erasable and rewritable.

management class
A named collection of management attributes describing the retention and backup characteristics for a group of data sets, or for a group of objects in an object storage hierarchy. For objects, the described characteristics also include class transition.

manual tape library (MTL)
An installation-defined set of tape drives and the set of volumes that can be mounted on the drives.

max connects
The maximum amount of foreground and background users and TSO/E connections allowed to a DB2 subsystem.

MB
Megabyte.

MEDIA1
IBM Cartridge System Tape.

MEDIA2
IBM Enhanced Capacity Cartridge System Tape.

MEDIA3
IBM High Performance Cartridge Tape.

MEDIA4
IBM Extended High Performance Cartridge Tape.

MEDIA5
IBM Enterprise Tape Cartridge.

MEDIA6
IBM Enterprise WORM Tape Cartridge

MEDIA7
IBM Enterprise Economy Tape Cartridge

MEDIA8
IBM Enterprise Economy WORM Tape Cartridge

MEDIA9
IBM Enterprise Extended Tape Cartridge

MEDIA10
IBM Enterprise Extended WORM Tape Cartridge

MEDIA11
Enterprise Advanced Tape Cartridge

MEDIA12
Enterprise Advanced WORM Tape Cartridge

MEDIA13
Enterprise Advanced Economy Tape Cartridge

media management system
A program that helps you manage removable media. DFSMSrmm is an example of a media management system.

megabyte (MB)
1 048 576 bytes.

MO
Magneto-optic recording technique for optical media.

mount
A host-linked operation that results in a tape cartridge being physically inserted into a tape drive.

MTL
See *manual tape library*.

Multiple OAM configuration
An OAM configuration that supports multiple OAM Object instances and a separate Tape Library instance per system. One or more of the OAM instances can also be used with other OAM instances on other systems in an OAMplex. Refer to "classic OAM configuration" for the original OAM configuration support.

MVS Configuration Program (MVSCP)
A single-step, batch program that defines the input/output configuration to z/OS.

MVSCP
See *MVS configuration program*.

OAM
See *object access method*.

OAM instance
An OAM subsystem and optionally an associated OAM address space. When the multiple OAM configuration support is used, multiple OAM instances can exist on the same system. Otherwise, only one instance of OAM is supported with the classic OAM configuration support.

OAMplex
The concept of connecting instances of OAM to a single XCF (cross-system coupling facility) group to create an OAMplex within the parallel sysplex environment. This includes using DB2 data sharing where the scope of a DB2 data sharing group equals the scope of the OAMplex.

OAM-managed volumes
Optical or tape volumes controlled by the object access method (OAM).

OAM Storage Management Component (OSMC)
Where objects should be stored, manages object movement within the object storage hierarchy, and manages expiration attributes based on the installation storage management policy.

OAM thread isolation support (OTIS)
An OAM subsystem providing OAM-DB2 functions that use a different thread to DB2 than the application program thread and support for removing OAM subsystems from an OAM configuration.

object
A named byte stream having no specific format or record orientation.

object access method (OAM)
A program that provides object storage, object retrieval, and object storage hierarchy management. OAM isolates applications from storage devices, storage management, and storage device hierarchy management.

Object Backup storage group
A type of storage group that contains optical or tape volumes used for backup copies of objects. See *storage group* and *object storage group*.

object directory tables
A collection of DB2 tables that contains information about the objects that have been stored in an SMS Object storage group.

Object Distribution Manager
The application that resides in the image host and provides services to the front-end application hosts for the storage, retrieval, and routing of image objects and coded data.

Object Storage and Retrieval (OSR)
Component of OAM that stores, retrieves, and deletes objects. OSR stores objects in the storage hierarchy and maintains the information about these objects in DB2 databases.

object storage database
A DB2 database that contains an object directory for an Object storage group, a storage table for objects less than or equal to 3980 bytes, and a storage table for objects greater than 3980 bytes.

Object storage group
One or more object storage destinations comprising an OAM storage hierarchy that is managed as a single object storage area. It consists of meta-data in an object directory (DB2 table) and disk sublevel 1 (DB2 table) storage and might also include object storage on disk sublevel 2 (file system), optical, or tape destinations. See *storage group* and *Object Backup storage group*.

object storage hierarchy
A hierarchy consisting of objects stored in DB2 table spaces on DASD, on optical or tape volumes that reside in a library, and on optical or tape volumes that reside on a shelf. See *storage hierarchy.*

object storage tables
A collection of DB2 tables that contain objects.

OCDB
See *OAM configuration database*.

OAM Configuration Database (OCDB)
The optical library table, the library slot table, the optical drive table, the optical volume table, and the tape volume table that reside in a DB2 database and describe the current OAM configuration.

optical disk
A disk that uses laser technology for data storage and retrieval.

optical disk cartridge
A plastic case that protects and contains the optical disk and permits insertion into an optical drive.

optical disk drive
The mechanism used to seek, read, and write data on an optical disk. An optical disk drive may reside in an optical library or as a stand-alone unit.

optical library
A set of optical disk drives and optical disks defined to the source control data set. An optical library can be a real library with the optical drives and optical disks residing within the same storage device, or a pseudo library that consists of operator-accessible drives and shelf-resident optical disks. See *real optical library, pseudo optical library*.

optical volume
One side of a double-sided optical disk.

OSMC
See *OAM Storage Management Component*.

OSR
See *Object Storage and Retrieval*.

OTIS
See *OAM thread isolation support*.

out-of-space condition

- A library is considered to be out-of-space for a storage group when:
  - there are no scratch volumes in the optical library
  - any library-resident volumes for the storage group are full.
- The DB2 object database is considered to be out-of-space when a new row cannot be inserted into the object directory.

OVTOC
Optical Volume Table of Contents.

Parallel Sysplex
A sysplex that uses one or more coupling facilities.

PCA
Parallel channel adapter.

PLT
Program list table.

PPT
Program properties table.

primary
An object that is in the object storage hierarchy and can be retrieved by OSREQ RETRIEVE. There is no connection to the last time the object was used or its actual or expected frequency of use.

private
The state of a tape volume that contains user-written data. A private volume is requested by specifying the volume serial number.

pseudo optical library
A set of shelf-resident optical volumes associated with stand-alone, or operator-accessible, or both, optical disk drives.

QEL
Query element list.

RCT
See *resource control table*.

real optical library
Physical storage device that houses optical disk drives and optical cartridges, and contains a mechanism for moving optical disks between a cartridge storage area and optical disk drives.

recording format
For a tape volume, the format of the data on the tape; for example, 18, 36, 128, 256, 384 tracks, or EFMT1.

Resource Control Table (RCT)
The CICS table that contains customization information for a particular Object Distribution Manager installation.

resource measurement facility (RMF)
An IBM licensed program or optional element of z/OS, that measures selected areas of system activity and presents the data collected in the format of printed reports, system management facilities (SMF) records, or display reports. Use RMF to evaluate system performance and identify reasons for performance problems. These reports provide a snapshot status of OAM's performance at a given time.

rewritable media
Media that can be erased, rewritten, or reused.

RMF
See *resource measurement facility*.

row
The horizontal component of a DB2 table. A row consists of a sequence of values, one for each column of a table.

SCDS
Source control data set.

scratch
The state of a tape volume that is available for general use. A scratch volume is requested by omitting the volume serial number.

scratch pool
The collection of tape cartridges from which requests for scratch tapes can be satisfied.

scratch tape
See *scratch volume*.

scratch volume
A tape volume that has been assigned the scratch use attribute by the software. If the cartridge resides in a tape library, it is assigned to a scratch category of the appropriate media type.

SCSI
See *Small Computer System Interface*.

SDR
See *sustained data rate*.

second backup object
The second backup copy of an object, which is stored in the Object Backup storage group that is specified as a second Object Backup storage group.

sector
On disk storage, an addressable subdivision of a track used to record one block of a program or data.

shelf
A place for storing removable media, such as tape and optical volumes, when they are not being written to or read.

shelf-resident optical volume
An optical volume that resides outside of an optical library.

shelf-resident tape volume
A tape volume that resides outside of a tape library.

slot
A space in a library where a cartridge is stored.

Small computer system interface (SCSI)
A mechanical, electrical, and functional standard for a small computer input/output bus and command sets for peripheral device types commonly used with small computers. **Note:** Laser Magnetic Storage International (LMSI) documentation sometimes uses ICI and ISI interchangeably with SCSI.

SMF
See *system measurement facility*.

SMP/E
See *System Modification Program/Extended*.

SMS
See *Storage Management Subsystem*.

SMS class
A list of attributes that SMS applies to data sets having similar allocation (data class), performance (storage class), or backup and retention (management class) needs.

SMS complex
A collection of systems or system groups that share a common configuration. All systems in an SMS complex share a common active control data set (ACDS) and a communications data set (COMMDS). The systems or system groups that share the configuration are defined to SMS in the SMS base configuration.

SMS-managed data set
A data set that has been assigned a storage class.

SPUFI
See *SQL Processing Using File Input*.

SQL
See *Structured Query Language*.

SQLCODE
Structured query language return code.

SQL Processing Using File Input
Used to perform groups of SQL statements in batch or online mode. SPUFI is option one under DB2I.

stand-alone optical drive
An optical drive housed outside of an optical library.

storage class
A collection of storage attributes that identify performance goals and availability requirements, defined by the storage administrator, used to select a device that can meet those goals and requirements.

storage group
One or more storage destinations for objects such as disk, optical, and tape that is managed as a single storage area. See *storage group*, *Object Backup storage group*, and *tape storage group* for definitions of specific types of storage groups.

storage hierarchy
An arrangement of storage devices with different speeds and capacities. The levels of the storage hierarchy include main storage (memory, DASD cache), primary storage (DASD containing uncompressed data), migration level 1 (DASD containing data in a space-saving format), and migration level 2 (tape cartridges containing data in a space-saving format). See *object storage hierarchy*.

storage management cycle
An invocation of the OAM Storage Management Component (OSMC). The storage management cycle ensures that every object scheduled for processing is placed in the correct level of the object storage hierarchy (as specified by its storage class), is expired or backed up (as specified by its management class or by an explicit application request). If necessary, the object is flagged for action during a subsequent storage management cycle.

Storage Management Subsystem (SMS)
A DFSMS facility used to automate and centralize the management of storage. Using SMS, a storage administrator describes data allocation characteristics, performance and availability goals, backup and retention requirements, and storage requirements to the system through data class, storage class, management class, storage group, and ACS routine definitions.

structured query language (SQL)
A DB2 query tool.

supervisor call (SVC)
A request that serves as the interface into operating system functions, such as allocating storage. The SVC protects the operating system from inappropriate user entry. All operating system requests must be handled by SVCs.

sustained data rate (SDR)
A parameter specified in the definition of an SMS storage class indicating the desired sustained data rate to read the next 4 KB block of a data entity assuming the prior 4 KB block has been read.

SVC
See *supervisor call*.

sysplex
A set of z/OS systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads.

system-managed tape library
A collection of tape volumes and tape devices, defined in the tape configuration database. A system-managed tape library can be automated or manual. See *tape library.*

system measurement facility (SMF)
An optional control program feature that provides the means for gathering and recording information that can be used to evaluate system usage.

System Modification Program/Extended (SMP/E)
Basic tool for installing software changes in programming systems. It controls these changes at the element (module or macro) level, which helps protect system integrity.

table
In DB2, a named data object consisting of a specific number of columns and some number of unordered rows.

table space
A page set used to store the records of one or more DB2 tables.

tape cartridge
A case containing a reel of magnetic tape that can be put into a tape unit without stringing the tape between reels.

tape configuration database (TCDB)
The set of tape library records and tape volume records that reside in ICF volume catalogs and describe the current tape library configuration.

tape library
A set of equipment and facilities that support an installation's tape environment. This can include tape storage racks, a set of tape drives, and a set of related tape volumes mounted on those drives. See *system-managed tape library, automated tape library data server*.

Tape Library Dataserver
A hardware device that maintains the tape inventory that is associated with a set of tape drives. An automated tape library dataserver also manages the mounting, removal, and storage of tapes.

tape storage group
A type of storage group that contains system-managed private tape volumes. The tape storage group definition specifies the system-managed tape libraries that can contain tape volumes. See *storage group*.

tape volume
A tape volume is the recording space on a single tape cartridge or reel. See *volume*.

TB
See *terabyte*.

TCDB
See *tape configuration database*.

Terabyte (TB)
Terabyte as used in data processing (1 099 511 627 776 bytes).

Time Sharing Option Extensions (TSO/E)
An option of the z/OS operating system that provides interactive time sharing from remote terminals.

TSO/E
See *Time Sharing Option/Extensions*.

user exit
A programming service provided by an IBM software product that may be requested by an application program for the service of transferring control back to the application program upon the later occurrence of a user-specified event.

vary offline
To change the status of an optical library or an optical drive from online to offline. Varying a library offline does not affect the online/offline status of the drives it contains. When a library or drive is offline, no data may be accessed on optical disks through the offline drive or the drives in the offline library.

vary online
To change the status of an optical library or an optical drive from offline to online. This makes the drive or drives in the library being varied online available for optical disk access.

virtual tape server (VTS)
A tape subsystem that combines the random access and high performance characteristics of DASD with outboard hierarchical storage management and virtual tape devices and tape volumes.

volume full threshold
When the number of free kilobytes on the volume falls below this threshold, the volume is marked full.

VOLSER
See *volume serial number*.

volume serial number (VOLSER)
An identification number in a volume label that is assigned when a volume is prepared for use on the system. For standard label volumes, the volume serial number is the VOL1 label of the volume. For no label volumes, the volume serial number is the name the user assigns to the volume.

In DFSMSrmm, volume serial numbers do not have to match rack numbers.

VTOC
Volume table of contents.

WORM
See *write-once, read-many*.

write-once, read-many (WORM) media
This type of optical disk or tape media cannot be rewritten nor erased.

XCF
See *cross-system coupling facility*.

Index

B

C

I

M

N

O

P

T

U

SC23-6866-30