



## **z/OS<sup>®</sup> and DFSMS<sup>™</sup>**

*OAM object support and ACS routines*

@business on demand software

© 2008 IBM Corporation

## Agenda

- Overview of OAM object support
- Description of what changes are needed to use tape for OAM object environments
  - ▶ CBROAMxx parmlib entries
  - ▶ ISMF definitions
  - ▶ ACS routines
- Walk through example scenarios

This module starts with an overview of using the object access method (OAM) to manage objects in the storage hierarchy. Then it will discuss what changes are needed in detail for OAM object tape support including storing, backup and OSMC transitioning objects to tape. Then it will walk through some of the examples on how to create/update CBROAMxx parmlib, ISMF definitions and ACS routines.

## OAM object support

- OAM storage hierarchy:
  - ▶ DASD
  - ▶ Optical
  - ▶ Tape
    - Tape sublevel 1 (TSL1)
    - Tape sublevel 2 (TSL2)
- Objects can be written to and read from any level
- Objects can be transitioned between any two levels

The object access method (OAM) is a component of DFSMSdfp, the base for the Storage Management Subsystem (SMS). OAM uses SMS to define a storage hierarchy for objects and the management parameters to place user-accessible objects anywhere in the OAM storage hierarchy.

During the storage management cycle, OAM determines whether the primary copy of an object is correctly positioned in the OAM storage hierarchy. If not, the primary copy of the object is moved to the correct storage medium.

## OAM object tape support

- OAM level2 has received many Customer PMR's with questions on the usage of OAM objects on tape:
  - ▶ How to store OAM objects to tape?
  - ▶ How to write OAM Backups to tape?
  - ▶ How to transition objects from DASD or optical to tape using OSMC?
  - ▶ How to steer OAM tape writes to an IBM automated tape library?
  - ▶ How to steer OAM tape writes to a NON-IBM automated tape library?
  - ▶ How to steer OAM tape writes to a stand-alone tape drive?

Those are the common questions that many customers and level2 people have on how to use OAM objects on tape.

## OAM object tape support

- Getting OAM objects to tape requires an understanding of ISMF, ACS routines, and the CBROAMxx parmlib member
- Changes or updates required to use OAM object tape Support:
  - ▶ Create new CBROAMxx parmlib member to specify OAM tape related options
  - ▶ Update the OAM cataloged procedure in SYS1.PROCLIB to add OAM=xx
  - ▶ Create new ISMF storage class with OAM specifications for IARS, SDR, and OSL for tape
  - ▶ Update the ACS routines for new tape storage class
  - ▶ Incorporate Tape Management System updates

Getting OAM objects to tape needs a good understanding of ISMF library management role within OAM, CBROAMxx parmlib member statements for object tape volumes, and automatic class selection (ACS) routines to determine, validate, or override the existing values that are assigned for the storage group, storage class, management class, and data class constructs used by OAM.

## New CBROAMxx parmlib member

- CBROAMxx parmlib member
  - ▶ Many SETOAM keywords can be specified: retrieve tasks, store tasks, tape expiration, data class, compaction, startup threshold and many more
  - ▶ Required keyword to get objects to go to TSL1:
    - TAPEUNITNAME (storage group level specification)
  - ▶ Required keyword to get objects to go to TSL2:
    - L2TAPEUNITNAME (storage group level specification)

The CBROAMxx parmlib member contains one or more SETOAM statements. Those statements contain keywords that you can use to tailor the object tape function. The CBROAMxx member is processed during OAM address space initialization to establish the tape-related values for the object tape storage. Creating or updating the CBROAMxx parmlib member with SETOAM statements is required to use object tapes within your environment.

TAPEUNITNAME is 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 sublevel 1 (TSL1) or backup volume.

L2TAPEUNITNAME is a required subparameter of the STORAGEGROUP parameter, if you use the tape sublevel 2 function. This keyword specifies 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 sublevel 2 (TSL2) volume.

Note: The L2TAPEUNITNAME keyword cannot be associated with an Object Backup storage group.

## New ISMF storage class

- ISMF storage class specification
  - ▶ For use both with storing to tape and OSMC transition to tape:
    - IARS (Initial Access Response Seconds) of greater than zero
    - SDR (Sustained DATA Rate) of greater than or equal to 3
    - OSL (OAM Sublevel) of 1 or 2

Use Interactive Storage Management Facility (ISMF) to integrate OAM into system-managed storage.

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 for an object. Every object is assigned to a storage class when it is created and this assignment determines where the object initially resides in the object storage hierarchy (optical, DASD, TSL1, and TSL2).

Initial Access Response Seconds (IARS):

- 0 = DASD
- 1-9999 = Removable Media

Sustained Data Rate (SDR):

- 0-2 = Optical
- 3-999 = Tape

OAM Sublevel (OSL):

- 1 = OAM Sublevel 1 (default)
- 2 = OAM Sublevel 2

Example: A storage class defined with IARS=1, SDR=3 and OSL=2 equates to TSL2.

## Modify existing ACS routines

- Add/Modify Storage Class ACS routine
  - ▶ Update to ensure correct storage class is assigned to objects for tape in environments (STORE, CHANGE, and CTRANS)
  - ▶ Update environment ALLOC to assign an SMS storage class if planning on going to an IBM tape library
- Add/Modify Management Class ACS routine
  - ▶ Update to ensure correct mgmt class is assigned to objects for tape in environments (STORE, CHANGE, and CTRANS)

Automatic class selection routines are used to implement the 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. There are four ACS routines: storage class (SC), management class (MC), data class (DC), and storage group (SG), all need to be modified to have OAM tape support.

Storage Class routine: make sure the correct storage class is assigned to the objects for tape in all three ACS environments (STORE, CHANGE, and CTRANS), and an SMS storage class is assigned in environment ALLOC (invoked by MVS during allocation) if you plan to go to an IBM tape library.

Management Class routine: make sure the correct management class is assigned to objects for tape in all three ACS environments (STORE, CHANGE, and CTRANS).



## Modify existing ACS routines

- Add/Modify data class ACS routine
  - ▶ Update to ensure that a DATACLASS is not assigned to OAM Object tape datasets:
    - OAM.PRIMARY.DATA
    - OAM.BACKUP.DATA
    - OAM.BACKUP2.DATA

(DATACLASS association for OAM Objects is done with the SETOAM statement in CBROAMxx Parmlib member)

Data class is an SMS construct that determines the characteristics for a tape volume during scratch tape allocation. If the allocation is steered to an ATLDS or a MTL, the data class subparameter on the SETOAM statement of the CBROAMxx parmlib member is used, if it is specified. Since the ACS routines can also supplement or override the data class values you specify for the tape volume, the recommendation is not to allow the ACS routines to assign or change the data class assignment of an OAM tape volume.

## Modify existing ACS routines

- Add/Modify storage group ACS routine
  - ▶ Ensure that the storage group ACS routine assigns a tape storage group to OAM tape allocations that you want to go to an IBM tape library
  - ▶ For non-IBM tape library or stand-alone devices, this step is not necessary

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 IBM tape library) that is associated with the Tape Storage group. So make sure that SG ACS routines assigns a Tape Storage group to OAM tape allocations if you want to go to an IBM tape library.

## Tape management system updates

- Tape management systems must be notified to ignore OAM datasets:
  - ▶ OAM.PRIMARY.DATA
  - ▶ OAM.BACKUP.DATA
  - ▶ OAM.BACKUP2.DATA
- Obtain special NEVER EXPIRE date for tape management system and then supply that date on a SETOAM statement in the CBROAMxx parmlib
- SETOAM TAPEEXPIRATION(yyyy/ddd)

If the tape volume is a primary volume that belongs to an Object storage group and contains the primary copy of the object, the dataset name of the physical sequential dataset is OAM.PRIMARY.DATA. If the tape volume is a backup volume that belongs to an Object Backup storage group and contains the first backup copy of the object, the dataset name of the physical sequential dataset is OAM.BACKUP.DATA. If the tape volume is a backup volume that belongs to an Object Backup storage group and contains the second backup copy of the object, the dataset name of the physical sequential dataset is OAM.BACKUP2.DATA.

The TAPEEXPIRATION date overrides the expiration date defined in the DATACLASS parameter for the datasets residing on the tape volume. If the specified TAPEEXPIRATION date is the current date or a date preceding the current system date, the datasets are considered previously expired and are therefore eligible for immediate replacement. So check with your tape management system to determine what 'never expire' 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.

## Samples and scenarios

- Following are some samples and scenarios that have been put together to help get a better feel for all that needs to be modified to get tape working for OAM Objects
- The samples are very simplistic, in most customer environments the OAM specifications have to be blended in with other system specifications for ACS routines and ISMF panels

The samples/scenarios in the following slides are given on:

- OAM catalog procedure and SETOAM statements in CBROAMxx
- defining SC, MC, DC, and SG constructs using ISMF for OAM tape
- modifying ACS storage class, management class, data class, and storage group routines,
- writing object/backup to tape

## Sample OAM catalog procedure

```
//OAM   PROC OSMC=YES,MAXS=2,UNLOAD=9999,OAM=XX
//IEFPROC EXEC PGM=CBROAM,REGION=0M,
//PARM=('OSMC=&OSMC,APLAN=CBROAM,MAXS=&MAXS,UNLOAD=&UNLOAD',
//      'OAM=&OAM')
//SYSABEND DD SYSOUT=A
```

- **OAM=xx** specifies the low order suffix of the **CBROAMxx** Parmlib member which OAM is to process during OAM initialization

Add 'OAM=xx' to your OAM catalog procedure in PROCLIB. Make sure the 'xx' is the same as the suffix of CBROAMxx parmlib member which OAM is to process during OAM initialization.

## Sample CBROAMxx parmlib member

```
SETOAM MAXTAPERETRIEVETASKS(4)
      MAXTAPESTORETASKS(4)
      MOUNTWAITTIME(5)
      TAPEEXPIRATION(9999/999)
      STORAGEGROUP(GROUP01
        TAPEUNITNAME(3590)
        DATACLASS(ATL1_DC)
        L2TAPEUNITNAME(3590)
        MAXTAPERETRIEVETASKS(2)
        TAPEPERCENTFULL(95))
      STORAGEGROUP(GROUP02
        TAPEUNITNAME(MYTAPE)
        L2TAPEUNITNAME(MYTAPE))
      STORAGEGROUP(GROUP03
        TAPEUNITNAME(3490E))
      STORAGEGROUP(BACKUP1
        TAPEUNITNAME(3490E))
```

This is a sample SETOAM statements, and is required to use object tapes within your environment. If no SETOAM statements exist in the CBROAMxx, objects can not be stored to tape media.

TAPEUNITNAME is required for a Object or Object Backup storage group if use tape sublevel 1 (TSL1) function for that group, while L2TAPEUNITNAME is required for a Object storage group if use tape sublevel 2 (TSL2) function for the group. Also use L2DATACLASS to specify data class for tape sublevel 2.

L2TAPEUNITNAME can not be associated with the Object Backup storage groups.

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.

## Sample ISMF definitions

- The following samples are used for ISMF
  - ▶ **Storage classes**
    - **DASD\_SC**: put object on DASD
    - **OPTI\_SC**: put object on optical
    - **TSL1\_SC** : put object on tape sublevel 1
    - **TSL2\_SC**: put object on tape sublevel 2
    - **ATL\_SC**: steer allocation into an IBM automated tape library
  - ▶ **Storage group (NON object)**
    - **ATL\_SG**: storage group for an IBM automated tape library
  - ▶ **Data classes**
    - **ATL1\_DC**: steer object to an IBM automated tape library

Define some SC, SG, and DC constructs for some examples used through out this presentation.

## Sample ISMF definitions

- These samples are used for ISMF
  - Management classes
    - DASDO      DASD -> Optical      transition to OPTIT in 90 days
    - DASDT      DASD -> TSL1            transition to TSL1E in 90 days
    - DASDU      DASD -> TSL2            transition to TSL2E in 90 days
    - DASDE      DASD -> DASD            expire in 7 years
    - OPTID      Optical -> DASD          transition to DASDO in 1 year
    - OPTIT      Optical -> TSL1          transition to TSL1E in 1 year
    - OPTIU      Optical -> TSL2          transition to TSL2E in 1 year
    - OPTIE      Optical -> Optical        expire in 7 years
    - TSL1D      TSL1 -> DASD            transition to DASDO in 1 year
    - TSL1O      TSL1 -> Optical          transition to OPTIT in 1 year
    - TSL1U      TSL1 -> TSL2            transition to TSL2E in 1 year
    - TSL1E      TSL1 -> TSL1            expire in 7 years
    - TSL2D      TSL2 -> DASD            transition to DASDO in 1 year
    - TSL2O      TSL2 -> Optical          transition to OPTIU in 1 year
    - TSL2T      TSL2 -> TSL1            transition to TSL1E in 1 year
    - TSL2E      TSL2 -> TSL2            expire in 7 years

Define management classes for some examples used though out this presentation.

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 be result in a variety of actions, such as 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 ACS routines, it is critical to analyze the end result of your class transitions to avoid processing inefficiencies, unexpected results, or both.



## Storage class ISMF definitions

- For DASD\_SC Storage Class
  - ▶ Initial Access Response Seconds (IARS) = 0
  - ▶ Sustained Data Rate (SDR) = no value specified
  - ▶ OAM Sublevel (OSL) = no value specified
- For OPTI\_SC Storage Class
  - ▶ Initial Access Response Seconds (IARS) = 1
  - ▶ Sustained Data Rate (SDR) = 2
  - ▶ OAM Sublevel (OSL) = no value specified
- For TSL1\_SC and ATL\_SC Storage Class
  - ▶ Initial Access Response Seconds (IARS) = 1
  - ▶ Sustained Data Rate (SDR) = 3
  - ▶ OAM Sublevel (OSL) = 1
- For TSL2\_SC Storage Class
  - ▶ Initial Access Response Seconds (IARS) = 1
  - ▶ Sustained Data Rate (SDR) = 3
  - ▶ OAM Sublevel (OSL) = 2

More details on ISMF SC definitions:

Initial Access Response Seconds (IARS): 0 = DASD, 1-9999 = Removable Media

Sustained Data Rate (SDR): 0-2 = Optical, 3-999 = Tape

OAM Sublevel (OSL): 1 = OAM Sublevel 1 (default), 2 = OAM Sublevel 2

## MGMT class ISMF definitions

- For DASDO, DASDT, and DASDU management class specifications:
  - ▶ Expiration attributes = NOLIMIT
  - ▶ AUTO BACKUP = YES|NO
  - ▶ Transition criteria = 90 days since creation
- For OPTID, OPTIT, OPTIU, TSL1D, TSL1O, TSL1U, TSL2D, TSL2O, and TSL2T management class specifications:
  - ▶ Expiration attributes = NOLIMIT
  - ▶ AUTO BACKUP = YES|NO
  - ▶ Transition criteria = 1 year since creation
- For DASDE, OPTIE, TSL1E, and TSL2E management class specifications:
  - ▶ Expiration attributes = 2555 (days since create)
  - ▶ AUTO BACKUP = YES|NO
  - ▶ Transition criteria = none specified

More details on ISMF MC definitions...

Note: There is another parameter BACKUP FREQUENCY associated with MC used for the immediate backup support. If you set AUTO BACKUP=YES and BACKUP FREQUENCY=0, OAM will schedule the first backup copy to be written immediately after the primary copy of the object is successfully stored.

## ACS invocation

- For the OSREQ CHANGE command (&ACSENVIR='CHANGE'):
  - ▶ 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, storage class and then management class ACS routines are invoked, using the requested MC and SC
- For the OSREQ STORE command (&ACSENVIR='STORE'):
  - ▶ When storing the first object into a new collection, the routines for SC, MC, and perhaps SG are invoked
  - ▶ When an explicit MC, SC, or both are specified, the routines for MC or SC are invoked the same way as for OSREQ CHANGE
- For the OSMC cycle (&ACSENVIR='CTRANS'):
  - ▶ Only invoked when MC transition criteria satisfied
  - ▶ The storage class routine and then the management class routine are executed
- For allocation (&ACSENVIR='ALLOC'):
  - ▶ Only invoked by MVS during allocation

The automatic class selection routines can be invoked in four different environments: OSREQ STORE when storing the first object to a new collection or when at least one of the MC/SC specified; OSREQ CHANGE when at least one of the MC/SC specified; OSMC cycle when MC transition criteria satisfied; and tape volume allocation by MVS.

## Sample STORCLAS ACS routine

```
PROC STORCLAS
/* OAM Change environment */
IF &ACSENVIR = 'CHANGE' THEN
DO
  SELECT (&STORCLAS)
  WHEN('DASD_SC'|'OPTI_SC'|'TSL1_SC '|'OPTI_SC')
  /* valid sc */
  DO /* do nothing, leave */
  END /* Storage class as is */
  OTHERWISE
  EXIT CODE(10) /* indicate not valid sc */
END
```

This storage class ACS routine with the CHANGE environment. It validates the storage class specified on a OSREQ CHANGE.

## Sample STORCLAS ACS routine

```

/* OAM store environment */
IF &ACSENVIR = 'STORE' then
DO
  IF &STORCLAS = ('DASD_SC'|'OPTI_SC'|'TSL1_SC'|'TSL2_SC')
    DO /* valid sc, do nothing, leave */
    END /* Storage class as is */
  ELSE
    SELECT (&MGMTCLAS) /* assign sc based on mc */
    WHEN('DASD'|'DASDT'|'DASDU'|'DASDE') /* its a dasd mc */
      SET &STORCLAS = 'DASD_SC' /* set dasd sc */
    WHEN('TSL1D'|'TSL1O'|'TSL1U'|'TSL1E') /* its a tsl1 mc */
      SET &STORCLAS = 'TSL1_SC' /* set tsl1 sc */
    WHEN('TSL2D'|'TSL2O'|'TSL2T'|'TSL2E') /* its a tsl2 mc */
      SET &STORCLAS = 'TSL2_SC' /* set tsl2 sc */
    WHEN('OPTID'|'OPTIT'|'OPTIU'|'OPTIE') /* its a optical mc */
      SET &STORCLAS = 'OPTI_SC' /* set optical sc */
    END /* select */
  END
END

```

This is storage class ACS routine with the STORE environment. It validates the storage class specified or assigns a proper storage class based on management class if the storage class is not specified on a OSREQ STORE.

## Sample STORCLAS ACS routine

```

/* OAM CTRANS environment */
IF &ACSENVIR = 'CTrans' THEN
DO
  SELECT (&MGMTCLAS) /* assign sc based on mc */
    WHEN('DASDO'|'TSL1O'|'TSL2O') /* trans to optical needed */
      SET &STORCLAS = 'OPTI_SC' /* set opti storage class */
    WHEN('TSL1D'|'TSL2D'|'OPTID') /* trans to DASD needed */
      SET &STORCLAS = 'DASD_SC' /* set DASD storage class */
    WHEN('DASDT'|'OPTIT'|'TSL2T') /* trans to tsl1 needed */
      SET &STORCLAS = 'TSL1_SC' /* set tsl1 storage class */
    WHEN('DASDU'|'OPTIU'|'TSL1U') /* trans to tsl2 needed */
      SET &STORCLAS = 'TSL2_SC' /* set tsl2 storage class */
    OTHERWISE /* invalid mc for trans */
      EXIT CODE(11) /* exit with an error */
END

```

This is the storage class ACS routine with CTRANS environment. It assigns a new storage class based on the management class during OSMC cycle when the transition criteria is satisfied.

## Sample STORCLAS ACS routine

```

/* ALLOC environment    (for oam DSN dataclass from SETOAM)    */
IF &ACSENVIR = 'ALLOC' THEN
DO
  IF (&DSN = 'OAM.PRIMARY.DATA') OR
    (&DSN='OAM.BACKUP.DATA') OR
    (&DSN = 'OAM.BACKUP2.DATA') THEN /* IF its an OAM dataset */
  DO
    IF &DATACLAS ^= '' THEN /* dataclass is specified */
      SELECT (&DATACLAS) /* set sc based on dataclas */
      WHEN('ATL1_DC') /* automated tape library dc */
        SET &STORCLAS = 'ATL_SC' /* set tape library storage class */
      OTHERWISE /* invalid dc for alloc */
        EXIT CODE(12) /* exit with an error */
    END
  END
END
END

```

This is the storage class ACS routine with ALLOC environment. It sets a tape storage class if the data class indicates an automated tape library. This is invoked during the MVS allocation of a tape volume.

## Sample MGMTCLAS ACS routine

```
PROC MGMTCLAS
/* OAM CHANGE environment ***** */
IF &ACSENVIR = 'CHANGE' then
DO
  IF &MGMTCLAS = ('DASDO'|'DASDT'|'DASDE'|'TAPEO'|'TAPED'|'TAPEE'|
                 'OPTID'|'OPTIT'|'OPTIE') THEN /* if its valid mc */
    DO /* do nothing, leave */
    END /* mgmt class as is */
  ELSE
    EXIT CODE(4) /* invalid mc */
  END
END
```

This is the management class ACS routine with CHANGE environment. It validates the management class specified on a OSREQ CHANGE.



## Sample MGMTCLAS ACS routine

```

/* OAM STORE environment ***** */
IF &ACSENVIR = 'STORE' THEN
DO
  IF &MGMTCLAS = ('DASDO'|'DASDT'|'DASDU'|'DASDE'|'TSL1O'|'TSL1D'|
    'TSL1U'|'TSL1E'|'TSL2O'|'TSL2D'|'TSL2T'|'TSL2E'|
    'OPTID'|'OPTIT'|'OPTIU'|'OPTIE') THEN /* if its valid MC */
    DO /* do nothing, leave */
    END /* mgmt class as is */
  ELSE
    SELECT (&STORCLAS) /* assign mc based on sc */
    WHEN('DASD_SC') /* its a DASD sc */
      SET &MGMTCLAS = 'DASDE' /* set to DASD forever */
    WHEN('TSL1_SC') /* its a tsl1 sc */
      SET &MGMTCLAS = 'TSL1E' /* set to tsl1 forever */
    WHEN('TSL2_SC') /* its a tsl2 sc */
      SET &MGMTCLAS = 'TSL2E' /* set to tsl2 forever */
    WHEN('OPTI_SC') /* its an optical sc */
      SET &MGMTCLAS = 'OPTIE' /* set to optical forever */
    OTHERWISE
      EXIT CODE(4) /* invalid sc */
  END
END

```

This is the management class ACS routine with STORE environment. It validates the management class specified on a OSREQ STORE or assigns a proper management class based on the storage class if the management class is not specified on the OSREQ STORE request.

## Sample MGMTCLAS ACS routine

```

/* OAM CTRANS environment */
IF &ACSENVIR = 'CTTRANS' then
DO
  SELECT(&MGMTCLAS) /* look at mgmt class */
  WHEN('DASDO'|'TSL1O'|'TSL2O ') /* trans to optical */
    SET &MGMTCLAS = 'OPTIT' /* set optical mc */
  WHEN('OPTIT'|'DASDT'|'TSL2T ') /* trans to tsl1 */
    SET &MGMTCLAS = 'TSL1E' /* set tsl1 expire mc */
  WHEN('OPTIU'|'DASDU'|'TSL1U ') /* trans to tsl2 */
    SET &MGMTCLAS = 'TSL2E' /* set tsl2 expire mc */
  WHEN('TSL1D'|'TSL2D'|'OPTID') /* trans to DASD */
    SET &MGMTCLAS = 'DASDO' /* set DASD mc */
  WHEN('DASDE'|'TSL1E'|'TSL2E'|'OPTIE') /* for expire mc's */
    DO END /* leave mc unchanged */
  OTHERWISE /* invalid mc for trans */
    EXIT CODE(15) /* exit with an error */
END

```

This is the management class ACS routine with CTRANS environment. It sets a new management class based the old management class during a OSMC cycle when transition criteria is satisfied.

## Sample STORGRP ACS routine

```

/* OAM Store environment */
IF &ACSENVIR = 'STORE' then
DO
  SELECT (&DSN(1)) /* Set SG per collection name */
    WHEN (*01) SET &STORGRP = 'GROUP01' /* GROUP01 */
    WHEN (*02) SET &STORGRP = 'GROUP02' /* GROUP02 */
    OTHERWISE
      SET &STORGRP = 'GROUP03' /* all others go to GROUP03 */
END

/* ALLOC environment */
IF &ACSENVIR = 'ALLOC' then
DO
  IF &STORCLAS = 'ATL_SC' THEN /* If you set the atl storage class */
    SET &STORGRP = 'ATL_SG' /* set atl storage group */
END

```

The first one is the storage group ACS routine with STORE environment. It sets the storage group based on collection name.

The second one is the storage group ACS routine with ALLOC environment. It sets the storage group to a tape storage group if the storage class indicates an IBM automated tape library.

## Sample DATACLAS ACS routine

```
PROC DATACLAS
/*****/
/* IF OAM OBJECT TAPE DSN, DO NOT SET THE DATACLASS */
/* (DC set per SETOAM statement in CBROAMxx PARMLIB) */
/*****/
IF (&DSN = 'OAM.PRIMARY.DATA') OR
   (&DSN = 'OAM.BACKUP.DATA') OR
   (&DSN = 'OAM.BACKUP2.DATA') THEN
DO
  EXIT CODE(0)
END
```

This is the data class ACS routine, and it is invoked only by MVS during allocation. Make sure that it does not set any data class in case of OAM object tape so the data class set by SETOAM statement in CBROAMxx parmlib member is not overridden by the ACS routines.

## Scenario: How to write OAM objects to TSL1

- Storage group you are storing to must have a SETOAM TAPEUNITNAME specified in CBROAMxx.
- OAM must be brought up using CBROAMxx parmlib member.
- Tape drives or tape library must be available.
- Object storage class must exist that has correct IARS, SDR, and OSL values for tape TSL1.
- Then just issue a store of an object to a TSL1 storage class

For example,

```
OSREQ STORE GROUP01 GROUP01.OBJECT LENGTH(5000)
      SC(TSL1_SC) MC(TSL1E)
```

This is an example on how to write OAM objects to tape sublevel 1. In this example, TAPEUNITNAME is specified in SETOAM for this object storage group GROUP01, the storage class TSL1\_SC is defined to tape sublevel 1, and the management class TSL1E is defined to stay in TSL1 and not transitioned.

Note: This object is written to tape in an IBM automated tape library because of the SETOAM DATACLASS specification.

## Scenario: How to write OAM objects to TSL2

- Storage group you are storing to must have a SETOAM L2TAPEUNITNAME specified in CBROAMxx.
- OAM must be brought up using CBROAMxx parmlib member.
- Tape drives or tape library must be available.
- Object storage class must exist that has correct IARS, SDR, and OSL values for tape TSL2.
- Then just issue a store of an object to a TSL2 storage class

For example,

```
OSREQ STORE GROUP01 GROUP01.OBJECT LENGTH(5000)
      SC(TSL2_SC) MC(TSL2E)
```

This is an example on how to write OAM objects to tape sublevel 2. In this example, L2TAPEUNITNAME is specified in SETOAM for this object storage group GROUP01, the storage class TSL2\_SC is defined to tape sublevel 2, and the management class TSL2E is defined to stay in TSL2 and not transitioned.

## Scenario: How to write OAM object backups to tape

- Backup Storage group must have a SETOAM TAPEUNITNAME specified in CBROAMxx.
- OAM must be brought up using CBROAMxx parmlib member.
- Tape drives or tape library must be available.
- Then just run OSMC, any object who's MGMT class specified AUTOBACKUP = YES, will have its OAM backups written to tape.

Note: For the Immediate Backup support, any object that has MGMT class specified AUTOBACKUP=YES and BACKUP FREQUENCY=0 (1 is the default value), will have the first backup copy written to tape immediately after the primary copy of the object has been successfully stored.

This is an example on how to write OAM object backups to tape (sublevel 1). Make sure that TAPEUNITNAME is specified in SETOAM for the object backup storage group, and the management class is defined with AUTOBACKUP=YES. The backup copies of the object are written to tape during the OSMC cycle.

Immediate Backup support: if management class is defined with AUTOBACKUP=YES and BACKUP FREQUENCY=0, then the first backup copy is written to tape immediately after the primary object has been successfully stored.

## Scenario: How to transition OAM objects to TSL1

- Store or change an OAM object to a management class that has transition criteria.
- ACS routines must be updated to cause transition to change the objects storage class to a tape TSL1 storage class for environment = 'CTTRANS'.
- Storage group must have a SETOAM TAPEUNITNAME specified in CBROAMxx.
- OAM must be brought up using CBROAMxx parmlib member.
- Tape drives or tape library must be available.
- Then just run OSMC, any object who's MGMT class specifies transition and ACS routines assign a TSL1 storage class, the object will transition to TSL1 with OSMC.

This is an example on how to transition OAM objects to tape sublevel 1.



## Scenario: How to transition OAM objects to TSL2

- Store or change an OAM object to a management class that has transition criteria.
- ACS routines must be updated to cause transition to change the objects storage class to a tape TSL2 storage class for environment = 'CTRANS'.
- Storage group must have a SETOAM L2TAPEUNITNAME specified in CBROAMxx.
- OAM must be brought up using CBROAMxx parmlib member.
- Tape drives or tape library must be available.
- Then just run OSMC, any object who's MGMT class specifies transition and ACS routines assign a TSL2 storage class, the object will transition to TSL2 with OSMC.

This is an example on how to transition OAM objects to tape sublevel 2

## Scenario: Steering OAM tape write to an IBM automated tape library

```
OSREQ STORE GROUP01 GROUP01.OBJECT LENGTH(5000)
      SC(TSL1_SC) MC(TSL1E)
```

Following is the path for this store:

- ▶ ACS routine invoked with ENVIRONMENT = 'STORE'
- ▶ SMS returns storage class information indicating TSL1\_SC
- ▶ SETOAM information obtained by OAM shows GROUP01 goes to 3590 tape drive with DATACLAS of ATL1\_DC
- ▶ OAM issues MVS allocation to allocate 3590 device with DATACLASS specified as ATL1\_DC
- ▶ MVS Allocation now invokes ACS routines with environment of "ALLOC"
- ▶ This Allocation is steered inside the IBM automated tape library

This is an example on how to steer the OAM tape write to an IBM automated tape library. In this example, tape sublevel1 is used for the tape write, DADACLASS is specified as ATL1\_DC in SETOAM for this object storage group indicating an automated tape library data class, the ACS SC routine will set the storage class to ATL\_SC and the ACS SG routine will set the storage group to ATL\_SG as a tape storage group during the MVS allocation, then this allocation is steered to the IBM automated tape library.

If the tape write is using tape sublevel2, make sure L2DATACLASS specified in SETOAM for the object storage group instead of DATACLASS in the above scenario.

## Scenario: Steering OAM tape write to a NON-IBM automated tape library

```
OSREQ STORE GROUP02 GROUP02.OBJECT LENGTH(5000)
      SC(TSL1_SC) MC(TSL1E)
```

Following is the path for this store

- ▶ ACS routine invoked with ENVIRONMENT = 'STORE'
- ▶ SMS returns storage class information indicating TSL1\_SC
- ▶ SETOAM information obtained by OAM shows GROUP02 goes to MYTAPE esoteric with no DATACLASS
- ▶ OAM issues MVS allocation to allocate MYTAPE esoteric device with no data class
- ▶ MVS allocation now invokes ACS routines with environment of "ALLOC"
- ▶ This allocation is steered to tape drives defined in esoteric of MYTAPE which would be defined as drives inside a NON-IBM automated tape library

This is an example on how to steer the OAM tape write to a non-IBM automated tape library. In this example, tape sublevel1 is used for the tape write, TAPEUNITNAME specified in SETOAM for this object storage group is MYTAPE esoteric with no data class specified, the ACS routines get invoked during MVS allocation, and this allocation is steered to tape drives defined in esoteric of MYTAPE inside a non-IBM automated tape library.

If the tape write is for tape sublevel 2, make sure L2TAPEUNITNAME specified as MYTAPE without L2DATACLASS specified in SETOAM for that group.

## Scenario: Steering OAM tape write to a stand-alone tape drive

```
OSREQ STORE GROUP03 GROUP03.OBJECT LENGTH(5000)
      SC(TSL1_SC) MC(TSL1E)
```

Following is the path for this store

- ▶ ACS routine invoked with ENVIRONMENT = 'STORE'
- ▶ SMS returns storage class information indicating TSL1\_SC
- ▶ SETOAM information obtained by OAM shows GROUP03 goes to 3490E device type with no DATACLASS
- ▶ OAM issues MVS allocation to allocate a 3490E device with no data class
- ▶ MVS allocation now invokes ACS routines with environment of "ALLOC"
- ▶ This allocation is steered to a stand-alone 3490E tape drive

This is an example on how to steer the OAM tape write to a stand-alone tape drive. In this example, tape sublevel1 is used for the tape write, TAPEUNITNAME specified in SETOAM for this object storage group is a device type with no DATACLASS specified, then MVS allocation will allocate the specified tape device with no data class, so this allocation is steered to a stand-alone tape drive.

## Scenario: Steering OAM tape write to a stand-alone tape drive

```
OSREQ STORE GROUP01 GROUP01.OBJECT LENGTH(5000)
      SC(TSL2_SC) MC(TSL2E)
```

Following is the path for this store

- ▶ ACS routine invoked with ENVIRONMENT = 'STORE'
- ▶ SMS returns storage class information indicating TSL2\_SC
- ▶ SETOAM information obtained by OAM shows GROUP01 goes to 3590 device type with no L2DATACLASS specified.
- ▶ OAM issues MVS allocation to allocate a 3590 device with no data class.
- ▶ MVS Allocation now invokes ACS routines with environment of "ALLOC".
- ▶ This Allocation is steered to a stand-alone 3590 tape drive.

This is another example on how to steer the OAM tape write to a stand-alone tape drive. In this example, tape sublevel2 is used for the tape write, L2TAPEUNITNAME specified in SETOAM for this object storage group is a device type with no L2DATACLASS specified, then MVS allocation will allocate the specified tape device with no data class, so this allocation is steered to a stand-alone tape drive.

## References

- *z/OS DFSMS V1R9 Object Access Method (OAM) Planning, Installation, and Storage Administration Guide for Object Support*

## Feedback

### Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

[mailto:iea@us.ibm.com?subject=Feedback\\_about\\_V1R0-General-OAMSupport.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_V1R0-General-OAMSupport.ppt)

This module is also available in PDF format at: <..\V1R0-General-OAMSupport.pdf>

You can help improve the quality of IBM Education Assistant content by providing feedback.

## Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

DFSMS IBM z/OS

A current list of other IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. 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 in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2008. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.