IBM Tivoli Advanced Allocation Management for z/OS
Version 3 Release 3

User's Guide

SC23-9817-05

IBM Tivoli Advanced Allocation Management for z/OS
Version 3 Release 3

User's Guide

SC23-9817-05

Note:
Before using this information and the product it supports, read the information in "Notices" on page 385.

This edition applies to Version 3 Release 3 of IBM Tivoli Advanced Allocation Management for z/OS (product number 5697-P35) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC23-9817-04.

© Copyright Rocket Software Inc., 2003 - 2016. All Rights Reserved.

Contents

About this information

IBM® Tivoli® Advanced Allocation Management for z/OS is a tool that helps prevent and reduce processing delays and interruptions that can result from x37 abends and errors that occur because of insufficient disk space during new allocations.

These topics provide instructions for configuring and using Advanced Allocation Management.

These topics are designed to help you perform these tasks:

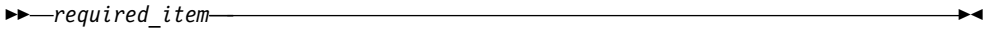
- Operate Advanced Allocation Management
- Customize your Advanced Allocation Management environment
- Diagnose and recover from Advanced Allocation Management problems

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli Information Center Web site at http://www.ibm.com/software/tivoli/library/.

How to read syntax diagrams

The following rules apply to the syntax diagrams that are used in this information:

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line. The following conventions are used:
  - The >>--- symbol indicates the beginning of a syntax diagram.
  - The ---> symbol indicates that the syntax diagram is continued on the next line.
  - The >--- symbol indicates that a syntax diagram is continued from the previous line.
  - The --->< symbol indicates the end of a syntax diagram.
- Required items appear on the horizontal line (the main path).

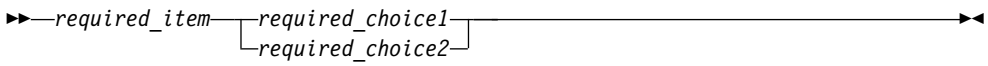


- Optional items appear below the main path.

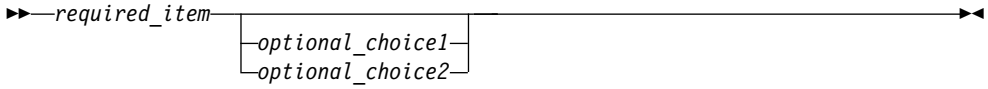
  If an optional item appears above the main path, that item has no effect on the execution of the syntax element and is used only for readability.

- If you can choose from two or more items, they appear vertically, in a stack.

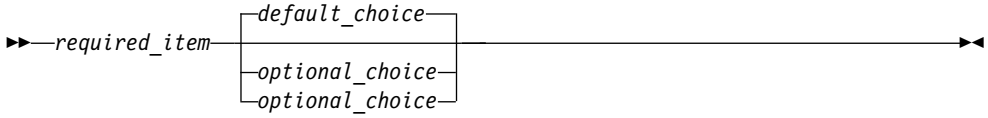
  If you *must* choose one of the items, one item of the stack appears on the main path.



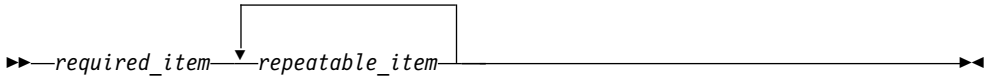
If choosing one of the items is optional, the entire stack appears below the main path.



If one of the items is the default, it appears above the main path, and the remaining choices are shown below.



- An arrow returning to the left, above the main line, indicates an item that can be repeated.



If the repeat arrow contains a comma, you must separate repeated items with a comma.

A repeat arrow above a stack indicates that you can repeat the items in the stack.

- Keywords, and their minimum abbreviations if applicable, appear in uppercase. They must be spelled exactly as shown. Variables appear in all lowercase italic letters (for example, *column-name*). They represent user-supplied names or values.
- Separate keywords and parameters by at least one space if no intervening punctuation is shown in the diagram.
- Enter punctuation marks, parentheses, arithmetic operators, and other symbols exactly as shown in the diagram.
- Footnotes are shown by a number in parentheses; for example, (1).

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other IBM Tivoli Advanced Allocation Management for z/OS documentation, send your comments by e-mail to comments@us.ibm.com. Be sure to include the name of the book, the part number of the book, the version of Advanced Allocation Management, and, if applicable, the specific location of the text that you are commenting on (for example, a page number or table number).

Chapter 1. Advanced Allocation Management overview

Advanced Allocation Management for z/OS® is a tool that helps prevent and reduce processing delays and interruptions that result when x37 (B37, D37, and E37) abends and errors occur because there is insufficient disk space for a new allocation. Using Advanced Allocation Management can help significantly increase productivity and data availability and minimize the downtime that occurs as a result of these conditions.

What's new in Advanced Allocation Management

This topic summarizes the technical changes for this edition.

New and changed information is indicated by a vertical bar (|) to the left of a change. Editorial changes that have no technical significance are not noted.

SC23-9817-05

- Information about Advanced Allocation Management and PDS/Es was clarified. For more information, see "Features and benefits" on page 7.
- Information about sharing rule definitions was added. For more information, see "Sharing rule definitions between instances" on page 22.
- Information about Advanced Allocation Management Extended Address Volume support and z/OS 1.11 and z/OS 1.12 was added. For more information, see "Features and benefits" on page 7.
- The following keyword descriptions were updated: "Bypass integrity check processing (BYPASS_CHECK)" on page 65, "Dynamically add volumes (VOL_ADD)" on page 108, "VOL_ADD_UNIT" on page 118, "VOL_ADD_VGRP" on page 120, and "VOL_ADD_VSER" on page 123.
- The ERASE and SET_ERASE keywords were added. For more information, see "ERASE" on page 198 and "SET_ERASE" on page 140.
- The VOL_ADD_UPG_VOL keyword was added. VOL_ADD_UPG_VOL helps improve performance when accessing the components of a base cluster and its upgrade set by extending components outside of the set of volumes that are already allocated, therefore minimizing contention among components. For more information, see "VOL_ADD_UPG_VOL" on page 119.
- The EATTR selection criteria keyword was added. The EATTR keyword specifies the assigned Extended Attribute value for the data set. z/OS uses the Extended Attribute value to assign eligibility for use of cylinder-managed space on an Extended Address Volume (EAV). In addition, the EATTR keyword of the Data Class is supported in the Data Class Override functionality of Advanced Allocation Management. For more information, see "EATTR" on page 197.
- In support of the EATTR keyword, a new Enhanced Allocation Management function, SET_EATTR, was added. For more information, see "SET_EATTR" on page 139.
- The AOMATCH DD statement identifies a matching Advanced Allocation Management rule definition at the job step level. For more information, see "Identifying a rule definition match" on page 316.
- Output options were added to the Tivoli Advanced Allocation Management Control Member Management panel. For more information, see "Syntax-checking control data set members" on page 277.

- The **Work DS Prefix** field was added to the Tivoli Advanced Allocation Management View Event History panel. For more information, see "Generating event history reports from the ISPF interface" on page 289.
- The PERFORMANCE option description for the AVS_ALGORITHM and VOL_ADD_ALGORITHM keywords was updated. For more information, "AVS_ALGORITHM" on page 58 and "VOL_ADD_ALGORITHM" on page 112.
- Information about VSAM support was updated. For more information, see "Features and benefits" on page 7.
- Information about Advanced Allocation Management processing and specific job names was added. For more information, see "Job names and Advanced Allocation Management processing" on page 18.
- Messages GLO043E, GLO044E, and GLO3220W were added. For more information, see "User interface messages" on page 330.
- Message GLO7003W was added and message GLO7603W was updated. For more information, see "Event recording and reporting messages" on page 374 and "SMF Reporting Utility messages" on page 380.
- Message GLO1108E was added. For more information, see "Subsystem and started task messages" on page 334.
- The explanation for message GLO3139E was updated to add information for VSAM data sets. For more information, see "Function messages (VSAM)" on page 362.
- Messages GLO2128I and GLO3128I were updated to include the total volume count. For more information, see "Function messages (non-VSAM, EAM, and JCL-allocated VSAM)" on page 351 and "Function messages (VSAM)" on page 362.

SC23-9817-04

- Information about coexistence with earlier versions was updated. For more information, see "Coexistence with earlier versions" on page 18.
- Information about Advanced Allocation Management and SORTWK data sets was added. For more information, see "Compatibility with other products" on page 17.
- Information about setting the Z1_INTERCEPT subsystem option was added to Chapter 4, "Configuring Advanced Allocation Management," on page 27. For more information, see "Specifying subsystem options" on page 29.
- The PRIM_REDUCE function description was updated to include information about the SMS Guaranteed Space attribute. Information was also added about PRIM_REDUCE and DSORG and RECORG. For more information, see "Controlled reduction of primary space (PRIM_REDUCE)" on page 85.
- A new BYPASS_CHECK function was added. Use BYPASS_CHECK to bypass integrity checks for multiple concurrent OPENs against a data set and multiple DDs for the same data set within a job step for VOLADD processing. For more information, see "Bypass integrity check processing (BYPASS_CHECK)" on page 65.
- The SMF Reporting Utility was added. The utility generates a Projected Savings Report that you can use to help you evaluate potential Advanced Allocation Management cost savings. See Chapter 11, "SMF Reporting Utility," on page 307 for details.
- The "PSPACE_MB" on page 223 JCL equivalent example was updated.
- New messages GLO2147I, GLO2157I, GLO2158I, GLO3145I, GLO3147I, GLO3160I, and GLO3161I were added. For more information, see "Function

messages (non-VSAM, EAM, and JCL-allocated VSAM)" on page 351 and "Function messages (VSAM)" on page 362.

- New messages were added for the SMF Reporting Utility. For more information, see "SMF Reporting Utility messages" on page 380.
- Information about how to gather diagnostic information was added to the Chapter 13, "Troubleshooting," on page 315 chapter. For more information, see "Gathering diagnostic information" on page 317.

SC23-9817-03

- The ability to create user-defined variables was added. User-defined variables are defined in the variables definitions member (VARDEFS). The variables can be specified on the following rule definition keywords and the volume group definition keyword VOLUMES:
  - VOLSER_FIRST
  - VOLSER_CURRENT
  - VOLSER_ANY
  - UNIT_NAME
  - USERID
  - DATACLAS
  - MGMTCLAS
  - STORCLAS
  - STORGRP
  - DDNAME
  - PROGRAM
  - DSNAME
  - DSNAME_QUALIFIER
  - JOBNAME
  - VOL_ADD_VGRP
  - VOL_ADD_VSER
  - VOL_ADD_UNIT
  - AVS_VGRP
  - AVS_VSER
  - AVS_UNIT

  For more information, see "Creating user-defined variable definitions" on page 271.
- Information about migrating to Advanced Allocation Management was updated. For more information, see Chapter 3, "Migrating to Advanced Allocation Management," on page 25.
- The configuration steps were updated for user-defined variables. For more information, see Chapter 4, "Configuring Advanced Allocation Management," on page 27.
- Information about the control data set and control member management options was updated for user-defined variables. For more information, see "Components" on page 10.
- Information about comparison operands was updated. For more information, see "Comparison operands and user-defined variables" on page 262 and "Comparison operands and lists" on page 262.

- Volume group definition syntax was updated. For more information, see "Volume group definition syntax" on page 268.
- Information about subsystem options was updated. For more information, see "Viewing and updating subsystem options" on page 278 and "Specifying subsystem options" on page 29.
- Command descriptions were updated. For more information, see "Display commands" on page 319 and "Dynamic configuration commands" on page 320.
- New messages were added to support user-defined variables. For more information, see "Subsystem and started task messages" on page 334.
- More examples were added to "Creating rule definitions" on page 45.

SC23-9817-02

- Information about how Advanced Allocation Management handles IBM DB2® data sets was added. See "Features and benefits" on page 7 for details.
- The following selection criteria keywords were enhanced to support multiple operands:
  - DATACLAS
  - DDNAME
  - DSNAME
  - DSNAME_QUALIFIER
  - DSORG
  - JOBNAME
  - MGMTCLAS
  - STORCLAS
  - STORGRP
  - PROGRAM
  - USERID
- The following selection criteria keywords were enhanced to support comparison operators:
  - DSORG (EQ or NE)
  - VOLSER_ANY
  - VOLSER_CURRENT
  - VOLSER_FIRST
- A new DSN_PATTERN_MATCHING subsystem option was added. For more information, see "Specifying subsystem options" on page 29 and "Data set name filtering" on page 258.
- The PQTY and SET_PQTY keyword descriptions were updated. For more information, see "PQTY" on page 219 and "SET_PQTY" on page 155.
- New messages, GLO041E, GLO2194I, GLO1376I, GLO1378E, GLO1123W, and GLO3194I were added. For more information, see "User interface messages" on page 330, "Subsystem and started task messages" on page 334, "Function messages (non-VSAM, EAM, and JCL-allocated VSAM)" on page 351, and "Function messages (VSAM)" on page 362.
- The following rule definitions keywords were changed to specify a minimum value of 1:
  - SET_NUMVOLS
  - SET_NUMVOLS_DATA
  - SET_NUMVOLS_INDEX

  - SET_VOLCT
  - NUMVOLS
  - NUMVOLS_DATA
  - NUMVOLS_INDEX
  - VOLCT

  For more information, see "Enhanced Allocation Management functions" on page 126 and "Selection criteria keywords" on page 180.
- Information about coexistence between Advanced Allocation Management V3.2 and earlier versions of Advanced Allocation Management or Allocation Optimizer, was added. For more information, see "Coexistence with earlier versions" on page 18.
- Information about comparison operands was updated. For more information, see "Comparison operands and lists" on page 262.
- Advanced Allocation Management must be installed on a system with z/OS 1.11 and later.

SC23-9817-01

- The NOTCAT2_DYNAL keyword was added to allow selective enablement of processing of dynamically allocated data sets for the NOTCAT2 function. For more information, see "NOTCAT2_DYNAL" on page 80.
- The NOTCAT2 function was updated. See "Process NOT CATLGD 2 errors (NOTCAT2)" on page 75 for details.
- Information about Advanced Allocation Management processing was added to the VOL_ADD function. For more information, see "Dynamically add volumes (VOL_ADD)" on page 108.
- The SPACE_RELEASE function description was clarified. For more information, see "Recover unused disk space (SPACE_RELEASE)" on page 105.
- The TERM_ALLOC function description was updated. For more information, see "Terminate an allocation (TERM_ALLOC)" on page 107.
- The MOUNT_USAGE function description was clarified. For more information, see "MOUNT_USAGE" on page 214.
- The description of the PRIM_REDUCE function was updated. For more information, see "Controlled reduction of primary space (PRIM_REDUCE)" on page 85.
- Information was added to the following Enhanced Allocation Management function keyword descriptions:
  - SET_DATACLAS
  - SET_MGMTCLAS
  - SET_STORCLAS
  - SET_STORGRP

  For more information, see "Enhanced Allocation Management functions" on page 126.
- Information was added to the following selection criteria keyword descriptions:
  - DEVICECLASS
  - GUARSPACE
  - MOUNT_USAGE
  - RELEASESPACE
  - UNIT_TYPE

  For more information, see "Selection criteria keywords" on page 180.

- Information was added to describe the processing order of the SEC_* functions. For more information, see “Add a secondary allocation (SEC_ALLOC)” on page 93, “Increase the size of secondary allocation (SEC_INC)” on page 98, “Recover from secondary allocation space failures (SEC_REDUCE)” on page 102, and “Reduce secondary allocation to largest available free extent (SEC_BEST)” on page 95.
- Default information was added for the following function keywords:
  - AVS_UNIT
  - AVS_VGRP
  - AVS_VSER
  - NOTCAT2_QUALIFIER
  - NOTCAT2_SUFFIX
  - SPACE_RELEASE
  - VOL_ADD_UNIT
  - VOL_ADD_VGRP
  - VOL_ADD_VSER

  For more information, see “Functions” on page 53.
- More examples were added to the MSG_LEVEL keyword description. For more information, see “MSG_LEVEL” on page 73.
- More examples were added to the topic “Wildcard characters” on page 257.
- The descriptions for the SEC_ALLOC and SEC_INC function keywords were clarified. For more information, see “Add a secondary allocation (SEC_ALLOC)” on page 93 and “Increase the size of secondary allocation (SEC_INC)” on page 98.
- The descriptions for the VOLCT, SET_VOLCT, NUMVOLS, and SET_NUMVOLS keywords were updated. For more information, see “Functions” on page 53 and “Selection criteria keywords” on page 180.
- Message GLO2197E was added. See “Function messages (non-VSAM, EAM, and JCL-allocated VSAM)” on page 351 for details.
- The message number for message GLO2150I (`GLO2150I VOLSER(S): VVVVVV`) was changed to GLO2156I. Message number GLO3150I was changed to GLO3156I.

SC23-9817-00

- As of version 3.1, IBM Tivoli Allocation Optimizer for z/OS was renamed to IBM Tivoli Advanced Allocation Management for z/OS.
- Advanced Allocation Management must be installed on a system with z/OS 1.8 and later.
- Selection criteria keywords were added and updated. For more information, see “Selection criteria keywords” on page 180.
- Support for Enhanced Allocation Management was added. For more information, see “Enhanced Allocation Management functions” on page 126.
- Advanced volume selection capabilities were added. For more information, see “Advanced Volume Selection (AVS)” on page 57.
- Information was added to the GLO1036E message. For more information, see “Subsystem and started task messages” on page 334.
- The GLOJIVP1 member steps and example output were updated. For more information, see “Verifying the installation” on page 34 and Table 3 on page 34.

What does Advanced Allocation Management do?

Advanced Allocation Management helps you avoid and recover from x37 type abends, which occur because of storage management issues such as insufficient disk space allocations and unavailable DASD extents.

Advanced Allocation Management also helps prevent JOB FAILED JCL errors that occur because of insufficient disk space during new allocations. Preventing and reducing the processing delays and interruptions that result from these abends and errors is important because they can negatively affect productivity and impede data availability.

In addition, Advanced Allocation Management provides the following features:

- Volume group support to define DASD groups from which to select additional volumes, as needed
- Volume selection for non-SMS managed data sets
- Centralized control of allocation attribute usage
- Extensive event recording and reporting to help you evaluate product benefits and view specific events and subsequent actions that were taken for a particular job or time frame, and so on
- Online reporting that provides a statistical summary of proactive and reactive product actions

Features and benefits

Advanced Allocation Management offers many features to help you get the most out of your z/OS environment.

x37 abend recovery

To help to recover from x37 type abends, Advanced Allocation Management performs the following tasks:

- Controls reduction of primary space to satisfy allocation
- Adds volumes dynamically at end of volume processing as required
- Recovers from secondary allocation space failures by reducing the allocation amount to a value that is consistent with the current free space on the volume
- Increases PDS directory size dynamically, if exhausted

x37 abend prevention and avoidance

To help prevent and avoid x37 type abends, Advanced Allocation Management performs the following tasks:

- Automatically adds a secondary allocation if one is not specified
- Reduces secondary allocation to the largest available free extent
- Increases the size of the secondary allocation as data set extends occur after a particular extent and at a particular volume number
- Recovers unused disk space by automatically releasing allocated, unused, data set space at data set close

Enhanced Allocation Management

To centralize the control of allocation attribute usage, you can perform the following actions:

- Set allocation attributes that are not specified, or override allocation attributes that do not meet your standards
- Control a wide variety of allocation attributes such as SMS-managed, non-SMS-managed, UNITNAME, UNITTYPE, VOLSER, DATACLAS, STORCLAS, MGMTCLAS, RECFM, LRECL, BLKSIZE (to force system-determined blksize), DISPOSITION (allows forcing catalog control), SPACE parameters, expiration date, and retention period values
- Fail jobs that specify installation-restricted allocation values
- Cause SMS DATACLAS values to override values that are specified through JCL, DEFINE statements or dynamic allocation

Volume group support

Using volume group support, you can define DASD groups from which to select additional volumes, as needed.

Supported data set types

Advanced Allocation Management supports the following data set types:

- VSAM and non-VSAM data sets
- SMS and non-SMS-managed data sets
- Physical sequential
- Partitioned
- Striped data sets
- Dynamically allocated data sets

Note:

- Only DASD data sets are supported. Advanced Allocation Management does not support hierarchical file system (HFS) or zSeries file system (ZFS) data sets.
- Advanced Allocation Management does not process space abends for virtual input/output (VIO) data sets.
- Because DB2 manages the space for its own data sets, Advanced Allocation Management automatically bypasses processing for DB2 allocated data sets. Bypassing processing eliminates the creation of unnecessary control blocks and therefore, eliminates the CPU time that is associated with managing those allocations. For more information, see "Job names and Advanced Allocation Management processing" on page 18.
- Advanced Allocation Management intervenes if an error is encountered at allocation time such as when a primary allocation request is too high. However, it does not perform any other space abend prevention or recovery actions for PDS/E data sets.

Advanced Allocation Management supports the LARGE data set format, which allows the specification of more than 65535 tracks for the primary and secondary allocation for non-VSAM data sets that are specified with a DSNTYPE of LARGE.

VSAM support

Advanced Allocation Management VSAM support includes the following items:

- VSAM data sets (KSDS, ESDS, RRDS, VRRDS, and LDS)

- Alternate indexes
- Index and data components of clusters, where applicable

Note: Only integrated catalog facility (ICF) data sets (SMS and non-SMS managed) are supported.

Advanced Allocation Management does not support VSAM record-level-sharing (RLS) access mode.

Extended Address Volume support

Advanced Allocation Management supports z/OS 1.10 Extended Address Volume (EAV) functionality for VSAM data sets. Preference for allocation in the Extended Addressing Space (cylinder-managed space) is based on the size of the allocation as compared to the Breakpoint Value (BPV) that is defined at the system level or within an SMS storage group. When Advanced Allocation Management determines the eligibility of a candidate volume for advanced volume selection (AVS) or VOL_ADD processing, it uses this preference, which is based on the BPV, within each of the selection algorithms as appropriate. Similarly, the preference is considered when a secondary allocation is reduced as part of SEC_REDUCE, SEC_BEST, PRIM_REDUCE on non-first volumes.

Note: To be eligible for allocation in the cylinder-managed space of an EAV volume, a VSAM data set must contain a control area size of 1, 3, 5, 7, 9, or 15. Data sets that have the IMBED or KEYRANGE attributes defined are not eligible for allocation.

Advanced Allocation Management supports the EAV functionality that was added in z/OS 1.11 for Extended Format Sequential data sets and the EAV functionality that was added in z/OS 1.12 for partitioned data sets, partitioned data sets extended, direct access (DA) data set organization, large format sequential data sets, basic format sequential data sets, and data sets of undefined data set organization. The EATTR Extended Attributes data set option is also recognized by the product and can be used as a selection criterion. The Enhanced Allocation Management function SET_EATTR supports the EATTR keyword. For more information, see “SET_EATTR” on page 139.

Event recording and reporting

To help you evaluate product benefits, specific events that Advanced Allocation Management encountered, and subsequent actions that were taken for a particular job or time frame, and so on, Advanced Allocation Management provides an extensive event recording and reporting facility.

To view event history, use the online user interface or the batch reporting utility. To create customized reporting programs to suit your needs, see Chapter 10, “Event recording and reporting,” on page 287.

Integration with the Tivoli Enterprise Portal (TEP)

Integration with IBM Tivoli OMEGAMON® XE for Storage through TEP delivers seamless integration with related storage management capability and helps you monitor the effect of allocation abends that are managed by Advanced Allocation Management. You can view information and be alerted to situations from a single GUI through integration with TEP.

Extensive and flexible rule definitions

Extensive rule definitions give you the flexibility to choose the combination of selection criteria and functions that best suit your needs.

User-defined variables

To further customize Advanced Allocation Management processing, you can specify user-defined variables on select keywords.

Online user interface

You can access Advanced Allocation Management through an online user interface. Authorized users can issue operator commands to perform selected operations. For more information, see Chapter 8, "Managing control data set members and subsystems," on page 275.

Operator commands

Authorized users can issue operator commands to view and update Advanced Allocation Management subsystem information, activate rule and volume group definitions, and enable or disable specific product functions such as product-level intercepts, from an operator console.

User-specified messaging

Advanced Allocation Management provides user-specified messaging for specific actions that were taken. You can direct the product messages to the JES SYSMSG data set, the operator console, or both.

SMF Reporting Utility

To help you evaluate potential Advanced Allocation Management cost savings, the SMF Reporting Utility generates a Projected Savings Report. For more information, see Chapter 11, "SMF Reporting Utility," on page 307.

Components and architecture

Advanced Allocation Management runs as a started task with a user-designated subsystem. You use a control data set to define where and how Advanced Allocation Management operates. The control data set contains PDS members that govern specific aspects of Advanced Allocation Management processing.

Components

Advanced Allocation Management consists of the following components.

Control data set

To determine how to handle a potential x37 abend or other similar error condition, Advanced Allocation Management employs user-defined processing definitions that reside in the Advanced Allocation Management control data set. The control data set is a partitioned data set. Each PDS member in the control data set contains a specific set of definitions that determine how Advanced Allocation Management performs a particular type of processing. A control data set can exist for an active or inactive Advanced Allocation Management subsystem.

Note:

- Multiple control data sets are permitted.
- The control data set requires fixed-format records that have a record length of 80 bytes.

The control data set and the following default members are generated during the installation and configuration process:

Subsystem options member (OPTIONS)

The OPTIONS member contains the global definitions and general operation options that determine how Advanced Allocation Management functions. The OPTIONS member contains the unique subsystem ID that is associated with a specific instance of Advanced Allocation Management. The subsystem ID is displayed on product panels and in messages.

You can define subsystem options when you customize a specific instance of Advanced Allocation Management. Authorized users can modify subsystem options such as the current rule definitions or volume group member, by using the options that are available on the Tivoli Advanced Allocation Management Subsystem Settings panel or by issuing operator commands.

Note: After you define the subsystem options, you rarely need to modify them unless you are adding or removing an Advanced Allocation Management subsystem.

Rule definitions member (RULEDEFS)

Advanced Allocation Management uses rule definitions to determine whether to take action when an x37 type abend or other out-of-space condition is about to occur. The RULEDEFS member contains *selection criteria* and *functions*. Selection criteria specify the job names, data set names, and so on to include or exclude from Advanced Allocation Management processing. Functions specify the actions (controlling reduction of primary space, adding a secondary allocation, and so on) that you want Advanced Allocation Management to perform using the selection criteria.

Note: By default, Advanced Allocation Management only includes items in its processing when specifically directed to do so.

Volume group definitions member (VGRPDEFS)

The VGRPDEFS member contains the volume group list that defines a group of volumes from which Advanced Allocation Management can select additional volumes as directed by the dynamic volume addition function.

Note: Volume groups are used only for non-SMS data sets. For SMS data sets, SMS is used to select a volume that is based on SMS constructs.

The dynamic volume addition function, provided with the Advanced Allocation Management rule definition syntax, helps avoid out-of-space conditions that occur during end-of-volume processing. During end-of-volume processing, the function dynamically adds another volume, from user-specified volume groups, when the secondary allocation space request cannot be satisfied on the current volume.

Volume group definitions enable you to specify the volume groups from which Advanced Allocation Management selects the additional volumes. Define the volume serial numbers either specifically, or generically, using a pattern masking facility. The volume addition function enables you to specify in any combination: a volume group, an esoteric name, or generic names.

Variable definitions member (VARDEFS)
The VARDEFS member contains user-defined variable definitions that further customize Advanced Allocation Management processing.

The variable definitions are used as substitution values that can be specified on select RULEDEF and VGRPDEF keywords When specified, they are incorporated in the selection criteria for the specified rule or volume definitions at the time that the RULEDEFS and VGRPDEFS are activated.

Started task
The Advanced Allocation Management started task JCL references the Advanced Allocation Management control data set subsystem OPTIONS member which contains the unique subsystem ID that is associated with a specific Advanced Allocation Management subsystem.

Advanced Allocation Management subsystem
Each instance of Advanced Allocation Management is considered to be a separate Advanced Allocation Management subsystem. Therefore, each instance must have a unique subsystem ID and started task JCL to differentiate it from other Advanced Allocation Management subsystems.

Note: The Advanced Allocation Management subsystem is not a true MVS™ subsystem and therefore does not require additional setup in z/OS. The Advanced Allocation Management identifier is used by Advanced Allocation Management to communicate with its component parts, particularly when a second instance of Advanced Allocation Management is active on the system.

User interface
The Advanced Allocation Management user interface consists of a series of ISPF panels from which you can manage Advanced Allocation Management control data set members, view and update (if authorized) Advanced Allocation Management subsystem settings, or view recovery statistics for an active Advanced Allocation Management subsystem.

Architecture and processing

Advanced Allocation Management operates at the system level to intercept x37 type abends and other errors that are caused by insufficient disk space allocations, unavailable DASD extents, and out-of-space conditions.

After you install and configure Advanced Allocation Management, determine what you want to recover in the event of an x37 type abend or other similar error and what Advanced Allocation Management should do if it encounters a particular error condition that involves the selected items.

With the extensive and flexible rule definitions, you can specify combinations of selection criteria (the items you want Advanced Allocation Management to include

or exclude from processing) and functions (the actions you want Advanced Allocation Management to perform when it detects an item that is specified by the selection criteria).

To help avoid out-of-space conditions that occur during end-of-volume processing, use the Advanced Allocation Management dynamic volume addition function and volume group support to define specific volume groups from which Advanced Allocation Management selects additional volumes, as needed.

After you initiate the Advanced Allocation Management started task, and customize and activate the appropriate definitions, no additional user intervention is required. When Advanced Allocation Management is running, it interacts with your operating system to determine when an x37 type abend or other similar error is about to occur, and manages the situation according to your specifications.

Note: Advanced Allocation Management does not attempt to prevent and recover from all situations; only the situations that you specify using Advanced Allocation Management definitions.

The following figure shows an example of a single instance of Advanced Allocation Management on a single image.

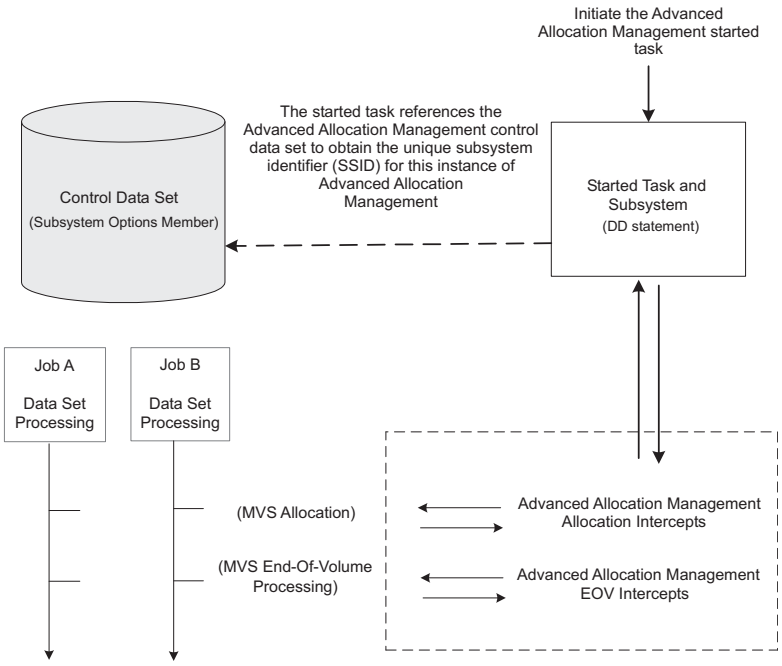


Figure 1. Advanced Allocation Management (single instance)

When the started task is initiated, it references the data in the subsystem options member to establish global settings, including the subsystem identifier for this specific instance of the product. The subsystem ID identifies this specific instance of Advanced Allocation Management within your environment. The started task then starts the Advanced Allocation Management subsystem and the product begins the initialization process. During the initialization process, the control blocks are built, the product modules are loaded, and other product files are read and processed.

After Advanced Allocation Management is running, its allocation and end-of-volume processing intercepts interact with your operating system to

determine when an x37 type abend or other similar error is about to occur. When Advanced Allocation Management detects a potential error condition, it manages the situation according to the product definitions that you specified in the control data set.

Note: An Advanced Allocation Management subsystem can be active or inactive. An Advanced Allocation Management subsystem is considered to be active when the Advanced Allocation Management started task that is associated with that subsystem ID is running and the product-level intercepts are enabled.

Multiple instances of Advanced Allocation Management can run concurrently on a single MVS image. Each instance of the product is considered to be a separate Advanced Allocation Management subsystem and therefore, must have a unique subsystem ID and started task JCL to differentiate it from other Advanced Allocation Management subsystems that might be running on the same image. For more information, see "Multiple instances of Advanced Allocation Management" on page 18.

Service updates and support information

Service updates and support information for this product, including software fix packs, PTFs, Frequently Asked Question (FAQs), technical notes, troubleshooting information, and downloads, are available from the web.

To find service updates and support information, including software fix packs, PTFs, Frequently Asked Question (FAQs), technical notes, troubleshooting information, and downloads, refer to the following web page:

http://www.ibm.com/software/sysmgmt/products/support/

The Advanced Allocation Management Library Web page provides current product documentation that you can view, print, and download. To locate publications with the most up-to-date information, refer to the following web page:

http://www.ibm.com/software/tivoli/sw-library/

Accessibility features

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use a software product successfully.

The major accessibility features in Advanced Allocation Management enable users to:

- Use assistive technologies such as screen readers and screen magnifier software. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.
- Customize display attributes such as color, contrast, and font size.
- Operate specific or equivalent features by using only the keyboard. Refer to the following publications for information about accessing ISPF interfaces:
  - *z/OS ISPF User's Guide, Volume 1*
  - *z/OS TSO/E Primer*
  - *z/OS TSO/E User's Guide*

These guides describe how to use ISPF, including the use of keyboard shortcuts or function keys (PF keys), include the default settings for the PF keys, and explain how to modify their functions.

Chapter 2. Implementing Advanced Allocation Management

When you implement Advanced Allocation Management, you determine if you want to run a single instance or multiple instances of the product and whether the instances will share a control data set or whether each instance will use a separate control data set.

To implement Advanced Allocation Management, complete the following steps:

1. Review the information about compatibility with other products. For more information, see "Compatibility with other products."
2. If you have an earlier version of Advanced Allocation Management or IBM Tivoli Allocation Optimizer for z/OS, and you want to install Advanced Allocation Management 3.3, see "Coexistence with earlier versions" on page 18.
3. Decide whether to run a single instance or multiple instances of Advanced Allocation Management.

   Implementing multiple instances of Advanced Allocation Management in test and production environments is useful when you first install the product or when you want to reserve one subsystem for testing purposes and another subsystem for production purposes. For more information, see "Multiple instances of Advanced Allocation Management" on page 18.
4. If you want to run multiple instances of the product, decide whether the instances will share a control data set or whether each instance will use a separate control data set. For more information, see "Separate control data sets for multiple instances" on page 20 and "One shared control data set for multiple instances" on page 21.
5. Review the guidelines for implementing the product. For more information, see "Guidelines for implementing Advanced Allocation Management" on page 23.

Compatibility with other products

To help prevent problems when you use Advanced Allocation Management, review the following compatibility information.

Products that have similar functionality

Do not use products that have similar functionality to Advanced Allocation Management because unpredictable results might occur.

DFSORT

The Advanced Allocation Management VOL_ADD function cannot be specified for SORTWK data sets. You cannot extend SORTWK data sets to another volume. SORTWK data sets can be on a single volume only. If you need more space, use the SEC_INC function, or add more SORTWK data sets.

For information about DFSORT, see *z/OS DFSORT: Getting Started* and *z/OS DFSORT Application Programming Guide*.

Coexistence with earlier versions

If you have an earlier version of Advanced Allocation Management or the IBM Tivoli Allocation Optimizer for z/OS product, and you are also installing Advanced Allocation Management 3.3, the order in which you start the products is important.

Running version 3.3 with versions earlier than version 3.2 (or IBM Tivoli Allocation Optimizer for z/OS)

If you are running Advanced Allocation Management 3.3 with an earlier version or with IBM Tivoli Allocation Optimizer for z/OS, follow these instructions for starting the programs:

- Perform an IPL, and then start Allocation Optimizer 2.1 or Advanced Allocation Management 3.1. Wait until initialization is complete, and then start Advanced Allocation Management 3.3.
- Start a version of Allocation Optimizer that is earlier than 2.1. Perform an IPL and then start Advanced Allocation Management 3.3.
- Perform an IPL and then start Advanced Allocation Management 3.3. Perform an IPL and then start Advanced Allocation Management 3.1 or Allocation Optimizer 2.1.

Running version 3.3 and version 3.2

If you are running Advanced Allocation Management 3.2 and 3.3, the order in which you start the products is not important.

Job names and Advanced Allocation Management processing

Advanced Allocation Management does not process jobs that end in specific character strings.

Because DB2 manages its own space and DB2 regions typically use the following character strings for the last part of a job name, Advanced Allocation Management bypasses processing for jobs that end in the following strings:

- DBM1
- MSTR
- DIST
- IRLM

Multiple instances of Advanced Allocation Management

Multiple instances of Advanced Allocation Management can run concurrently on the same image or on different images. A maximum of four concurrently active subsystems on a single MVS system are permitted.

Multiple instances of Advanced Allocation Management on separate MVS images

To run multiple instances of Advanced Allocation Management on separate MVS images (one instance per image), specify the same or a different subsystem ID and started task name for each instance. You do not need to specify a unique subsystem identifier and started task name for each instance because each instance is on a different image.

For example, to run two instances of Advanced Allocation Management (one instance on image A and one instance on image B), you can define each subsystem identifier as OPT1 and each started task as ALLOCPT1.

Multiple instances of Advanced Allocation Management on the same MVS image

To run multiple instances of Advanced Allocation Management on the same MVS image, each instance must have a unique subsystem identifier and started task name. For example, if the Advanced Allocation Management subsystem OPT1 runs on image B and was initiated from the started task ALLOCPT1, specify a different subsystem ID and started task (such as subsystem ID OPT2 and started task ALLOCPT2) to initiate a second instance.

Note: If you concurrently run multiple Advanced Allocation Management subsystems on the same MVS image, the subsystem that was most recently initiated is the subsystem processing that takes precedence. For example, if subsystems OPT1 and OPT2 are running on image A and the OPT2 subsystem was most recently initiated, OPT2 subsystem processing takes precedence over OPT1 subsystem processing on image A.

Example

Figure 2 on page 20 shows examples that illustrate multiple instances of Advanced Allocation Management.

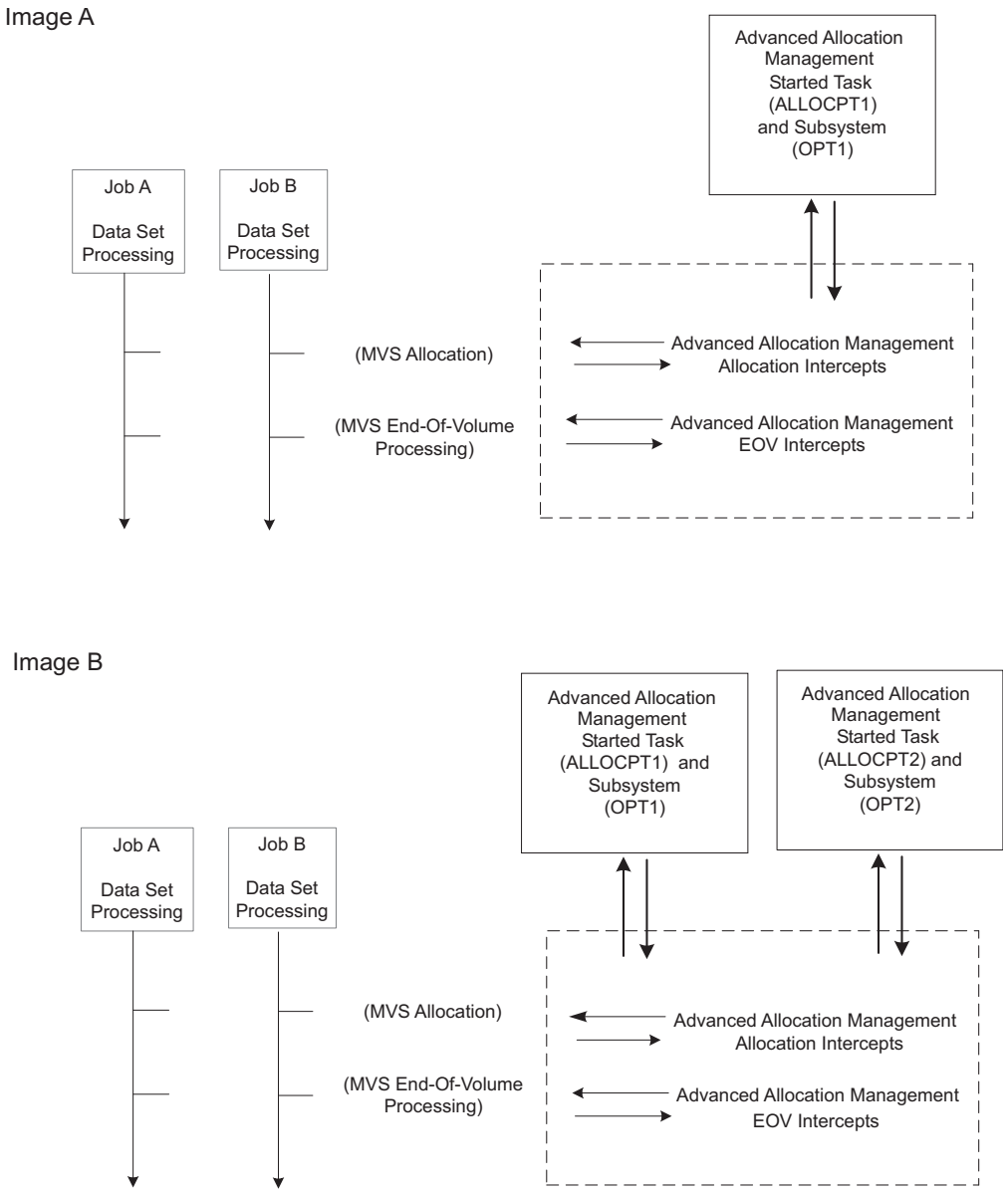


Figure 2. Advanced Allocation Management (multiple instances)

Note:

- The Advanced Allocation Management subsystem OPT1 that runs on image A is a different subsystem than the Advanced Allocation Management subsystem OPT1 that runs on image B.
- Each Advanced Allocation Management subsystem that runs on image B has a unique subsystem identifier and started task name.

An Advanced Allocation Management control data set can exist for each Advanced Allocation Management subsystem, or one control data set can be shared among subsystems.

Separate control data sets for multiple instances

If your installation requires users to have separate authorizations for each Advanced Allocation Management subsystem, use a separate control data set for

each instance. For example, you might want some users to use a specific Advanced Allocation Management subsystem, but limit their access to other Advanced Allocation Management subsystems.

Figure 3 shows an example where two separate control data sets exist for two separate Advanced Allocation Management subsystems (one control data set per subsystem).

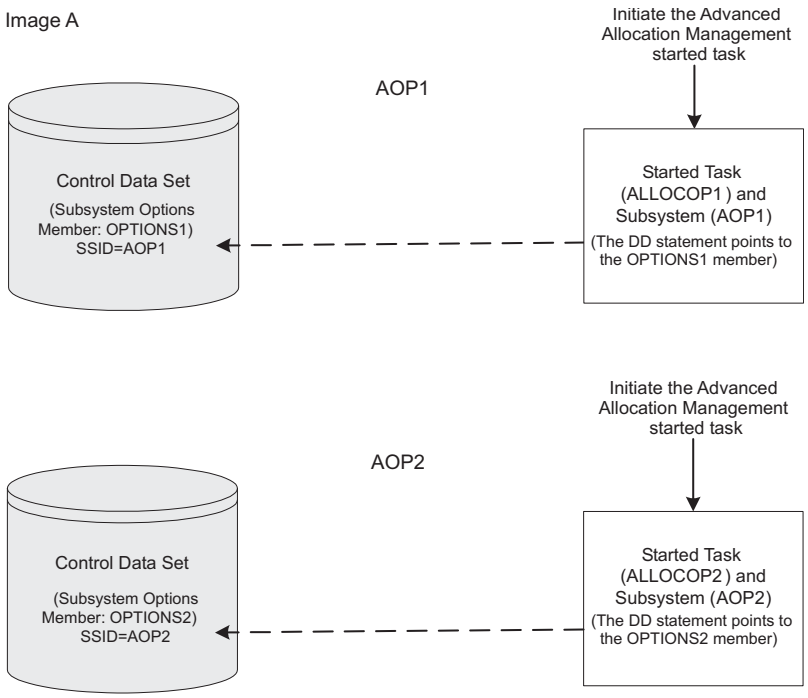


Figure 3. Multiple instances of Advanced Allocation Management (separate control data sets)

In Figure 3, each control data set is associated with a specific Advanced Allocation Management subsystem (AOP1 or AOP2), and each control data set contains a subsystem options member, a rule definitions member, and a volume group definitions member for the specific subsystem.

One shared control data set for multiple instances

For installations that do not require users to have separate authorization for individual Advanced Allocation Management subsystems, you can use a shared control data set for multiple instances. For example, you might want all users to be able to work with all Advanced Allocation Management subsystems in your environment.

Figure 4 on page 22 shows an example where a single control data sets exists for two separate Advanced Allocation Management subsystems.

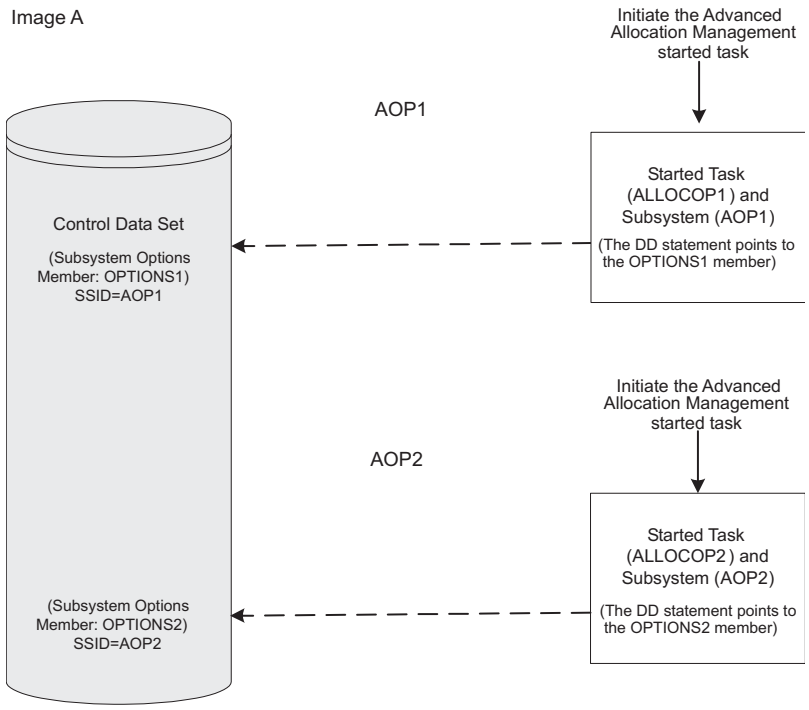


Figure 4. Multiple instances of Advanced Allocation Management (shared control data set)

In Figure 4, one control data set is associated with the AOP1 and AOP2 Advanced Allocation Management subsystems. The control data set contains a subsystem options member for each Advanced Allocation Management subsystem. The OPTIONS1 member specifies the subsystem options for subsystem AOP1. The OPTIONS2 member specifies the subsystem options for subsystem AOP2. You can define rule definitions and volume group definitions members for each subsystem, or the subsystems can share them.

Sharing rule definitions between instances

In most situations, sharing rule definitions between instances of Advanced Allocation Management that run on separate systems and that use different load libraries is not problematic. Every effort is made to ensure that RULEDEF syntax is upwardly compatible but downward compatibility is not guaranteed. If you plan to share rule definitions between instances of Advanced Allocation Management that run on separate systems and that use different load libraries, and you want to check for unidentified keywords, complete the following steps for each load library each version and level of Advanced Allocation Management with which you plan to use the rule definitions.

Complete the following steps:

1. Modify the GLOCLIST JCL by replacing `&HILEVEL` on the first statement in the CLIST.
2. Run Advanced Allocation Management. When you run Advanced Allocation Management, it fills in the subsequent references to the &HILEVEL variable automatically.
3. Run the syntax-check as described in "Syntax-checking control data set members" on page 277.

If the Advanced Allocation Management started task encounters an unidentified keyword, the rule definition activation fails and the message: `GLO1176E RULEDEFS MEMBER membername ACTIVATION FAILED - SEE JESYSMSG FOR DETAILS` is generated. In this situation, review the Advanced Allocation Management started task JESYSMSG.

If you see the message: `GLO4030E INVALID KEYWORD-keyword`, remove the invalid keyword from the rule definitions and activate the rule definitions again. For more information, see "Activating rule definitions" on page 46.

Guidelines for implementing Advanced Allocation Management

The following guidelines help facilitate the implementation of Advanced Allocation Management.

Refer to the following guidelines when you implement Advanced Allocation Management in your environment:

- If this is a new installation of Advanced Allocation Management or if Advanced Allocation Management is replacing a similar product, install and run Advanced Allocation Management in a test environment before you implement it in your production environment. Then you can ensure that the product is correctly configured before you use it with your current production data.
- After you install Advanced Allocation Management in a test environment, run a series of initial tests, followed by more extensive testing as you gain experience with the product. After successful testing in a test environment, gradually implement the product in your production environment.

  For example, create a series of test jobs that you can use to become familiar with Advanced Allocation Management features and functions. Use Advanced Allocation Management selection criteria to specify that only these test jobs are to be included in Advanced Allocation Management processing.
- Start with specific applications that have job names that begin with the same characters. For example, you might have a series of jobs that have job names that begin with the characters ABC. To specify that only these jobs are to be included in Advanced Allocation Management processing, use the selection criteria keyword JOBNAME with the INCLUDE keyword.
- Specify a select set of data set name high-level qualifiers and then gradually add selection criteria, as needed. For example, you might have a group of data sets that have the high-level qualifiers of SS22 and SS23. To include only these data sets in Advanced Allocation Management processing, use the selection criteria keyword DSNAME with the INCLUDE keyword. Add selection criteria as you become familiar with Advanced Allocation Management.
- When you add rule definitions, add the RULEDEF_TEST keyword and set it to ENABLE to verify that the selection criteria matches the jobs, data sets, and so on, that you intend. After you successfully test the rule definitions, set RULEDEF_TEST to DISABLE, or remove the keyword and allow Advanced Allocation Management to perform processing.

Chapter 3. Migrating to Advanced Allocation Management

The migration process consists of reviewing the migration requirements, installing Advanced Allocation Management in a test environment, converting product definitions, and migrating Advanced Allocation Management to the production environment.

When you migrate to Advanced Allocation Management, complete these steps:

1. Review the migration requirements. For more information, see "Migration requirements"
2. Install Advanced Allocation Management in a test environment. For more information, see "Installing Advanced Allocation Management in a test environment."
3. Convert product definitions and test. For more information, see "Converting product definitions and testing" on page 26.
4. Migrate Advanced Allocation Management to your production environment. For more information, see "Migrating to production" on page 26.

Migration requirements

During the migration process it is not necessary to uninstall similar products.

However, to avoid potential conflicts and help ensure trouble-free operation, keep in mind the following points:

- Do not run Advanced Allocation Management concurrently with a similar product. Advanced Allocation Management cannot run concurrently with other products that perform similar functionality.
- If a similar product was running on the system, perform an IPL before you start Advanced Allocation Management (and vice versa).
- If Advanced Allocation Management is replacing a similar product, be sure to perform an IPL before you start Advanced Allocation Management.

Installing Advanced Allocation Management in a test environment

Fully testing Advanced Allocation Management in a test environment before you install it in your production environment helps ensure that the migration works as expected and that the product is configured according to your needs without impacting your current production data.

1. Review all installation and migration documentation, including the prerequisites.
2. Collect existing product data from the production environment.
3. Determine the test system on which to install Advanced Allocation Management.

If Advanced Allocation Management is replacing a similar product, perform the migration process in a test environment before you perform the migration process in your production environment. After you successfully perform the migration process in the test environment, you can begin the process of migrating from your previous product to Advanced Allocation Management in your production environment.

Complete these steps to install Advanced Allocation Management in a test environment:

1. Install Advanced Allocation Management on the test system using the supplied sample rule definitions (RULEDEFS), volume group definitions (VGRPDEFs), and user-defined variable (VARDEFS) members that are generated during the installation process.
2. Install the required and recommended Advanced Allocation Management maintenance on the test system.
3. Start Advanced Allocation Management. For more information, see "Starting the started task" on page 33.
4. Run the installation verification program (IVP) job that is supplied in the sample library data set. For more information, see "Verifying the installation" on page 34.

Converting product definitions and testing

Complete these steps to convert the product definitions.

1. If, after you run the IVP job, you want to run additional test jobs, create a copy of the RULEDEFS, VGRPDEFS, and VARDEFS members and modify them to include the test jobs for Advanced Allocation Management processing.

   When you modify the RULEDEFS and VGRPDEFS members to specify your test jobs, add the RULEDEF_TEST keyword and set it to ENABLE. Specifying `RULEDEF_TEST=ENABLE` enables rule definition testing to verify that the selection criteria that you specify match the jobs, data sets, and so on, that you intend. For more information, see "Test a rule definition (RULEDEF_TEST)" on page 92.
2. After you modify the rule or volume group definitions, you must activate the definitions so that the settings take effect. For more information, see "Activating rule definitions" on page 46 and "Activating volume group definitions" on page 267.
3. Run the test jobs to confirm that Advanced Allocation Management is working as you intended.

   During testing, modify the primary and secondary space values in your test application JCL so that out-of-space conditions occur.
4. After you successfully test your rule definitions, set RULEDEF_TEST to DISABLE, or remove it and allow Advanced Allocation Management to perform processing.
5. Manually convert your existing product processing definitions to Advanced Allocation Management rule and volume group definitions.
6. In the test environment, run Advanced Allocation Management with the converted production control members and copies of the production jobs.

Migrating to production

After testing is complete and all issues are resolved, run Advanced Allocation Management in place of your existing product on the production system.

Note: An IPL is required if the existing product was previously running on the z/OS image after the last IPL.

Chapter 4. Configuring Advanced Allocation Management

To install Advanced Allocation Management, follow the installation instructions that are provided in the *Program Directory for Advanced Allocation Management for z/OS*.

After you install Advanced Allocation Management, perform the following steps to configure Advanced Allocation Management.

Configuration summary

The following table summarizes the steps that you must complete to successfully configure Advanced Allocation Management for your environment.

Note: The product prefix for Advanced Allocation Management is GLO. This prefix identifies the product in product messages, data set members, and so on.

Table 1. Summary of Advanced Allocation Management configuration steps

Step	Description	Instructions
1	Authorize the required load libraries.	"APF-authorizing the load libraries"
2	Configure the started task JCL (GLOJSTC).	"Configuring the started task JCL" on page 28
3	Allocate the Advanced Allocation Management control data set (GLOJCNTL).	"Allocating the control data set" on page 29
4	Specify the Advanced Allocation Management subsystem options (OPTIONS).	"Specifying subsystem options" on page 29
5	Secure the user interface update functions, the control data set, and internal work data sets.	"Securing Advanced Allocation Management" on page 31
6	Configure the CLISTS (GLO, GLOCLIST).	"Configuring the CLISTs and starting the user interface" on page 33
7	Initiate the Advanced Allocation Management started task.	"Starting the started task" on page 33
8	Verify the installation (GLOJIVP1).	"Verifying the installation" on page 34

Prerequisite tasks

Before you configure Advanced Allocation Management, perform the following steps.

1. Copy your SMP/E target libraries to your runtime libraries.
2. Configure the runtime libraries.

APF-authorizing the load libraries

The load library that contains `HILEVEL.SGLOLOAD` must be APF authorized.

To ensure that the Advanced Allocation Management (HILEVEL.SGLOLOAD) library is APF-authorized, include it in your system APF-authorized list. For information about APF authorization, see the *MVS Programming: Authorized Assembler Services Guide*.

Configuring the started task JCL

The GLOJSTC member of the Advanced Allocation Management sample library (SGLOSAMP) contains the started task JCL.

Note: To help ensure that Advanced Allocation Management is available to handle abends, configure the started task to start automatically after an IPL of z/OS. Add the appropriate command to the COMMND*xx* member in SYS1.PARMLIB.

To configure the started task JCL, complete these steps:

1. Copy the started task JCL to your system PROCLIB.

   Tip: Rename the started task member so that it is easily identifiable with Advanced Allocation Management. For example: ALLOCOPT.

2. Verify that the JCL contains the statement `//stcname PROC OPTSMBR=OPTIONS` where *stcname* is the name of the started task and OPTSMBR references the member name OPTIONS.

   Note: OPTIONS is the default member name of the subsystem options member of the Advanced Allocation Management control data set that is allocated in "Allocating the control data set" on page 29.

3. To configure the started task JCL that you copied to your system PROCLIB replace GLO.V3R3 with the high-level qualifier of the installed Advanced Allocation Management load libraries. Then verify the following items:
   - Ensure that a DD statement with a DDNAME of OPTIONS is in the started task. The DD statement points to the subsystem options member of the Advanced Allocation Management control data set (OPTIONS), which contains the global settings for the product.
   - Ensure that a DD statement with a DDNAME of CONTROL is present in the started task. For example:

     ```
     //CONTROL DD DSN=GLO.V3R3.CONTROL,DISP=SHR
     ```

     The DD statement points to the control data set that contains the rule definitions (RULEDEFS), volume group definitions (VGRPDEFS), and user-defined variables (VARDEFS).
   - Ensure that the started task has the authority to read the Advanced Allocation Management control data set and load library.
4. To use the NOTCAT2 functions of UNCATALOG, SCRATCH, and RENAME, use your security authorization product to authorize the started task to allow the uncataloging, deleting, and renaming of data sets.
5. Add the started task JCL to z/OS for started task initiation.

Note: Do not initiate the started task until you finish configuring Advanced Allocation Management.

Allocating the control data set

The JCL to create the control data set is in the GLOJCNTL member of the SGLOSAMP library.

To allocate the control data set and generate the sample PDS members, complete the following steps:

1. To configure the GLOJCNTL member, replace `GLO.V3R3` with the high-level qualifier of the installed Advanced Allocation Management load libraries.

   Note: The high-level qualifier that you specify must match the high-level qualifier that you specify in the started task JCL.
2. Submit the JCL to create the control data set. The JCL creates the control data set and populates the data set with the following sample members: subsystem options (OPTIONS), rule definition (RULEDEFS), volume group definition (VGRPDEFS), and user-defined variables (VARDEFS).

Specifying subsystem options

The subsystem options are in the OPTIONS member of the control data set that the GLOJCNTL member JCL generates.

Note:

- Subsystem options settings take effect when the started task that is associated with the specific subsystem options member is started.
- You cannot specify the subsystem options member name as parameter on a command.

To specify subsystem options, modify the following settings.

DSN_PATTERN_MATCHING
Specify data set name pattern-matching for the selection criteria DSNAME, DSNAME_DATA, DSNAME_INDEX, and LOGSTREAMID. Specify one of the following options:

BASIC
(Default.) Specify default data set name pattern-matching.

- One percent sign matches one (required) character.
- One asterisk within a qualifier that contains other characters and wildcard characters, matches 0 - 8 characters within the qualifier.

  If you specify a single asterisk to denote a qualifier, for example .*. or .*, it matches on one or more qualifiers. If no such qualifier exists, the match fails because BASIC pattern matching searches for a string and is not aware of individual qualifiers.

Important: For existing users who installed Advanced Allocation Management, changing from the default of BASIC requires that you carefully review the data set name patterns that are specified for DSNAME, DSNAME_DATA, DSNAME_INDEX, and DSNAME_LOGSTREAMID selection criteria to ensure that your current rule definitions continue to work as expected.

EXTENDED1

Specify extended data set name pattern matching.

- A single percent sign matches on a single (required) character.
- A single asterisk, *within a qualifier that contains other characters*, matches on 0 - 8 characters. A qualifier that consists of a single asterisk represents one (required) qualifier.
- Double asterisks match zero or more qualifiers. A trailing qualifier of double asterisks (.**) matches zero or more qualifiers.

EXTENDED2
The EXTENDED2 option works in the same way as the EXTENDED1 option. In addition, a trailing qualifier of a single asterisk (.*) matches zero or more qualifiers.

In all cases, specifying double asterisks within a qualifier, such as ABC.D**EF, is not permitted.

For more information, see "Data set name filtering" on page 258.

INITIAL_RULEDEF
Specify the name of the rule definitions member to use upon startup. For example: RULESABC, RULESET2, and so on. The default is RULEDEFS.

INITIAL_VGRPDEF
Specify the name of the volume group definitions member to use upon startup. For example: VGRPFST, VGRPD002. The default is VGRPDEFS.

S1_INTERCEPT
Specify whether to enable or disable the initialization of the S1 (STOW) intercept at product startup. Specify one of the following options:

ENABLE
(Default.) Enable the initialization the S1 (STOW) intercept at product startup. To use the PDS_DIR_INC function, you must set S1_INTERCEPT to ENABLE.

DISABLE
Disable the initialization of the S1 (STOW) intercept at product startup. Specify DISABLE only when conflicts occur with other software that intercepts STOW processing.

Note: If you modify the S1_INTERCEPT setting in the OPTIONS member, you must IPL the system and restart Advanced Allocation Management so that the new setting takes effect.

SECURITY_CLASS
The default is FACILITY.

SMF_RECORD_ID
Specify the SMF record type to use for Advanced Allocation Management.

Notes:

1. If you do not specify the SMF record type, Advanced Allocation Management will not write SMF records.
2. If you are already recording SMF data for a product and want to record SMF data for Advanced Allocation Management, you must select a new, previously unused record type.

For information about how Advanced Allocation Management uses the SMF record type that you specify, see Chapter 10, "Event recording and reporting," on page 287.

SSID You must specify a unique 4 character alphanumeric subsystem ID for each specific instance of Advanced Allocation Management. For example, specify AOP1, AOP2, and so on. The default subsystem ID is AOPT.

VARDEFS_MEMBER
Specify the name of the variables definitions member to use upon startup. The default variables definitions member name is VARDEFS.

Note: To later change the variables definitions member, use the following command:

```
F glostc,SETVARDEFS membername
```

Where *membername* is the name of the variables definitions member to use.

Z1_INTERCEPT
Specify whether to enable or disable initialization of the Z1 (step termination) intercept at product startup. Specify one of the following options:

ENABLE
Initialize the Z1 intercept at product startup. Specify ENABLE only if you want to use the DATASET_INIT function.

DISABLE
(Default.) Do not initialize the Z1 intercept at product startup.

Note: If you modify the Z1_INTERCEPT setting within the OPTIONS member, you must IPL the system and restart Advanced Allocation Management so that the setting takes effect.

Securing Advanced Allocation Management

Complete the following steps to secure Advanced Allocation Management.

1. Because the control data set contains options that define Advanced Allocation Management processing, secure the control data set as you would any other system data set.
2. Secure the user interface update functions. For more information, see "Securing the user interface update functions."
3. Define security profiles to allow the creation and deletion of the internal data sets. For more information, see "Defining security profiles" on page 33.

Securing the user interface update functions

By default, the Tivoli Advanced Allocation Management Subsystem Settings panel provides users with the ability to view and update selected subsystem options. Advanced Allocation Management uses RACF® and RACF profiles and access authorities to determine whether a user can perform update functions.

Note: While this discussion refers specifically to RACF, it also applies to other SAF-based security products that you can use to secure the user interface update functions.

Advanced Allocation Management uses the following RACF profiles to determine whether a specific user can perform update functions:

- GLO.ADMIN.*ssid* (Where *ssid* is a 4 character Advanced Allocation Management subsystem, for example: AOP1)
- GLO.ADMIN.* (Where * indicates all subsystems)

To perform update functions, a user must have a record in one of the GLO.* profiles that are within the FACILITY class and the user's access authority must be UPDATE. If the access authority is not UPDATE, the user can only view information on the Tivoli Advanced Allocation Management Subsystem Settings panel.

Note: The access authority for a specific subsystem profile (GLO.ADMIN.*ssid*) takes precedence over the generic "all subsystems" profile (GLO.ADMIN.*). For example, if a user has UPDATE authority for all subsystems, and has only view privileges for a specific subsystem profile, the user can only view subsystem settings for that particular subsystem.

The profiles and access authorities are summarized in Table 2.

Table 2. RACF profiles and access authorities used to secure update functions

Profile	Access authority	Functions
GLO.ADMIN.*ssid*	UPDATE	The user can perform update functions for the subsystem that is specified by *ssid*
GLO.ADMIN.*	UPDATE	The user can perform update functions for all subsystems
GLO.ADMIN.*ssid*	(not UPDATE)	The user can perform only view functions for the subsystem that is specified by *ssid*.
GLO.ADMIN.*	(not UPDATE)	Perform only view functions for all subsystems.

Note: The profiles that are described in Table 2 must exist in the RACF FACILITY class. If the FACILITY class does not exist, the update functions are not secure.

If you use RACF to secure the update functions, the following actions occur when a user selects an update option from the Tivoli Advanced Allocation Management Subsystem Settings panel:

1. Advanced Allocation Management instructs RACF to verify whether the resource (the update options on the Tivoli Advanced Allocation Management Subsystem Settings panel) is protected.
2. If the resource is protected, RACF uses the resource name to search for the GLO.ADMIN.*ssid* profile. If RACF does not find the profile, it searches for generic profiles in the following order: GLO.ADMIN.*, GLO.*, and finally, **.
3. When RACF finds a profile, it determines whether the user is on the access list of the profile:
   - If the user is on the access list of the profile, RACF compares the access intent of the user to the access authority. If the user has access authority, the user is allowed to perform the update option.
   - If the user is not on the access list of the profile, RACF compares the access intent of the user to the universal access (UACC) of the profile, and the user is permitted (or not permitted) to perform the update option.

To secure the Tivoli Advanced Allocation Management Subsystem Settings panel update functions, complete these steps:

1. Use RACF to create the GLO.ADMIN.*ssid* and GLO.ADMIN.* profiles.
2. Add user records to the appropriate profiles, and give each user the appropriate access authority. For information about RACF profiles and access authorities, see *RACF General Information*.

Defining security profiles

Some Advanced Allocation Management internal work data sets are created using a high-level qualifier that consists of the value of the TSO PROFILE PREFIX. If the TSO PROFILE is set to NOPREFIX, the high-level qualifier is set to the TSO USERID.

If the TSO PROFILE is set to NOPREFIX, complete the following step:

Define the security profiles to allow the creation and deletion of these internal data sets

Configuring the CLISTs and starting the user interface

Before you start the Advanced Allocation Management user interface, you must configure two CLISTs (GLO and GLOCLIST) in the SGLOSAMP sample library.

Using the statements within the GLO member, you can start the Advanced Allocation Management user interface CLIST (GLOCLIST) as a separate application, therefore enabling you to continue your ISPF session should you encounter a problem with this instance of Advanced Allocation Management. The GLOCLIST starts the Advanced Allocation Management user interface.

To configure the CLISTS and start the Advanced Allocation Management user interface, complete these steps:

1. Copy the GLO member to your system CLIST library.
2. To configure the GLO CLIST, specify the name of the data set where the GLOCLIST member was copied in the PROC 0 CLISTLIB () statement. For example, use the following statement to specify the name of the data set to which you copied the GLOCLIST member:

   ```
   PROC 0 CLISTLIB(ABCD.USER.CLIST)
   ```
3. To configure the GLOCLIST, specify the high-level qualifier between the parentheses. Use the following statement to specify the high-level qualifier:

   PROC 0 HILEVEL(*HILEVEL*)

   Where *HILEVEL* is the high-level qualifier. For example: PROC 0 HLQ(GLO.V3R3)
4. To start the user interface and access the Tivoli Advanced Allocation Management Primary Option Menu, type the TSO command: TSO GLO

Starting the started task

Issue commands at the console to start and stop the started task.

- To start the started task, issue the following command from the operator console: **START** *stcname* where *stcname* is the name of the started task.
- To stop the started task, issue the following command from the operator console: **STOP** *stcname*, where *stcname* is the name of the started task.

Verifying the installation

Use the JCL in the GLOJIVP1 member of the SGLOSAMP Advanced Allocation Management sample library to verify the installation.

The JCL in the GLOJIVP1 member is used with the sample rule definitions (RULEDEFS), volume group definitions (VGRPDEFS), and user-defined variable (VARDEFS) members to verify the basic operation of the Advanced Allocation Management functions.

Run the JCL in the GLOJIVP1 member on the z/OS image where you want to verify the installation of Advanced Allocation Management.

- Any SMS data class to which an IVP test data set is assigned should have the SMS ADD'L VOLUME AMOUNT attribute option set to (P) Primary.
- Modify the GLOSVARS VARDEFS member VALUE= keyword to specify the name of the data set that you modified in the VARSUB step of GLOJIVP1.

To verify the Advanced Allocation Management installation, complete the following steps:

1. Modify the JCL in the GLOJIVP1 member and change all occurrences of the high-level qualifier `?GLOHLQ?` to `GLO.V3R3` or to the high-level qualifier of the installed Advanced Allocation Management libraries.
2. Modify the VGRPDEFS member and add volume serial numbers for the initial test volume group.

   Important: Do not modify the RULEDEFS member. The RULEDEFS member is predefined to work with the installation verification JCL in member GLOJIVP1.
3. If you have not already done so, start the Advanced Allocation Management started task.
4. Submit the modified JCL in the GLOJIVP1 member.
5. For each job step, compare your results with the expected results described in Table 3.

Table 3. GLOJIVP1 summary and expected results

Step	Description	Expected results
DELETE1-DELETE9, DELETEA-DELETED	The DELETE steps deletes any previously used test data sets that might alter your results.	Return code 0.
ALLOC1	The ALLOC1 step allocates the test data set.	Return code 0.

AVS	The AVS step tests the AVS function for non-VSAM data sets.
	Note: The data set that is created in this step must be a non SMS-managed data set.
		Return code 0 and the following messages:
		`GLO2148I AVS HAS SUCCESSFULLY SELECTED ALL VOLUMES`
		`GLO2156I VOLSER(S): vvvvvv`
		`GLO2123I GLOJIVP1,AVS,,SYSUT2,dsname`
		Where *dsname* is the name of the data set, and *vvvvvv* is the chosen volser.

Table 3. GLOJIVP1 summary and expected results (continued)

Step	Description	Expected results
DIRINC	The DIRINC step tests the PDS_DIR_INC function by dynamically adding a second directory block to a PDS that has a full directory.	Return code 0 and the following messages: `GLO2150I INCREASING PDS DIRECTORY SIZE` `GLO2123I GLOJIVP1,DIRINC,,DD21,`*`dsname`* Where *dsname* is the name of the data set.
DSINIT	The DSINIT step tests the DATASET_INIT function.	Return code 0 and the following messages: `GLO2080I INITIALIZING DATA SET` `GLO2123I GLOJIVP1,DSINIT,,DD1,`*`dsname`* Where *dsname* is the name of the data set. **Note:** These results occur only if Advanced Allocation Management was initialized with `Z1_INTERCEPT=ENABLE` specified in the OPTIONS member. Otherwise, the step ends with RC=0; the data set is not initialized; and the messages are not issued.
NCATCRE	The NCATCRE step creates the test data set for NOTCAT2=SCRATCH in preparation for the next test.	Return code 0.
NCATSCR	The NCATSCR step uses a test data set to test the NOTCAT2 function. This step tests the NOTCAT2=SCRATCH function. After the test finishes, the data set is deleted.	Return code 0 and the following messages: `GLO2084I DATA SET ON VOLSER `*`vvvvvv`*` SCRATCHED AT ALLOCATION TIME` `GLO2123I GLOJIVP1,NCATSCR,,SYSUT2,`*`dsname`* Where *dsname* is the name of the data set, and *vvvvvv* is the volser of the volume on which the old data set was cataloged.
NOTCAT2	The NOTCAT2 step tests the NOTCAT2 function. **Note:** • You must manually delete the SYSUT4 and SYSUT5 data sets that are created as a result of this step because successful verification of the NOTCAT2 step results in the termination of the job. • The data sets you specify for this step must not be SMS-managed data sets.	Return code 0 and the following messages: `GLO2051I JOB SET TO FAIL DUE TO NOT CATLGD 2 ON VOLSER `*`vvvvvv`* `GLO2123I GLOJIVP1,NOTCAT2,,SYSUT5,`*`dsname`* Where *dsname* is the name of the data set, and *vvvvvv* is the volser of the volume on which the NOT CATLGD 2 error occurred.
PRIMRED	The PRIMRED step tests the PRIM_REDUCE function by reducing a high primary allocation quantity for the test data set until the allocation is successful.	Return code 0 and the following messages: `GLO2118I REDUCING PRIMARY ALLOCATION TO `*`nnnnn`*` TRACKS` `GLO2123I GLOJIVP1,PRIMRED,,SYSUT2,`*`dsname`* Where *nnnnn* is the new allocation quantity, and *dsname* is the name of the data set. **Note:** Multiple sets of the messages might occur.

Table 3. GLOJIVP1 summary and expected results (continued)

Step	Description	Expected results
SECALLOC	The SECALLOC step tests the SEC_ALLOC function by adding a secondary allocation amount when the test data set expands to its second extent.	Return code 0 and the following messages: `GLO2100I SETTING SECONDARY ALLOCATION TO 1 TRKS` `GLO2123I GLOJIVP1,SECALLOC,,SYSUT2,`*`dsname`* Where *dsname* is the name of the data set.
SECBEST	The SECBEST step tests the SEC_BEST function by reducing the secondary allocation quantity for the test data set to a value that fits on the volume.	Return code 0 and the following messages: `GLO2103I EXTENT-MATCHING SECONDARY ALLOCATION TO `*`nnnnn`*` TRKS` `GLO2123I GLOJIVP1,SECBEST,,SYSUT2,`*`dsname`* Where *nnnnn* is the new allocation quantity that reflects the largest available extent, and *dsname* is the name of the data set.

SECINC	The SECINC step tests the SEC_INC function by starting at the third extent of the test data set and increasing the secondary allocation amount by 100% of the previous amount.	Return code 0 and the following messages:
		`GLO2106I INCREASING SECONDARY ALLOCATION TO 2 TRKS`
		`GLO2123I GLOJIVP1,SECINC,,SYSUT2,`*`dsname`*
		`GLO2106I INCREASING SECONDARY ALLOCATION TO 4 TRKS`
		`GLO2123I GLOJIVP1,SECINC,,SYSUT2,`*`dsname`*
		Where *dsname* is the name of the data set.

SECRED	The SECRED step tests the SEC_REDUCE function by reducing the secondary allocation amount for the test data set to the largest available extent, after a secondary allocation failure occurs.	Return code 0 and the following messages: `GLO2109I REDUCING SECONDARY ALLOCATION TO `*`nnnnn`*` TRKS` `GLO2123I GLOJIVP1,SECINC,,SYSUT3,`*`dsname`* Where *nnnnn* is the new allocation quantity that reflects the largest available extent, and *dsname* is the name of the data set.
SECREL	The SECREL step tests the SEC_RELEASE function by allocating several tracks for the test data set. Only one track is used. (A partial release truncates the unused tracks and leaves only one track for the primary allocation.)	Return code 0 and the following messages: `GLO2125I SPACE RELEASE RECOVERED `*`nnnnnnnnn`*` TRACKS` `GLO2123I GLOJIVP1,SECREL,,SYSUT2,`*`dsname`* Where *nnnnnnnnn* is the number of tracks that are released, and *dsname* is the name of the data set.
SETCLASS	The SETCLASS step tests the SET_DATACLAS function. The data set is assigned an incorrect data class in the JCL, and then the SET_DATACLAS function removes the incorrect data class.	Return code 0 and the following messages: `GLO2194I SET PROCESSING COMPLETE; DATACLAS=(NULL) (OLD VALUE=`*`ddddddddd`*`)` `GLO2123I GLOJIVP1,,,SYSUT9,`*`dsname`* Where *dsname* is the name of the data set, and *ddddddddd* is the name of the data class.

Table 3. GLOJIVP1 summary and expected results (continued)

Step	Description	Expected results
SETUNITNT	The SETUNITNT step tests the SET_UNITNAME function. The data set is assigned an incorrect unit name in the JCL. Then the SET_UNIT_NAME function changes the name to SYSALLDA.	Return code 0 and the following messages: `GLO2194I SET PROCESSING COMPLETE; UNIT_NAME=SYSALLDA (OLD VALUE=ABCDEFGH)` `GLO2123I GLOJIVP1,SETUNITN,,DD1,`*`dsname`* Where *dsname* is the name of the data set.
VARSUB	The VARSUB step tests the variable substitution selection substitution.	Return code 0 and the following messages: `GLO2188I MATCHING INCLUDE RULEDEF LINE NUMBER =` *`nnnnn - ssys`* `GLO2123I GLOJIVP1,VARSUB,,SYSUT1,`*`ddddd.dddddd.dddddd - ssys`* Where *nnnnn* is the matching RULEDEF line number; *ssys* is the Advanced Allocation Management subsystem identifier; and *ddddd.dddddd.dddddd* is the modified data set name that is specified by &GLOVAR1 in the GLOSVARS variable definition control member.

VAVS	The VAVS step tests the AVS function for VSAM data sets.
	Note: The data set that is created in this step must be a non SMS-managed data set.
		Return code 0 and the following messages:
		
		`GLO3148I AVS HAS SUCCESSFULLY SELECTED ALL VOLUMES-DATA`
		
		`GLO3156I VOLSER(S):` *`vvvvvv`*
		
		`GLO3123I GLOJIVP1,VAVS,,`*`dsname`*
		
		`GLO3148I AVS HAS SUCCESSFULLY SELECTED ALL VOLUMES-INDEX`
		
		`GLO3156I VOLSER(S):` *`vvvvvv`*
		
		`GLO3123I GLOJIVP1,VAVS,,`*`dsname`*
		
		Where *dsname* is the name of the data set, and *vvvvvv* is the chosen volser.

VOLADD	The VOLADD step tests the VOL_ADD function by testing the need to extend to an additional volume.	Return code 0 and the following messages: `GLO2128I DATA SET SUCCESSFULLY EXTENDED TO VOLUME` *`vvvvvv`* `GLO2123I GLOJIVP1,VOLADD,,SYSUT4,`*`dsname`* Where *dsname* is the name of the data set, and *vvvvvv* is the volser of the volume to which the data set was extended.
VPRIMRED	The VPRIMRED step tests the PRIM_REDUCE function by defining a VSAM data set with a high primary allocation quantity.	Return code 0 and the following messages: `GLO3118I REDUCING PRIMARY ALLOCATION TO` *`nnnnn`* `CYLINDERS` `GLO3123I GLOJIVP1,VPRIMRED,,`*`dsname`* Where *nnnnn* is the new allocation quantity, and *dsname* is the name of the data set. **Note:** Multiple sets of the first two messages might occur.

Table 3. GLOJIVP1 summary and expected results (continued)

Step	Description	Expected results
VSECALC	The VSECALC step tests the SEC_ALLOC function by defining a VSAM data set with a zero secondary allocation quantity and then copying data into it.	Return code 0 and the following messages: `GLO3100I SETTING SECONDARY ALLOCATION TO 1 CYLS` `GLO3123I GLOJIVP1,VSECALC,,SYS0002,`*`dsname`* Where *dsname* is the name of the data set.
VSECBEST	The VSECBEST step tests the SEC_BEST function by defining a VSAM data set with a high secondary allocation quantity and then copying data into it.	Return code 0 and the following messages: `GLO3103I EXTENT-MATCHING DATA SECONDARY ALLOCATION TO `*`nnnnn`*` CYLS` `GLO3123I GLOJIVP1,VSECBST,,SYS0004,`*`dsname`* Where *nnnnn* is the new allocation quantity that reflects the largest available extent, and *dsname* is the name of the data set.
VSECRED	The VSECRED step tests the SEC_REDUCE function by defining a VSAM data set with a high secondary allocation quantity and then copying data into it.	Return code 0 and the following messages: `GLO3109I REDUCING DATA SECONDARY ALLOCATION TO `*`nnnnn`*` CYLS` `GLO3123I GLOJIVP1,VSECRED,,SYS0006,`*`dsname`* Where *nnnnn* is the new allocation quantity that reflects the largest available extent, and *dsname* is the name of the data set.
VVOLADD	The VVOLADD step tests the VOL_ADD function by defining a VSAM data set with a zero secondary allocation quantity and then copying data into it.	Return code 0 and the following messages: `GLO3128I DATA SUCCESSFULLY EXTENDED TO VOLUME `*`vvvvvv - ssss`* `GLO3123I GLOJIVP1,VVOLADD,,SYS0008,`*`dsname`* Where *dsname* is the name of the data set; *vvvvvv* is the volser of the volume to which the data set was extended; and *ssss* is the subsystem.

Successful GLOJIVP1 results

Figure 5 on page 39 shows an example of successful GLOJIVP1 JCL results.

```
IRR010I  USERID PDABC1   IS ASSIGNED TO THIS JOB.
GLO2194I SET PROCESSING COMPLETE; DATACLAS=(NULL) (OLD VALUE=TEST)
GLO2123I GLOJIVP1,,,SYSUT9,NSMS.GLO.TEST.DATA
ICH70001I PDABC1   LAST ACCESS AT 11:54:14 ON TUESDAY, JUNE 30, 2009
$HASP373 GLOJIVP1 STARTED - INIT 1    - CLASS A - SYS RL13
RKTSW01I                                      ------------ Timings (HH:MM:SS.hh) --------------
RKTSW01I   JOBNAME  STEPNAME PROCSTEP STEP    RC  CPU (Total)  CPU (TCB)    CPU (SRB)    CPU (Other)
RKTSW01I   GLOJIVP1 DELETE1             1     00  00:00:00.47  00:00:00.06  00:00:00.00  00:00:00.41
RKTSW01I            PRIMRED             2     00  00:00:00.10  00:00:00.08  00:00:00.00  00:00:00.02
RKTSW01I            DELETE2             3     00  00:00:00.08  00:00:00.06  00:00:00.00  00:00:00.02
GLO2100I SETTING SECONDARY ALLOCATION TO 1  TRKS -J310
GLO2123I GLOJIVP1,SECALLOC,,SYSUT2,NSMS.GLO.TEST.DATA                                      -J310
RKTSW01I            SECALLOC            4     00  00:00:00.10  00:00:00.08  00:00:00.00  00:00:00.02
RKTSW01I            DELETE3             5     00  00:00:00.08  00:00:00.06  00:00:00.00  00:00:00.02
GLO2103I EXTENT-MATCHING SECONDARY ALLOCATION TO 3930  TRKS -J310
GLO2123I GLOJIVP1,SECBEST,,SYSUT2,NSMS.GLO.TEST.DATA                                       -J310
RKTSW01I            SECBEST             6     00  00:00:00.10  00:00:00.08  00:00:00.00  00:00:00.02
RKTSW01I            DELETE4             7     00  00:00:00.08  00:00:00.06  00:00:00.00  00:00:00.02
GLO2106I INCREASING SECONDARY ALLOCATION TO 2  TRKS -J310
GLO2123I GLOJIVP1,SECINC,,SYSUT2,NSMS.GLO.TEST.DATA                                        -J310
GLO2106I INCREASING SECONDARY ALLOCATION TO 4  TRKS -J310
GLO2123I GLOJIVP1,SECINC,,SYSUT2,NSMS.GLO.TEST.DATA                                        -J310
RKTSW01I            SECINC              8     00  00:00:00.12  00:00:00.09  00:00:00.00  00:00:00.03
RKTSW01I            DELETE5             9     00  00:00:00.08  00:00:00.06  00:00:00.00  00:00:00.02
GLO2109I REDUCING SECONDARY ALLOCATION TO 65534  TRKS -J310
GLO2123I GLOJIVP1,SECRED,,SYSUT3,NSMS.GLO.TEST.DATA                                        -J310
RKTSW01I            SECRED             10     00  00:00:00.10  00:00:00.08  00:00:00.00  00:00:00.02
RKTSW01I            DELETE6            11     00  00:00:00.08  00:00:00.06  00:00:00.00  00:00:00.02
GLO2125I SPACE RELEASE RECOVERED 9 TRACKS -J310
GLO2123I GLOJIVP1,SECREL,,SYSUT2,NSMS.GLO.TEST.DATA -J310
RKTSW01I            SECREL             12     00  00:00:00.10  00:00:00.08  00:00:00.00  00:00:00.02
RKTSW01I            DELETE7            13     00  00:00:00.08  00:00:00.06  00:00:00.00  00:00:00.02
GLO2128I DATA SET SUCCESSFULLY EXTENDED TO VOLUME NSL204 -J310
GLO2123I GLOJIVP1,VOLADD,,SYSUT4,NSMS.GLO.TEST.DATA                                        -J310
RKTSW01I            VOLADD             14     00  00:00:00.21  00:00:00.16  00:00:00.02  00:00:00.03
RKTSW01I            DELETE8            15     00  00:00:00.08  00:00:00.06  00:00:00.00  00:00:00.02
RKTSW01I            ALLOC1             16     00  00:00:00.07  00:00:00.06  00:00:00.00  00:00:00.01
GLO2150I INCREASING PDS DIRECTORY SIZE -J310
GLO2123I GLOJIVP1,DIRINC,,DD21,NSMS.GLO.TEST.DATA                                          -J310
RKTSW01I            DIRINC             17     00  00:00:00.40  00:00:00.27  00:00:00.02  00:00:00.11
RKTSW01I            DELETE9            18     00  00:00:00.08  00:00:00.06  00:00:00.00  00:00:00.02
GLO2080I INITIALIZING DATA SET -J310
GLO2123I GLOJIVP1,DSINIT,,DD1,NSMS.GLO.TEST.DATA                                           -J310
RKTSW01I            DSINIT             19     00  00:00:00.09  00:00:00.07  00:00:00.00  00:00:00.02
RKTSW01I            DELETEA            20     00  00:00:00.08  00:00:00.06  00:00:00.00  00:00:00.02
RKTSW01I            SETCLASS           21     00  00:00:00.08  00:00:00.06  00:00:00.00  00:00:00.02
RKTSW01I            DELETEB            22     00  00:00:00.08  00:00:00.06  00:00:00.00  00:00:00.02
GLO2194I SET PROCESSING COMPLETE; UNIT_NAME=SYSALLDA (OLD VALUE=ABCDEFGH)                   -J310
GLO2123I GLOJIVP1,SETUNITN,,DD1,NSMS.GLO.TEST.DATA                                         -J310
RKTSW01I            SETUNITN           23     00  00:00:00.08  00:00:00.06  00:00:00.00  00:00:00.02
RKTSW01I            DELETEC            24     00  00:00:00.08  00:00:00.06  00:00:00.00  00:00:00.02
GLO2148I AVS HAS SUCCESSFULLY SELECTED ALL VOLUMES -J310
GLO2156I VOLSER(S): NSL204                                          -J310
GLO2123I GLOJIVP1,AVS,,SYSUT2,NSMS.GLO.TEST.DATA                                           -J310
RKTSW01I            AVS                25     00  00:00:00.09  00:00:00.07  00:00:00.00  00:00:00.02
RKTSW01I            DELETED            26     00  00:00:00.08  00:00:00.06  00:00:00.00  00:00:00.02
RKTSW01I            VPRIMRED           27     00  00:00:00.25  00:00:00.25  00:00:00.00  00:00:00.00
GLO3100I SETTING SECONDARY ALLOCATION TO 1  CYLS -J310
GLO3123I GLOJIVP1,VSECALC,,SYS00005,RSTEST.VSAM.TEST                                       -J310
RKTSW01I            VSECALC            28     00  00:00:00.32  00:00:00.31  00:00:00.00  00:00:00.01
GLO3103I EXTENT-MATCHING DATA  SECONDARY ALLOCATION TO 291  CYLS -J310
GLO3123I GLOJIVP1,VSECBST,,SYS00007,RSTEST.VSAM.TEST                                       -J310
RKTSW01I            VSECBST            29     00  00:00:00.31  00:00:00.30  00:00:00.00  00:00:00.01
GLO3109I REDUCING DATA  SECONDARY ALLOCATION TO 291  CYLS -J310
GLO3123I GLOJIVP1,VSECRED,,SYS00009,RSTEST.VSAM.TEST                                       -J310
RKTSW01I            VSECRED            30     00  00:00:00.30  00:00:00.30  00:00:00.00  00:00:00.00
GLO3128I DATA  SUCCESSFULLY EXTENDED TO VOLUME NSL204 -J310
GLO3123I GLOJIVP1,VVOLADD,,SYS00013,NSMS.VSAM.TEST                                         -J310
RKTSW01I            VVOLADD            31     00  00:00:00.23  00:00:00.23  00:00:00.00  00:00:00.00
GLO3148I AVS HAS SUCCESSFULLY SELECTED ALL VOLUMES-DATA  -J310
GLO3156I VOLSER(S): NSL204                                          -J310
GLO3123I GLOJIVP1,VAVS,,,NSMS.VSAM.TEST                                                    -J310
GLO3148I AVS HAS SUCCESSFULLY SELECTED ALL VOLUMES-INDEX -J310
GLO3156I VOLSER(S): NSL204                                          -J310
GLO3123I GLOJIVP1,VAVS,,,NSMS.VSAM.TEST                                                    -J310
RKTSW01I            VAVS               32     00  00:00:00.17  00:00:00.17  00:00:00.00  00:00:00.00
RKTSW01I            NCATCRE            33     00  00:00:00.07  00:00:00.06  00:00:00.00  00:00:00.01
GLO2084I DATA SET ON VOLSER NSL201 SCRATCHED AT ALLOCATION TIME -J310
GLO2123I GLOJIVP1,NCATSCR,,SYSUT7,NSMS.GLO.TEST.DATA -J310
RKTSW01I            NCATSCR            34     00  00:00:00.08  00:00:00.06  00:00:00.00  00:00:00.02
RKTSW01I            DELETE10           35     00  00:00:00.08  00:00:00.06  00:00:00.00  00:00:00.02
IEF377I GLOJIVP1 NOTCAT2  980
        NSMS.GLO.TEST.DATA NOT CATLGD 2
GLO2051I JOB SET TO FAIL DUE TO NOT CATLGD 2 ON VOLSER NSL200 -J310
GLO2123I GLOJIVP1,NOTCAT2,,SYSUT5,NSMS.GLO.TEST.DATA -J310
RKTSW01I            NOTCAT2            36     00  00:00:00.08  00:00:00.06  00:00:00.00  00:00:00.02
IEF378I GLOJIVP1 NOTCAT2 - JOB FAILED  984
        CATALOG DISPOSITION ERROR
RKTSW01I   GLOJIVP1 JOB TOTALS:                   00:00:04.96  00:00:03.82  00:00:00.04  00:00:01.10
$HASP395 GLOJIVP1 ENDED
```

Figure 5. Example of successful GLOJIVP1 results

Chapter 5. Rule definitions

After you install and configure Advanced Allocation Management, the next step is to define Advanced Allocation Management processing for your installation. For example, you might want Advanced Allocation Management to perform processing only for jobs that reference certain data, or that have specific job name characteristics.

Rule definitions control the actions that Advanced Allocation Management takes when an x37 type abend or other out-of-space condition, is about to occur. Rule definition syntax consists of keywords and keyword values that are grouped into functions and selection criteria by the DEFAULTS, INCLUDE, and EXCLUDE keywords.

The DEFAULTS keyword establishes the default settings for the Advanced Allocation Management functions and selection criteria and is required. The INCLUDE keyword specifies the items to select for processing and the functions to invoke for the selected items. The EXCLUDE statements specify the items to exclude from processing.

Note: Advanced Allocation Management does not perform processing unless you specify at least one INCLUDE statement in the rule definitions.

The **ACTIVATE** command causes the contents of the specified rule definitions member to become the active set of definitions that Advanced Allocation Management uses during processing. This feature enables dynamic modification of selection criteria and actions at any time without requiring a shutdown or quiesce of the product.

The following example shows the result of a successful activation of a rule definitions member from the **ACTIVATE** command as it is displayed in the JES SYSMSG output:

```
GLO4199I                1 * RULE DEFINITIONS
GLO4199I                2 DEFAULTS PRIM_REDUCE=ENABLE
GLO4199I                3 PRIM_REDUCE_LIM=50
GLO4199I                4 PRIM_REDUCE_INC=10
GLO4199I                5
GLO4199I                6 INCLUDE JOBNAME=ABCJOB2
GLO4199I                7 INCLUDE JOBNAME=ABCJOB3
GLO4199I                8 PRIM_REDUCE_INC=05
GLO4199I                9 INCLUDE JOBNAME=ABCJOB1
GLO4105I RULE DEFINITIONS SUCCESSFULLY ACTIVATED -AOPT
```

If an error occurs during the activation process, review the messages in the JES SYSMSG output to help determine the cause of the error.

Rule definition processing

After you activate the rule definitions, Advanced Allocation Management evaluates rule definitions separately at each of the following points in allocation processing (primarily to provide for volume-based selection criteria).

1. At the initial allocation of a data set on the first volume (to allow for the fact that volume-based selection criteria cannot be performed at this point in the allocation because the volume is not known unless it is explicitly defined).

2. At the first secondary allocation of a data set (at which time volume-based information is known).
3. At the first allocation on a new volume (to account for volume-based selection criteria for the new volume).

Processing order

Advanced Allocation Management processes rule definition syntax in top-down order. Once Advanced Allocation Management finds an item that matches a selection criteria, it exits and does not process the remaining statements in the rule definitions member unless `CONTINUE=ENABLE` is specified.

Consider the following rule definition syntax:

```
* RULE DEFINITIONS
DEFAULTS PRIM_REDUCE=ENABLE       * Default settings
PRIM_REDUCE_LIM=50
PRIM_REDUCE_INC=10
*
EXCLUDE  JOBNAME=ABCJOB2          * Item to exclude from processing
INCLUDE  JOBNAME=ABCJOB*          * Item to include in processing
```

Advanced Allocation Management processes the syntax as follows:

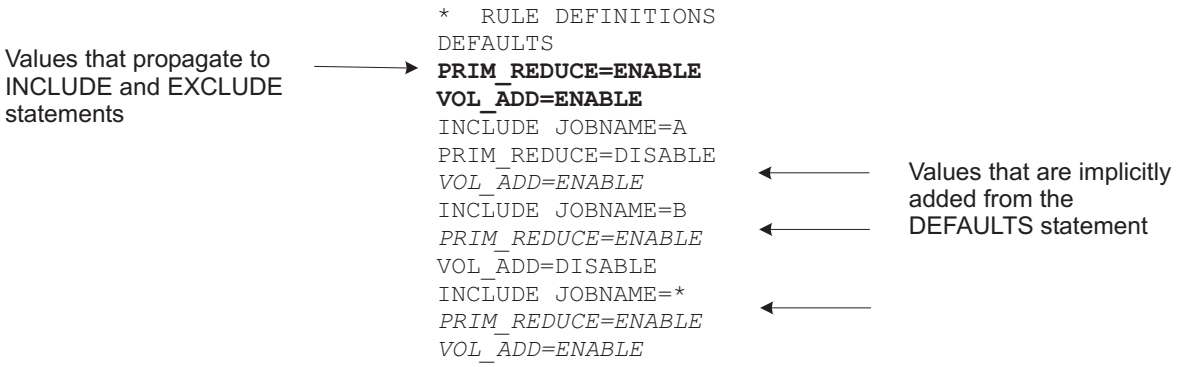
- The DEFAULTS keyword specifies that Advanced Allocation Management performs the PRIM_REDUCE function for the items that are included in its processing.
- The EXCLUDE keyword specifies that jobs named ABCJOB2 are excluded from Advanced Allocation Management processing.
- The INCLUDE keyword specifies that any job with a job name that begins with the characters "ABCJOB" is included in Advanced Allocation Management processing. Therefore, jobs that have names such as ABCJOB1 or ABCJOBA, are processed

Figure 6 on page 43 provides an overview of rule definition processing.

Rule Definitions

```
*  RULE DEFINITIONS
DEFAULTS
PRIM_REDUCE=ENABLE
VOL_ADD=ENABLE
INCLUDE JOBNAME=A
PRIM_REDUCE=DISABLE
INCLUDE JOBNAME=B
VOL_ADD=DISABLE
INCLUDE JOBNAME=*
```

In the first phase of processing each INCLUDE and EXCLUDE statement is processed and any value that is not specified on an INCLUDE or EXCLUDE is implicitly added from the DEFAULTS statement.



In the second phase, the first matching RULEDEF is found that matches all of the specified selection criteria. All of the explicitly or implicitly specified values from that RULEDEF become the base set of values that are used.

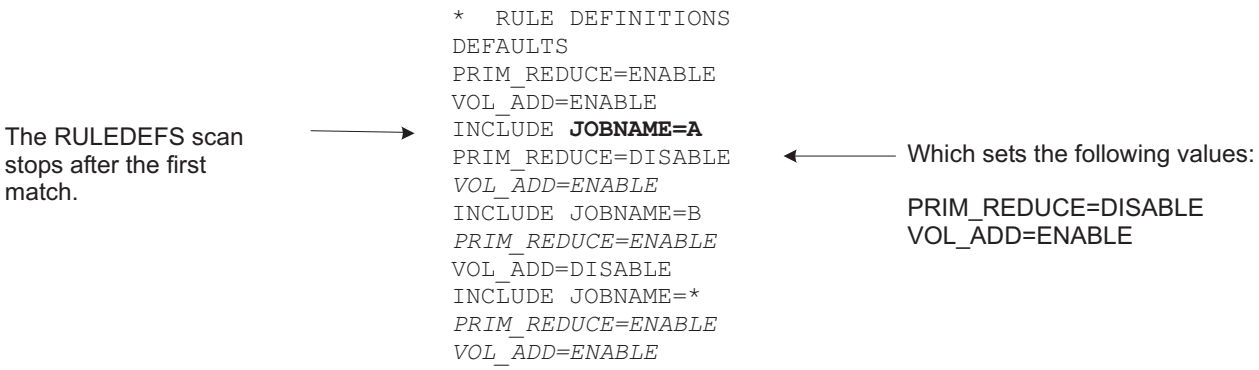


Figure 6. Advanced Allocation Management rule definition processing

CONTINUE=ENABLE rule definition processing

Specifying CONTINUE=ENABLE adds the capability to continue processing rule definitions to look for more matches. If more matches are found, Advanced Allocation Management replaces the functions it has accumulated thus far with the

functions that are explicitly specified on the newly matching rule definition or that are implicitly specified on the matching rule definition that is inherited from the DEFAULTS statement. CONTINUE=ENABLE does not modify the DEFAULTS statement functionality. For more information, see "Continue rule definition processing (CONTINUE)" on page 66.

Figure 7 provides an overview of CONTINUE=ENABLE rule definition processing.

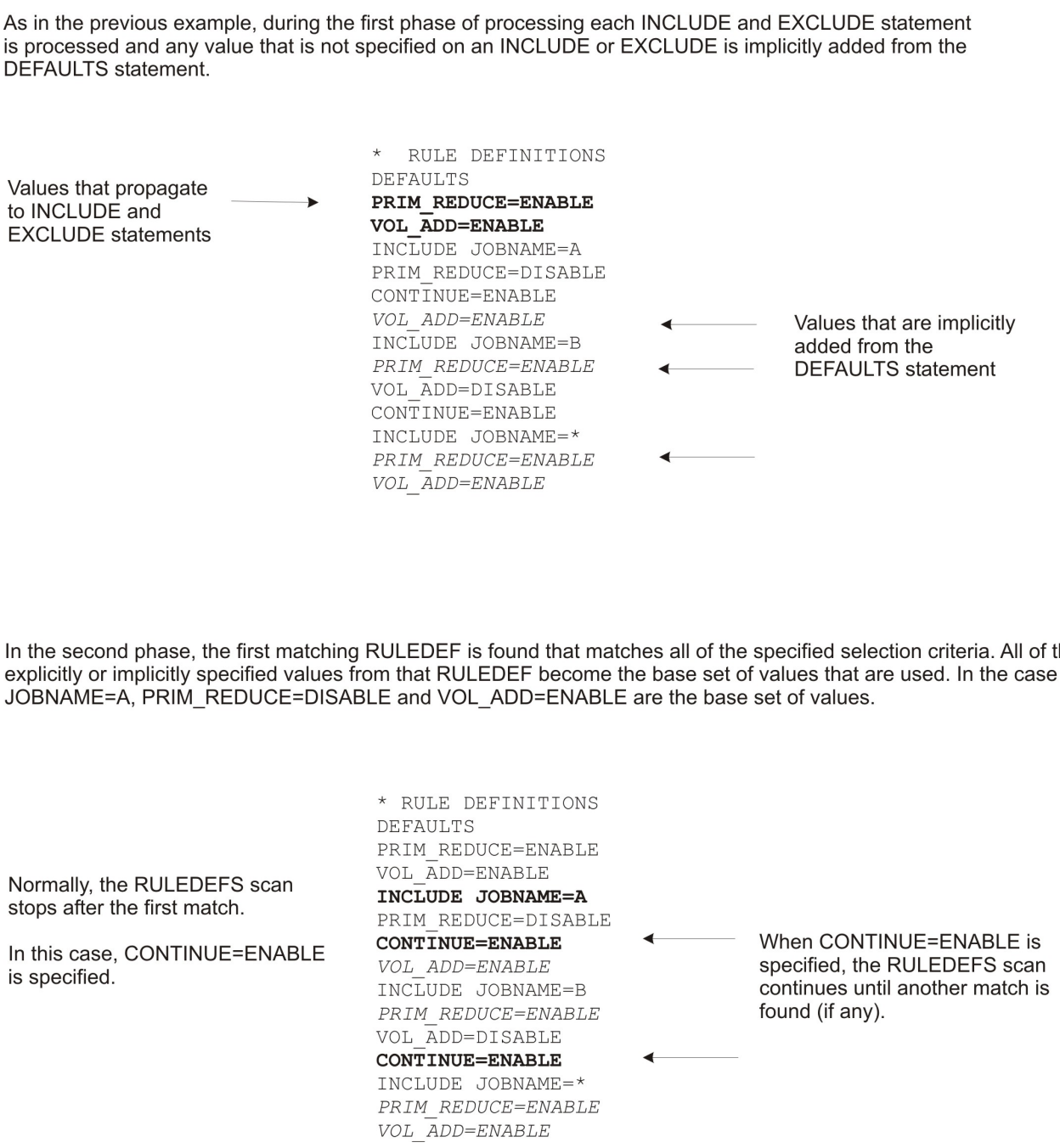


Figure 7. Advanced Allocation Management rule definition processing when CONTINUE=ENABLE is specified

If a match is found, the values that are explicitly or implicitly specified for functions on that RULEDEF overwrite the previous values.

In the example, when processing the RULEDEFS for a job that has a job name of B, the initial match occurs on the second RULEDEF, which establishes the base set of values of PRIM_REDUCE=ENABLE and VOL_ADD=DISABLE. Then, because `CONTINUE=ENABLE` is specified, the scan continues and finds a second match on JOBNAME=*, which causes the implicit values of PRIM_REDUCE=ENABLE and VOL_ADD=ENABLE to override the base set of values of PRIM_REDUCE=ENABLE and VOL_ADD=DISABLE.

How allocation attributes are used during selection criteria processing

Some values can be specified in the SMS data class and in the JCL. When the selection criteria are processed, Advanced Allocation Management uses the value that the system uses, as determined by z/OS rules of precedence.

For example, if an allocation attribute is specified in the SMS data class and in the JCL or IDCAMS equivalent, the system uses the JCL value (unless in z/OS 1.10 the SMS data class Override Space value is specified, in which case the data class space-related values, data control interval size, and CI and CA freespace values are used).

For more information about the Override Space parameter, see the *z/OS V1R10.0 DFSMS Storage Administration Reference for DFSMSdfp, DFSMSdss, DFSMShsm*.

Creating rule definitions

Implement simple rule definitions. Gradually add selection criteria and functions as needed.

To create a rule definitions member, complete the following steps:

1. Create a copy of the sample rule definitions member (RULEDEFS) from the control data set that was generated during the installation and configuration process.
2. Use the options on the Tivoli Advanced Allocation Management Control Member Management panel (preferred) or the ISPF EDIT function to perform the following steps:
   a. Confirm that the first line of the rule definitions member is the comment:
      ```
      * RULE DEFINITIONS
      ```
   b. Specify default processing. For more information, see "DEFAULTS" on page 47.
   c. Specify the selection criteria that defines the items you want to include or exclude from processing and the functions to perform when an item matches the selection criteria. For more information, see "Rule definition syntax" on page 46.

      Important: By default, Advanced Allocation Management processes items only when specifically directed to do so. If you do not create INCLUDE statements, Advanced Allocation Management does not perform processing. For more information, see "INCLUDE" on page 49.
3. Use a valid member name to save the member. All control data set member names must adhere to standard PDS member name conventions.

Note: The names of rule definitions members can be specified as parameters on operator commands.

4. Activate the rule definitions. For more information, see "Activating rule definitions."

Activating rule definitions

To activate rule definitions, use one of the following methods.

- Use the **Active Rule Definition** option on the Tivoli Advanced Allocation Management Subsystem Settings panel. For more information, see "Viewing and updating subsystem options" on page 278.
- Issue the **ACTIVATE** operator command whenever the Advanced Allocation Management started task is running. Use the following format:

  MODIFY *stcname*,ACTIVATE RULEDEFS *member*

  Where *stcname* is the name of the started task, and *member* is the name of the rule definitions member. For example: `ACTIVATE RULEDEFS RULESET1`

  Note: If you do not specify a member name on the **ACTIVATE** command, the member name RULEDEFS is used.

Displaying active rule definitions

Use the **DISPLAY** command to obtain information about active rule definitions, including the member name and the date and time of activation.

To display the active rule definitions, issue the following command from the operator console:

MODIFY *stcname*,DISPLAY RULEDEFS
Where *stcname* is the name of the started task.

Rule definition syntax

Use rule definition syntax to define Advanced Allocation Management processing. The DEFAULTS, INCLUDE, and EXCLUDE keywords define the three types of rules that control Advanced Allocation Management processing.

Syntax diagram

The following diagram shows an overview of the rule definition syntax.

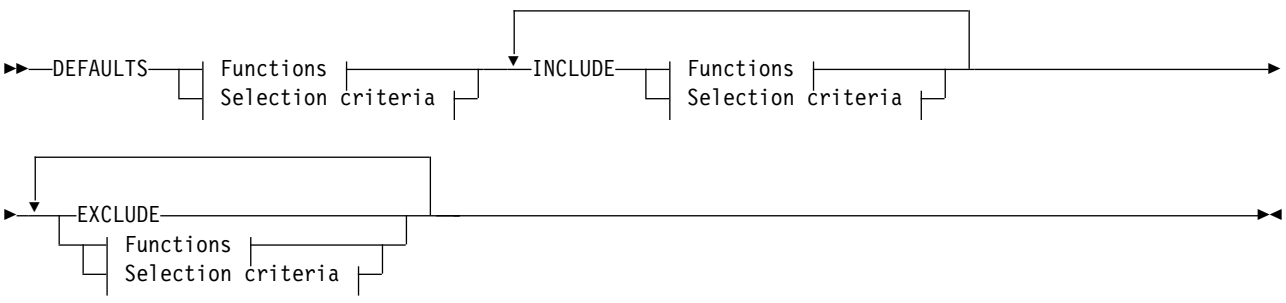


Figure 8. Rule definition syntax

For rule definition syntax, keywords can be specified in columns 1 - 72 inclusive. The maximum record length for the Advanced Allocation Management control data set is 80 bytes. Except for lists of items, a keyword and its operands must be contained within a single record.

Main keywords

Main keywords define the three types of rules that control Advanced Allocation Management processing.

Note: The underlined portion of the keyword indicates the minimally acceptable abbreviation.

DEFAULTS

The first keyword in the rule definitions member after the required comment that identifies the member type, must be the DEFAULTS keyword.

The DEFAULTS keyword establishes the default settings for the Advanced Allocation Management functions and selection criteria. The default settings remain active until you use the INCLUDE keyword to override them with an equivalent specification.

The DEFAULTS keyword can be followed by any number of INCLUDE and EXCLUDE keywords. For example:

```
DEFAULTS PRIM_REDUCE=ENABLE
         VOL_ADD=ENABLE
 *
         INCLUDE  JOBNAME=A
         PRIM_REDUCE=DISABLE
         INCLUDE JOBNAME=B
         VOL_ADD=DISABLE
         INCLUDE JOBNAME=*
```

The DEFAULTS statement propagates the options to the INCLUDE and EXCLUDE statements that follow, which eliminates the need to specify each option on every INCLUDE or EXCLUDE statement. Therefore, the following set of statements is equivalent to the statements in the previous example:

```
DEFAULTS PRIM_REDUCE=ENABLE
         VOL_ADD=ENABLE
 *
         INCLUDE JOBNAME=A
         PRIM_REDUCE=DISABLE
         VOL_ADD=ENABLE
         INCLUDE JOBNAME=B
         PRIM_REDUCE=ENABLE
         VOL_ADD=DISABLE
         INCLUDE JOBNAME=*
         PRIM_REDUCE=ENABLE
         VOL_ADD=ENABLE
```

Syntax

DEFAULTS

Example: Establish default settings for functions

In the following example, the DEFAULTS keyword establishes the default settings for the PRIM_REDUCE, PRIM_REDUCE_LIM, and PRIM_REDUCE_INC functions.

```
DEFAULTS PRIM_REDUCE=ENABLE
         PRIM_REDUCE_LIM=50
         PRIM_REDUCE_INC=10
*
INCLUDE  JOBNAME=ABCDJOB2
INCLUDE  JOBNAME=ABCDJOB3
INCLUDE  JOBNAME=ABCDJOB1
```

Advanced Allocation Management applies the PRIM_REDUCE, PRIM_REDUCE_LIM, and PRIM_REDUCE_INC functions to the INCLUDE statements.

The INCLUDE statements indicate that Advanced Allocation Management is to include jobs that have a job name of ABCDJOB2, ABCDJOB3, or ABCDJOB1, in its processing.

Note: Advanced Allocation Management automatically places any function that is specified on the DEFAULTS statement on an INCLUDE statement *unless you explicitly specify a function on an INCLUDE statement* as shown in the next example.

Example: Override a default function

If you specify a function on an INCLUDE statement, the value that is specified on the INCLUDE statement overrides the value that is specified on the DEFAULTS statement.

For example, if you set PRIM_REDUCE=ENABLE using the DEFAULTS statement, the PRIM_REDUCE function remains enabled until you specify PRIM_REDUCE=DISABLE on an INCLUDE statement:

```
DEFAULTS
      PRIM_REDUCE=ENABLE
      SEC_REDUCE=ENABLE
INCLUDE
      JOBNAME=JJHTEST1     *PRIM_REDUCE is enabled for this include statement
INCLUDE                    *(PRIM_REDUCE=ENABLE is in effect)
      JOBNAME=JJHTEST2
      PRIM_REDUCE=DISABLE  *(override)
INCLUDE
      JOBNAME=JJHTEST3     *PRIM_REDUCE is enabled for this include statement
                           *(PRIM_REDUCE=ENABLE is in effect)
```

Note: Any selection criteria or function keyword that is *not* specified on an INCLUDE or EXCLUDE statement inherits the value that is specified on the DEFAULTS statement.

Example: Establish default settings for a function and selection criteria

In the following example, the DEFAULTS keyword establishes the default settings for the MESSAGES, MSG_LEVEL, and MSG_DESTINATION functions and the JOBTYPE selection criteria to help ensure that no started task or TSO user was processed by Advanced Allocation Management. The SEC_BEST function is disabled:

```
DEFAULTS MESSAGES=ENABLE
         MSG_LEVEL=ALL
         MSG_DESTINATION=JES
         SEC_BEST=DISABLE
         JOBTYPE=JOB
```

INCLUDE

The INCLUDE keyword specifies the items to select for processing and the functions to invoke for the selected items.

Note: Advanced Allocation Management does not perform processing unless you specify at least one INCLUDE statement in the rule definitions.

Syntax

INCLUDE

Usage and processing considerations

- To evaluate each selection criterion separately, specify each item on a separate INCLUDE statement.

  When Advanced Allocation Management finds a match, it stops checking for more matches and performs processing on the selected item.

  Note: If the CONTINUE function is specified, Advanced Allocation Management continues to check and act on other matching rule definitions. For more information, see "Continue rule definition processing (CONTINUE)" on page 66.
- To evaluate multiple selection criteria, specify each item on one INCLUDE statement.

  When all of the selection criteria match, Advanced Allocation Management performs processing.

Note:

- The function statement that is placed immediately after an INCLUDE statement applies only to that INCLUDE statement.
- INCLUDE statements are processed in top-down order; therefore list the INCLUDE statements from most specific to least specific. For example:

  ```
  DEFAULTS VOL_ADD=ENABLE
  INCLUDE JOBNAME=HSMPDO*
  INCLUDE DDNAME=SORTWK*
  VOL_ADD=DISABLE
  ```

  Because the `INCLUDE JOBNAME=HSMPDO*` statement was placed before the `INCLUDE DDNAME=SORTWK*` statement, Advanced Allocation Management might process the VOL_ADD statement for a SORTWK* DD if it is written to by a job that has a job name where the first six characters are HSMPDO. This situation can occur because there are potentially many ddnames for jobs that begin with HSMPDO*, which is less specific.

  To ensure VOL_ADD=DISABLE is applied to all processing for any sort work DD regardless of the job name, place the INCLUDE statement for SORTWK* (more specific) before the INCLUDE statement for HSMPDO* (less specific).

Example: Separate INCLUDE statements

When you include multiple selection criteria and you want each selection criterion to be evaluated separately, specify each item that you want to include in Advanced Allocation Management processing on a separate INCLUDE statement. For example:

```
INCLUDE JOBCLASS=A
INCLUDE DSORG=PS
INCLUDE DSNAME=DISK*
INCLUDE DSNAME=WORK*
```

In the example, once Advanced Allocation Management finds a match, it stops looking for more matches. This means that Advanced Allocation Management processes all jobs that have a job class of A. It also processes data sets that have a data set organization of physical sequential, or all data sets that have a data set name that begins with DISK, or all data sets with a name that begins with WORK. Only *one* of the selection criteria must match for Advanced Allocation Management to perform processing.

Example: Separate INCLUDE statements followed by a function

The following example shows separate INCLUDE statements that are followed by a function:

```
INCLUDE JOBCLASS=A
INCLUDE DSORG=PS
DATASET_INIT=ENABLE
```

The function statement that is placed immediately after an INCLUDE statement applies only to that INCLUDE statement. Therefore, in the example, the DATASET_INIT function is enabled for data sets that have a DSORG of PS. The function is not enabled for jobs that have a job class of A, unless the function is also enabled by the DEFAULTS statement.

Remember: The DEFAULTS keyword establishes the default settings for the Advanced Allocation Management function and selection criteria.

Example: Separate INCLUDE statements followed by a function (DSNAME selection criteria)

The following example shows separate INCLUDE statements that specify data set names. The INCLUDE statements are followed by a function:

```
INCLUDE DSNAME=DISK*
INCLUDE DSNAME=WORK*
INCLUDE DSNAME=DSKABC*
INCLUDE DSNAME=WRKABC*
DATASET_INIT=ENABLE
```

As in “Example: Separate INCLUDE statements followed by a function,” the function statement that is placed immediately after an INCLUDE statement applies only to that INCLUDE statement.

Therefore, in the example, the DATASET_INIT function is enabled for data set names that begin with WRKABC. The DATASET_INIT function is not enabled for data set names that begin with DISK, WORK, or DSKABC, unless the DATASET_INIT function is also enabled by the DEFAULTS statement.

Example: Apply a function to INCLUDE statements

The following example shows how to apply the DATASET_INIT function to the JOBCLASS INCLUDE statement and the DSORG INCLUDE statement that is shown in “Example: Separate INCLUDE statements followed by a function”:

```
INCLUDE JOBCLASS=A
DATASET_INIT =ENABLE
INCLUDE DSORG=PS
DATASET_INIT =ENABLE
```

In the example, the DATASET_INIT function is enabled for JOBCLASS=A or DSORG=PS, even if DATASET_INIT is not enabled by the DEFAULTS statement.

Example: Apply a function to an INCLUDE statement (DSNAME selection criteria)

The following example shows how to apply the DATASET_INIT function to a list of INCLUDE DSNAME selection criteria statements as shown in "Example: Separate INCLUDE statements followed by a function (DSNAME selection criteria)" on page 50:

```
INCLUDE DSNAME=DISK*
DATASET_INIT=ENABLE
INCLUDE DSNAME=WORK*
DATASET_INIT=ENABLE
INCLUDE DSNAME=DSKABC*
DATASET_INIT=ENABLE
INCLUDE DSNAME=WRKABC*
DATASET_INIT=ENABLE
```

In the example, the DATASET_INIT function is enabled for the list of INCLUDE DSNAME selection criteria, even if DATASET_INIT is not enabled by the DEFAULTS statement.

Example: Specify selection criteria on a single INCLUDE statement

When you include multiple selection criteria on a single INCLUDE statement, *all* of the selection criteria that are specified on the INCLUDE statement must match for Advanced Allocation Management to perform the functions that are associated with that INCLUDE statement. For example:

```
INCLUDE JOBCLASS=A
        DSORG=PS
DATASET_INIT=ENABLE
```

In the example, Advanced Allocation Management processes all jobs that have a job class of A where the data set organization is physical sequential. All criteria (JOBCLASS=A and DSORG=PS) must be met to be included in processing.

Example: Specify selection criteria on a single INCLUDE statement (multiple job names)

The following example shows the specification of selection criteria on a single INCLUDE statement:

```
INCLUDE JOBNAME=(A,B,C)
        DSORG=PS
DATASET_INIT=ENABLE
```

In the example, a job that has a job name of A, B, or C and that has a DSORG of physical sequential, has the DATASET_INIT function enabled.

Example: Specify selection criteria on a single INCLUDE statement (lists)

Some selection criteria keywords support the specification of a list of items, such as data set names. When you include multiple criteria for these selection criteria keywords, you can specify each item that you want to include in Advanced Allocation Management processing on a single INCLUDE statement using the following format. For example:

```
INCLUDE DSNAME=(DISK*,DSKABC*,WORK*,WRKABC*,PABC.CDEF.*)
DATASET_INIT=ENABLE
```

In the example, Advanced Allocation Management performs the associated function when the data set name matches *any* of the data set names in the DSNAME list even if the DATASET_INIT function is not enabled by the DEFAULTS statement.

EXCLUDE

The EXCLUDE keyword specifies the items to exclude from Advanced Allocation Management processing.

Syntax

EXCLUDE

Usage and processing considerations

You can specify selection criteria on EXCLUDE statements in the following ways:

- Specify each selection criterion on a separate EXCLUDE statement.
- Specify multiple selection criteria on a single EXCLUDE statement.

Note:

- Because Advanced Allocation Management processes rule definitions in top-down order, place EXCLUDE statements before INCLUDE statements. For an example, see "Example: Always exclude SYS1.* and SYS2.* data sets" on page 53.
- Function-specific keywords are permitted for the EXCLUDE keyword, but are ignored. Therefore, you can easily switch a rule definition between an INCLUDE and an EXCLUDE without specifying the function keywords again.

Example: Establish the default settings for the PRIM_REDUCE, PRIM_REDUCE_LIM, and PRIM_REDUCE_INC functions

In the following example, the DEFAULTS keyword establishes the default settings for the PRIM_REDUCE, PRIM_REDUCE_LIM, and PRIM_REDUCE_INC function keywords. The EXCLUDE statements that follow the defaults instruct Advanced Allocation Management to exclude jobs that have job names of ABCDJOB2 or ABCDJOB3 from processing. Other jobs that have a job name that begins with ABCDJOB are included in processing because the INCLUDE statement specifies that Advanced Allocation Management is to include jobs that have job names that begin with ABCDJOB.

```
DEFAULTS PRIM_REDUCE=ENABLE
         PRIM_REDUCE_LIM=50
         PRIM_REDUCE_INC=10
```

```
 *
 EXCLUDE  JOBNAME=ABCDJOB2
 EXCLUDE  JOBNAME=ABCDJOB3
 INCLUDE  JOBNAME=ABCDJOB*
```

Example: Exclude specific jobs

In the following example, the EXCLUDE statements that follow the DEFAULTS statement indicate that Advanced Allocation Management is to exclude jobs that have a job name of ABC1 or TEST from processing. Jobs that have a job name that begins with ABCDJOB are included in processing.

```
DEFAULTS PRIM_REDUCE=ENABLE
         PRIM_REDUCE_LIM=50
         PRIM_REDUCE_INC=10
 *
 EXCLUDE  JOBNAME=ABC1
 EXCLUDE  JOBNAME=TEST
 INCLUDE  JOBNAME=ABCDJOB*
```

Example: Always exclude SYS1.* and SYS2.* data sets

In the following example, Advanced Allocation Management always excludes `SYS1.*` and `SYS2.*` data sets from processing. Placing the EXCLUDE statements before all of the INCLUDE statements prevents Advanced Allocation Management from processing an `SYS1.*` or `SYS2.*` data set when it is accessed by something for which there is an INCLUDE statement, for example: `PROG=IKJ*`.

```
EXCLUDE  DSNAME=SYS1.*
EXCLUDE  DSNAME=SYS2.*
INCLUDE  PROG=IKJ*
INCLUDE  JOBNAME=ABCDJOB*
```

Example: EXCLUDE statements and functions

In the following example, Advanced Allocation Management always excludes `SYS1.*` and `SYS2.*` data sets from processing. The first occurrence of the DATASET_INIT function is ignored for the EXCLUDE statements. The second occurrence of the DATASET_INIT function is enabled for the `INCLUDE JOBNAME=ABCDJOB*` statement. The DATASET_INIT function is not enabled for the `INCLUDE PROG=IKJ*` statement unless the DATASET_INIT function is also enabled by the DEFAULTS statement.

```
   EXCLUDE  DSNAME=SYS1.*
   EXCLUDE  DSNAME=SYS2.*
   INCLUDE  PROG=IKJ*
   INCLUDE  JOBNAME=ABCDJOB*
   DATASET_INIT=ENABLE
```

Functions

Function keywords define the functions that Advanced Allocation Management performs, and the values it is to use for those functions when it finds a match using the associated selection criteria.

The following table shows common issues, the corresponding abends (if applicable), and the functions to use.

Table 4. Functions to use based on issue and abend type

Type of issue and abend	Functions
During end-of-volume processing, no more space was available on the volume, the data set reached 16 extents, or the VTOC was full. (IEC030I, B37 abend.)	Increase, reduce, or adjust secondary to best fit; dynamically add volumes. For more information, see "Reduce secondary allocation to largest available free extent (SEC_BEST)" on page 95 and "Dynamically add volumes (VOL_ADD)" on page 108.
A data set that was opened for output used all of the primary space and no secondary space was requested. (IEC031I, D37 abend.)	Add a secondary allocation. For more information, see "Add a secondary allocation (SEC_ALLOC)" on page 93.
A data set that was opened for output used all of the available space on the current volume and no more volumes were available. (IEC032I, E37 abend.)	Dynamically add volumes, reduce secondary, or adjust it to best fit. For more information, see "Dynamically add volumes (VOL_ADD)" on page 108 and "Reduce secondary allocation to largest available free extent (SEC_BEST)" on page 95.
In a DD statement, the SPACE parameter requested more tracks than were available on the requested direct-access volume. (IEF257I, JOB FAILED - JCL ERROR.)	Reduce primary. For more information, see "Controlled reduction of primary space (PRIM_REDUCE)" on page 85.
Ineffective use of existing storage resources; poor planning; arbitrary use of existing JCL.	Adjust secondary to best fit, release unused space. For more information, see "Reduce secondary allocation to largest available free extent (SEC_BEST)" on page 95.
An attempt was made to store directory information for a new member in a PDS using the MVS STOW service, and all of the directory blocks are full.	Dynamically increase the directory space within the partitioned data set. For more information, see "Dynamic PDS directory increase (PDS_DIR_INC)" on page 83.
An attempt was made to catalog a new data set, but the data set is cataloged on another volume.	Detect the attempt to catalog the duplicate data set and perform processing to prevent the potential use of the wrong data set. For more information, see "Process NOT CATLGD 2 errors (NOTCAT2)" on page 75.
Control the initial allocation of non-SMS volumes.	Use the Advanced Volume Selection keywords to select volumes from user-defined pools. For more information, see "Advanced Volume Selection (AVS)" on page 57.
Modify data set attributes to enforce standards.	Use the SET_ functions of Enhanced Allocation Management. For more information, see "Enhanced Allocation Management functions" on page 126.
Terminate allocation upon matching selection criteria.	Use the TERM_ALLOC function after matching on various selection criteria. For more information, see "Terminate an allocation (TERM_ALLOC)" on page 107.
Tune volume selection.	Specify the volume selection algorithm for initial and subsequent volume allocation. For more information, see "Advanced Volume Selection (AVS)" on page 57.

For more information, see "Function keywords" on page 57.

Determining the functions to include in rule definitions

Functions specify the actions that you want Advanced Allocation Management to perform using the selection criteria. To determine the functions to include in your rule definitions, review the following information.

Common issues and abends

Table 5 on page 55 shows common issues, the corresponding abends (if applicable), and the functions to use to avoid the issue.

Table 5. Functions based on issue and abend type

Type of issue and abend	Function
During end-of-volume processing, no more space was available on the volume, the data set reached 16 extents, or the VTOC was full. (IEC030I, B37 abend.)	Increase, reduce, or adjust secondary to best fit; dynamically add volumes.
A data set that was opened for output used all of the primary space and no secondary space was requested. (IEC031I, D37 abend.)	Add a secondary allocation.
A data set that was opened for output used all of the space available on the current volume and no more volumes were available. (IEC032I, E37 abend.)	Dynamically add volumes, reduce secondary, or adjust it to best fit.
In a DD statement, the SPACE parameter requested more tracks than were available on the requested direct access volume. (IEF257I, JOB FAILED - JCL ERROR.)	Reduce primary.
Ineffective use of existing storage resources; poor planning; arbitrary use of existing JCL.	Adjust secondary to best fit; release unused space.
An attempt was made to store directory information for a new member in a PDS (using the MVS STOW service) and all of the directory blocks are full.	Dynamically increase the directory space within the partitioned data set.
An attempt was made to catalog a new data set, but the data set is currently cataloged on another volume.	Detect the attempt to catalog the duplicate data set and perform processing to prevent the potential use of the wrong data set.
Control the initial allocation of non-SMS volumes.	Use the Advanced Volume Selection keywords to select volumes from user-defined pools.
Modify data set attributes to enforce standards.	Use the Enhanced Allocation Management SET_ functions.
Terminate allocation upon matching selection criteria.	Use the TERM_ALLOC function after matching on various selection criteria.
Tune volume selection.	Specify volume selection algorithm for initial and subsequent volume allocation.

Abend recovery functions

Table 6 shows Advanced Allocation Management abend recovery functions and their corresponding keywords.

Table 6. x37 Abend recovery functions (keyword reference)

Function	Keywords
Controlled reduction of primary space to satisfy allocation.	PRIM_REDUCE, PRIM_REDUCE_LIM, and PRIM_REDUCE_INC.
Dynamically add volumes at end-of-volume processing.	VOL_ADD, VOL_ADD_LIM, VOL_ADD_PROMPT, VOL_ADD_SGRP, VOL_ADD_UNIT, VOL_ADD_VGRP, VOL_ADD_VSER, and VOL_ADD_MOUNT_USAGE.

Table 6. x37 Abend recovery functions (keyword reference) (continued)

Function	Keywords
Recover from secondary allocation space failures by reducing the allocation amount to a value that is consistent with the current free space on the volume.	SEC_REDUCE, SEC_REDUCE_INC, and SEC_REDUCE_LIM.
Detect and dynamically recover from DIRECTORY FULL conditions.	PDS_DIR_INC.

Abend prevention and avoidance functions

Table 7 shows the Advanced Allocation Management abend prevention and avoidance functions and their corresponding keywords.

Note: Using abend prevention functions can result in improved processing time because, in general, abend prevention functions incur less processing overhead.

Table 7. x37 Abend prevention and avoidance functions (keyword reference)

Function	Keywords
Automatically add a secondary allocation if none is specified.	SEC_ALLOC and SEC_ALLOC_AMT.
Reduce secondary allocation to the largest available free extent.	SEC_BEST and SEC_BEST_LIM.
Increase the size of the secondary allocation as data set extends occur at a particular extent and at a particular volume number.	SEC_INC, SEC_INC_AMT, SEC_INC_VOLNO, and SEC_INC_XTNT.
Recover unused disk space by automatically releasing allocated, unused data set space when a data set is closed.	SPACE_RELEASE.
Detect the attempt to catalog a newly created data set that is already cataloged on another volume.	NOTCAT2, NOTCAT2_CHECK, NOTCAT2_DYNAL, NOTCAT2_QUALIFIER, NOTCAT2_QUALIFIER _POSITION, and NOTCAT2_SUFFIX.
Initialize a newly allocated, but not opened, non-VSAM data set by performing an OPEN/CLOSE sequence on it during step termination.	DATASET_INIT.

Allocate management functions

Table 8 shows Advanced Allocation Management allocate management functions and their corresponding keywords.

Table 8. Allocate management functions and corresponding keywords

Function	Corresponding keywords
Advanced volume selection	AVS, AVS_VGRP, AVS_UNIT, AVS_VSER, AVS_MOUNT_USAGE, AVS_ALGORITHM, and AVS_VGRP_SEARCH.

Table 8. Allocate management functions and corresponding keywords (continued)

Function	Corresponding keywords
Enhanced allocation management	Enhanced Allocation Management function keywords. Enhanced Allocation Management function keywords begin with the prefix "SET_".
Standards enforcement	TERM_ALLOC.

Function keywords

Function keywords specify the functions that Advanced Allocation Management performs and the values it should use for those functions, when it finds a match using the associated selection criteria.

Note:

1. In the keyword descriptions, the underlined portion of the keyword indicates the minimally acceptable abbreviation.
2. All functions that have an ENABLE or DISABLE option default to DISABLE if no DEFAULTS value is set (or if the function is not explicitly specified).
3. *user_var* indicates a user-defined variable.

Advanced Volume Selection (AVS)

Advanced Volume Selection enables Advanced Allocation Management to select the volumes for a non-SMS managed data set at initial allocation time. The AVS keyword specifies whether Advanced Volume Selection is enabled or disabled.

Syntax

AVS=ENABLE | DISABLE

Operands

ENABLE
Enable Advanced Volume Selection.

DISABLE
Disable Advanced Volume Selection.

Usage and processing considerations

To specify the volsers to search for a suitable volume, the AVS keyword requires at least one of the AVS_VGRP, AVS_UNIT, or AVS_VSER keywords.

VSAM considerations:

- AVS is performed at the component level for VSAM data sets using the algorithm you specify using the AVS_ALGORITHM keyword. For the BESTFIT, FIRST, and PERFORMANCE algorithms, the size of the extent to locate is determined as follows:

- With the space parameters defined at the CLUSTER level, the primary allocation quantity is used for the data and index components to search for space.
- With the space parameters defined separately at the DATA and INDEX levels, the respective primary quantity of each component is used to search for space.

- When the cluster definition contains volume serial numbers, AVS overlays user-specified values and truncates or extends the volume list, as needed.
- If no volumes are located using the requested AVS algorithm, the volume list is emptied and the cluster definition fails.

Default

DISABLE

Environments

ALLOCATE

Related keywords

AVS_ALGORITHM, AVS_MOUNT_USAGE, AVS_UNIT, AVS_VGRP, AVS_VGRP_SEARCH, and AVS_VSER.

Example

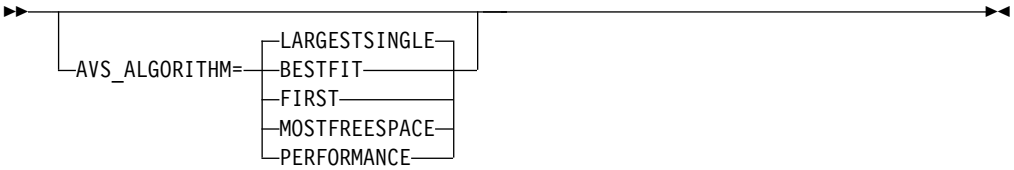
In the following example, AVS is set to ENABLE; therefore, Advanced Volume Selection is enabled. When an AVS is performed for a non-SMS-managed data set for job JJHTEST01, the volumes that Advanced Allocation Management considers are all of the volumes that have a mount usage attribute of STORAGE in the volume group named GROUP01:

```
INCLUDE JOBNAME=JJHTEST01
AVS=ENABLE
AVS_VGRP=GROUP01
AVS_MOUNT_USAGE=STORAGE
```

AVS_ALGORITHM

The AVS_ALGORITHM keyword specifies a volume selection algorithm to use when Advanced Volume Selection is performed.

Syntax



AVS_ALGORITHM=BESTFIT | FIRST | LARGESTSINGLE | MOSTFREESPACE | PERFORMANCE

Operands

BESTFIT
Select the volume (or volumes) with a single extent in which the requested amount is closest to the requested primary allocation quantity without being less than the requested primary allocation quantity.

FIRST
Select the first volume (or volumes) that have enough total freespace to satisfy the requested allocation amount.

Note: The allocation amount might not be able to be satisfied because the freespace that is needed might be spread across more than five extents, which is the normal limit within allocation for satisfying a space request. Therefore, you might need to specify PRIM_REDUCE to help make the allocation successful.

LARGESTSINGLE
Select the volume (or volumes) with the largest single free extent.

Note: Advanced Allocation Management selects the volume with the largest single extent. This does not mean that the volume has enough space to satisfy the requested amount, it indicates that it is the volume with the largest single extent. Therefore, you might need to use the PRIM_REDUCE function to handle those situations where the selected volume does not have adequate space available.

MOSTFREESPACE
Select the volume with the most available freespace. The most available freespace is defined as the total number of free cylinders or tracks, on the volume, depending upon the allocation unit that is used. If you specify cylinder allocation, the total number of free cylinders is considered. Otherwise, the total number of free tracks is considered.

PERFORMANCE
Select the volume with the lowest fragmentation index, giving preference to volumes that have enough available freespace to satisfy the allocation request.

Usage and processing considerations

- If more than one volume is to be selected as specified in the volume count, Advanced Allocation Management chooses multiple volumes, starting with the best candidate as defined by the algorithm that is chosen, down to the least eligible candidate.
- For the BESTFIT and FIRST algorithm, as the size of the extent must at least contain the requested primary quantity, it is possible that Advanced Allocation Management might not find enough volumes that are based on the algorithm to satisfy the volume count that is specified. In this case, the remaining volumes, up to the volume count specified, are chosen using the MOSTFREESPACE algorithm.
- When Advanced Allocation Management chooses initial volumes using AVS, the allocation request is converted to a specific volume request.

Tips:

The following suggestions are dependent upon sizes and patterns of data set creation and deletion and your results might vary.

- To fill volumes in sequence, leaving other volumes mostly empty for large allocations, consider specifying FIRST or BESTFIT. Specifying FIRST tends to fill the volumes in order of specification in the various volume groups.
- To distribute allocations evenly among all volumes in a volume group, consider specifying MOSTFREESPACE and LARGESTSINGLE. In most cases, specifying MOSTFREESPACE followed by LARGESTSINGLE tends to evenly distribute the allocations across the volumes that are listed in the volume groups.

Default

LARGESTSINGLE

Environments

ALLOCATE

Related keywords

AVS, AVS_MOUNT_USAGE, AVS_UNIT, AVS_VGRP, AVS_VGRP_SEARCH, and AVS_VSER.

Example

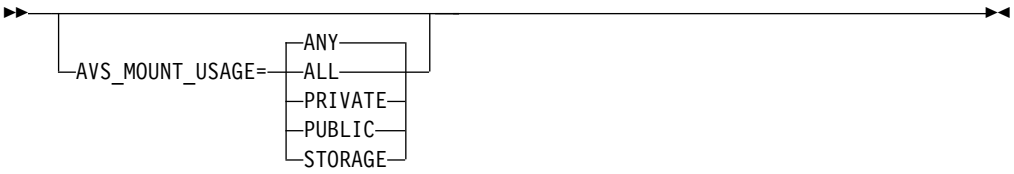
In the following example, the volume with most available freespace is selected:

```
AVS_ALGORITHM=MOSTFREESPACE
```

AVS_MOUNT_USAGE

The AVS_MOUNT_USAGE keyword limits volume selection to those volumes that match the requested mount usage attributes.

Syntax



AVS_MOUNT_USAGE=ALL | ANY | PRIVATE | PUBLIC | STORAGE

Operands

Specify one of the following operands:

ALL
All mount usage attributes are acceptable.

Note: Specifying ALL is the same as specifying ANY.

ANY
Any mount usage attribute is acceptable.

PRIVATE, PUBLIC, and STORAGE
The PRIVATE, PUBLIC, and STORAGE operands are equivalent to the mount usage attribute for a volume as defined in the Unit Control Block for a device and as defined by z/OS.

Usage and processing considerations

Only one operand can be specified. For example:

- Valid: AVS_MOUNT_USAGE=STORAGE
- Not valid: AVS_ADD_MOUNT_USAGE=STORAGE,PUBLIC

Default

ANY

Environments

ALLOCATE

Related keywords

AVS, AVS_ALGORITHM, AVS_UNIT, AVS_VGRP, AVS_VGRP_SEARCH, and AVS_VSER.

Example

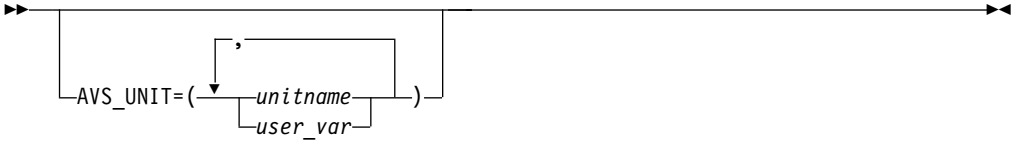
In the following example, when an AVS is performed for a non-SMS-managed data set for job JJHTEST01, the volumes that Advanced Allocation Management considers are all of the volumes that have a mount usage attribute of STORAGE in the volume group named GROUP01:

```
INCLUDE JOBNAME=JJHTEST01
      AVS=ENABLE
      AVS_VGRP=GROUP01
      AVS_MOUNT_USAGE=STORAGE
```

AVS_UNIT

The AVS_UNIT keyword specifies the units to search for eligible volumes when Advanced Volume Selection is performed.

Syntax



AVS_UNIT=(*unitname* | *user_var*, *unitname* | *user_var*,....)

Where:

unitname
is a unit to search for eligible volumes. Each unit name can be a maximum of eight characters in length.

Default

None.

Environments

ALLOCATE

Related keywords

AVS, AVS_ALGORITHM, AVS_MOUNT_USAGE, AVS_VGRP, AVS_VGRP_SEARCH, and AVS_VSER.

Example: Search esoteric groups

In the following example, Advanced Allocation Management searches the esoteric groups with the unit names of DISKBIG and DISKALT for eligible volumes:

```
AVS_UNIT=(DISKBIG,DISKALT)
```

Example: Search esoteric groups as defined by user-defined variables

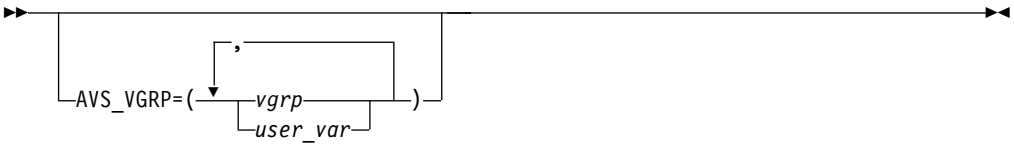
In the following example, Advanced Allocation Management searches the esoteric groups with the unit names as defined by the user-defined variables &UNITA, &UNITB, and &UNITC, for eligible volumes:

```
AVS_UNIT=(&UNITA,&UNITB,&UNITC)
```

AVS_VGRP

The AVS_VGRP keyword specifies the volume groups to search for eligible volumes when Advanced Volume Selection is performed.

Syntax



AVS_VGRP=(*vgrp* | *user_var*, *vgrp* | *user_var*,....)

Where:

vgrp
: is a volume group to search. Each volume group name can be a maximum of eight characters in length.

Default

None.

Environments

ALLOCATE

Related keywords

AVS, AVS_ALGORITHM, AVS_MOUNT_USAGE, AVS_UNIT, AVS_VGRP_SEARCH, and AVS_VSER.

Example: Search multiple volume groups

In the following example, Advanced Allocation Management searches the GROUP01, GROUP02, and GROUP03 volume groups for eligible volumes:

```
AVS_VGRP=(GROUP01,GROUP02,GROUP03)
```

Example: Search a volume group as defined by user-defined variables

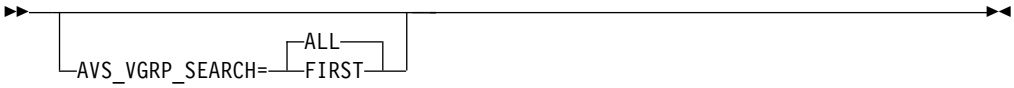
In following example, Advanced Allocation Management searches volume groups as defined by the user-defined variable &VOLGRP1, for eligible volumes:

```
AVS_VGRP=(&VOLGRP1)
```

AVS_VGRP_SEARCH

The AVS_VGRP_SEARCH keyword specifies the volume selection volume group search method to use when Advanced Volume Selection is performed.

Syntax



AVS_VGRP_SEARCH=ALL | FIRST

Operands

ALL
Search all volume groups for a suitable volume.

FIRST
Stop searching for volumes after the volume group that was searched contains a volume that meets the criterion that is established by the AVS_ALGORITHM keyword.

Note: Other than BESTFIT when all volumes that are requested are able to be satisfied using the BESTFIT algorithm, there is no guarantee that the requested space is available.

For example, if you specify `AVS_ALGORITHM= LARGESTSINGLE` and Advanced Allocation Management selects a volume with the largest single extent in a volume group, it does not mean that the volume has enough space to satisfy the requested amount, it means that it is the volume with the largest single extent. Therefore, you might need to use the PRIM_REDUCE function to handle situations where the selected volume does not have adequate space available.

Default

ALL

Environments

ALLOCATE

Related keywords

AVS, AVS_ALGORITHM, AVS_MOUNT_USAGE, AVS_UNIT, AVS_VGRP, and AVS_VSER.

Example

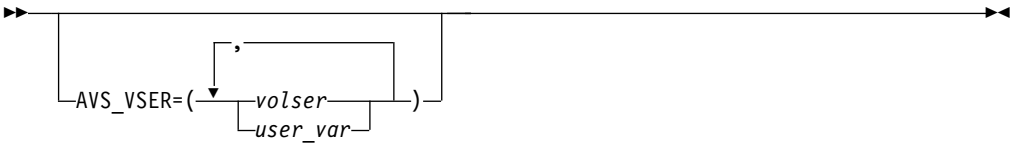
In the following example, Advanced Allocation Management stops searching for volumes if a suitable volume exists in the volume group in which it is searching:

```
AVS_VGRP_SEARCH=FIRST
```

AVS_VSER

The AVS_VSER keyword specifies the volsers to search for eligible volumes when Advanced Volume Selection is performed.

Syntax



AVS_VSER=(*volser* | *user_var*, *volser* | *user_var*,....)

Where:

volser
is the 1 - 6 character volume serial number to search for eligible volumes.

Default

None.

Environments

ALLOCATE

Related keywords

AVS, AVS_ALGORITHM, AVS_MOUNT_USAGE, AVS_UNIT, AVS_VGRP, and AVS_VGRP_SEARCH.

Example: Search multiple volsers

In the following example, Advanced Allocation Management searches the VOL100, VOL200, and VOL300 volsers for eligible volumes:

```
AVS_VSER=(V0L100,VOL200,VOL300)
```

Example: Search volsers as defined by user-defined variables

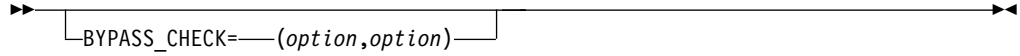
In the following example, Advanced Allocation Management searches the volsers as defined by the &VOL1 and &VOL2 user-defined variables, for eligible volumes:

```
AVS_VSER=(&V0L1,&VOL2)
```

Bypass integrity check processing (BYPASS_CHECK)

BYPASS_CHECK specifies whether to bypass integrity check processing.

Syntax



BYPASS_CHECK=(*option*,*option*)

Operands

Where *option* is one of the following values:

NOMULTOPEN
(Default.) Do not perform a VOL_ADD if multiple concurrent DCBs or ACBs are open against the data set within a step.

NOMULTDD
(Default.) Do not perform a VOL_ADD if multiple DDs point to the same data set name within a step.

MULTOPEN
During VOL_ADD processing, bypass the check for multiple concurrent OPENs against the same data set within a step.

MULTDD
During VOL_ADD processing, bypass the check for multiple DDs that point to the same data set name within a step.

Usage and processing considerations

Important: Use BYPASS_CHECK only in specific circumstances where you are certain that the specifications that bypass integrity checks will not cause problems for the program that is processing the data sets. In most cases, application programs should not use the BYPASS_CHECK options.

Generally, these circumstances (multiple concurrent opens against a data set or multiple DDs pointing to the same data set) can result in unexpected results, lost data, or program failures if a volume addition is performed against one of the allocations without the knowledge of the new volume addition being communicated in some manner to the other DDs or open DCBs or ACBs. Some system software packages are designed to handle multiple concurrent OPENs against a data set, multiple DDs that point to the same data set, or both. If you are certain that the program that is processing the data sets can handle these circumstances, you can specify the BYPASS_CHECK keyword to bypass the specific integrity checks.

You can specify up to two values for the BYPASS_CHECK function although it is counterproductive to specify both MULTOPEN and NOMULTOPEN or MULTDD and NOMULTDD at the same time. If you specify one of these combinations, Advanced Allocation Management performs the NOMULTOPEN, NOMULTDD option (default processing).

Default

NOMULTOPEN, NOMULTDD

Environments

EXTEND

Related keywords

VOL_ADD

Example: Bypass the check for multiple concurrent OPENs and multiple DDs

In the following example, during VOL_ADD processing, Advanced Allocation Management bypasses the check for multiple concurrent OPENs and multiple DDs against the same data set within a step.

```
BYPASS_CHECK=(MULTOPEN,MULTDD)
```

Example: Bypass check for multiple DDs; do not bypass check for multiple OPENs

In the following example, during VOL_ADD processing, Advanced Allocation Management does *not* bypass the check for multiple DDs (it *does* bypass the check for multiple OPENs).

```
BYPASS_CHECK=(NOMULTDD,MULTOPEN)
```

Example: Bypass the check for multiple DDs

In the following example, during VOL_ADD processing, Advanced Allocation Management bypasses the check for multiple DDs pointing to the same data set name within a step.

```
BYPASS_CHECK=(MULTDD)
```

Continue rule definition processing (CONTINUE)

The CONTINUE function directs Advanced Allocation Management to perform applicable actions for a matched rule definition and then continue to look for additional matching rule definitions.

Syntax

CONTINUE=ENABLE | DISABLE

Operands

ENABLE
Enable CONTINUE processing.

Note: If CONTINUE=ENABLE, more than one rule definition can match an allocation. When this situation occurs, only the final match is reported.

DISABLE
Disable CONTINUE processing.

Default

DISABLE

Environments

ACS, ALLOCATE, CATCHECK, EXTEND, STOW, and UNALLOCATE.

Example

In the following example, CONTINUE=ENABLE directs Advanced Allocation Management to change the unit name when UNIT_NAME=VIO is specified, and then continue checking and acting on other matching rule definitions:

```
RULE DEFINITIONS

DEFAULTS
*
INCLUDE
  UNIT_NAME=VIO
  SET_UNIT_NAME=SYSDA
INCLUDE
  UNIT_NAME=VIO
  SET_UNIT_NAME=SYSDA
  CONTINUE=ENABLE
INCLUDE
  PROGRAM=ADRDSSU
  PRIM_REDUCE=(DISABLE,DISABLE)
```

Note the following items:

- If the first rule definition occurs near the top of the RULEDEFS member when a request for VIO is encountered, it matches the rule definition, changes the unit name to SYSDA, and stops processing subsequent rule definitions. In this case, you might not want to stop processing at this point because the change is a global change and more specific changes might follow the definition.
- If you specify a DFDSS step (ADRDSSU) and SYSPRINT is directed to VIO, Advanced Allocation Management changes the specification to SYSDA and disables PRIM_REDUCE for all allocations within this job step.

Example

The following is an example of accumulating rule definition information:

```
INCLUDE DDNAME=SYSUT1
              PRIM_REDUCE=ENABLE
              MSG_LEVEL=RULEDEF
              CONTINUE=ENABLE
```

```
INCLUDE JOBNAME=JJHTEST1
               SEC_ALLOC=ENABLE
               MSG_LEVEL=EXCEPTIONS
```

The following behavior occurs using the rule definition:

1. For a data set with a DDNAME of SYSUT1 in a job with a job name of JJHTEST2, the PRIM_REDUCE=ENABLE and MSG_LEVEL=RULEDEF options are set because only the first rule definition matches.
2. For a data set with a DDNAME of SYSUT2 in a job with a job name of JJHTEST1, the SEC_ALLOC=ENABLE and MSG_LEVEL=EXCEPTIONS options are set because only the second rule definition matches.
3. For a data set with a DDNAME of SYSUT1 in a job with a job name of JJHTEST1, the PRIM_REDUCE=ENABLE, SEC_ALLOC=ENABLE, and MSG_LEVEL=EXCEPTIONS options are set because both rule definitions match. Because MSG_LEVEL is specified on both rule definitions, the last MSG_LEVEL specification overrides the first MSG_LEVEL specification.

For an overview of CONTINUE=ENABLE processing, see "Processing order" on page 42.

Initialize a newly allocated data set (DATASET_INIT)

The DATASET_INIT function initializes a newly allocated (but not opened) non-VSAM data set by performing an OPEN/CLOSE sequence on it during step termination. The initialization causes a valid end-of-file mark to be written to the data set, eliminating the possibility of input/output errors or invalid data being read by programs that attempt to read the empty data set. The initialization process also causes space release (if coded in the JCL) to be driven for the new data set, which you might want for certain applications.

The DATASET_INIT keyword specifies whether to initialize the data set in question.

Syntax

DATASET_INIT=ENABLE | DISABLE

Operands

ENABLE
Perform data set initialization.

DISABLE
Do not perform data set initialization.

Usage and processing considerations

To use the DATASET_INIT function, add the Z1_INTERCEPT parameter to the Advanced Allocation Management OPTIONS member and set it to ENABLE.

Note: Each time that you modify the Z1_INTERCEPT parameter setting within the OPTIONS member, you must IPL the system and then restart Advanced Allocation Management for the new setting to take effect.

For a data set to be eligible for initialization, the following criteria must be met:

- The data set must be non-VSAM.
- If the DSORG is coded in the JCL, it must be set to PS or PSU.
- The data set must have a volume that is assigned to it during allocation.
- The data set cannot be allocated to VIO.
- If the step is ending normally, the normal disposition in the JCL cannot be DELETE - DISP=(x,DELETE,x).
- If the step is ending abnormally, the abnormal disposition in the JCL cannot be DELETE - DISP=(x,x,DELETE).

Note: If a step is ending abnormally, MVS does not release space at data set close.

Default

DISABLE

Environments

UNALLOCATE

Example

In the following example, all newly allocated unopened data sets that match the required criteria and that are allocated by job ABC, are initialized:

```
RULE DEFINITIONS

DEFAULTS
*
INCLUDE JOBNAME=ABC
  DATASET_INIT=ENABLE
```

Display diagnostic information (DIAG)

The DIAG keyword specifies whether to display diagnostic information.

Syntax

DIAG=ENABLE | DISABLE

Note: Use the DIAG keyword only at the direction of IBM personnel.

Operands

ENABLE
Display diagnostic information.

DISABLE
Do not display diagnostic information.

Default

DISABLE

Environments

ACS, ALLOCATE, CATCHECK, EXTEND, STOW, and UNALLOCATE.

Example

In this example, DIAG is set to DISABLE; therefore, no diagnostic information is displayed:

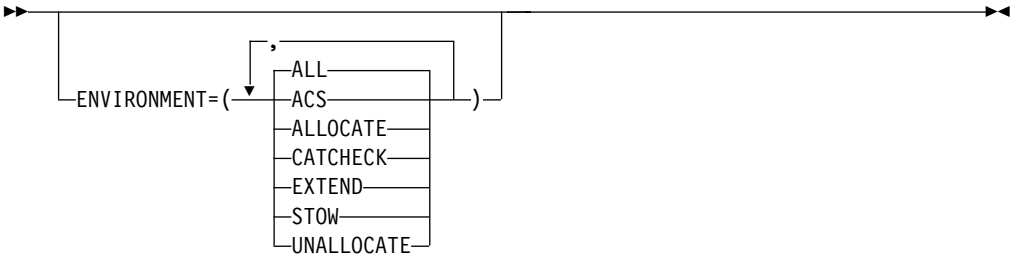
```
DIAG=DISABLE
```

Limit the operation of a rule definition to specific environments (ENVIRONMENT)

Within Advanced Allocation Management, the term *environment* describes a particular phase in the processing of a data set during which Advanced Allocation Management can perform function and selection processing.

The ENVIRONMENT keyword limits the operation of a rule definition to one or more specific environments. You can specify the ENVIRONMENT keyword on DEFAULTS, INCLUDE, or EXCLUDE statements.

Syntax



ENVIRONMENT=(ACS, ALL, ALLOCATE, CATCHECK, EXTEND, STOW, UNALLOCATE)

Operands

ACS
The point in SMS processing directly following the assignment of one or more of the SMS constructs of Data Class, Storage Class, Management Class, or Storage Group.

ALL
All points where Advanced Allocation Management intercepts processing. Specifying ALL is the same as specifying all of the previous six options (ACS, ALLOCATE, CATCHECK, EXTEND, STOW, and UNALLOCATE).

ALLOCATE
The point in processing where a new data set is allocated (including VSAM and non-VSAM, SMS and non-SMS data sets).

CATCHECK
The point in allocation processing where the determination of the prior

existence of a data set is made. For SMS data sets, this point is just before the allocation of a data set. For non-SMS data sets, this point can be just before the allocation of a data set (or at unallocation processing time during catalog disposition processing).

EXTEND
The point in processing where a new extent is added to a data set on the current volume or on a new volume (including volume addition).

STOW
The point in PDS processing when a new member is being added to a data set and the directory is being updated.

UNALLOCATE
The point in processing where a data set is unallocated.

Note: You can specify one or more of the specified options.

Usage and processing considerations

While not required, the ENVIRONMENT keyword can be helpful when you are specifying and converting to rule definitions.

Default

ALL

Environments

ACS, ALLOCATE, CATCHECK, EXTEND, STOW, and UNALLOCATE.

Example

In the following example, the specified environment is EXTEND:

```
ENVIRONMENT=(EXT)
```

Display messages (MESSAGES)

The MESSAGES keyword specifies whether to display messages.

Syntax

MESSAGES=ENABLE | DISABLE

Operands

ENABLE
Display messages.

DISABLE
Do not display messages.

Default

DISABLE

Environments

ACS, ALLOCATE, CATCHECK, EXTEND, STOW, and UNALLOCATE.

Related keywords

MSG_DESTINATION and MSG_LEVEL.

Example

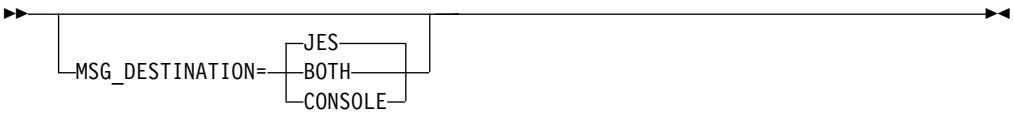
In the following example, MESSAGES is set to DISABLE; therefore, no messages are displayed:

```
MESSAGES=DISABLE
```

MSG_DESTINATION

The MSG_DESTINATION keyword specifies where to direct messages (the JES SYSMSG data set, the operator console, or both).

Syntax



MSG_DESTINATION=BOTH | CONSOLE | JES

Operands

BOTH
JES SYSMSG data set and operator console.

CONSOLE
Operator console.

JES
JES SYSMSG data set.

Usage and processing considerations

The MSG_DESTINATION keyword is associated with the MESSAGES keyword. You must specify `MESSAGES=ENABLE` to implement this function.

Default

JES

Environments

ACS, ALLOCATE, CATCHECK, EXTEND, STOW, and UNALLOCATE.

Related keywords

MESSAGES and MSG_LEVEL.

Example

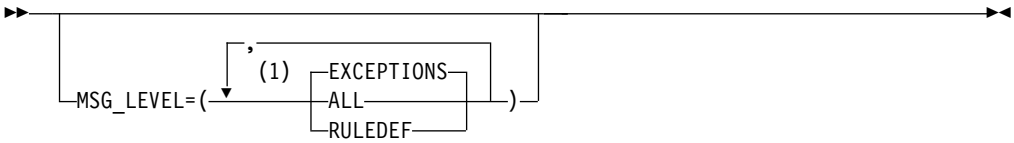
In this example, MSG_DESTINATION is set to CONSOLE; therefore, messages are directed to the operator console:

```
MSG_DESTINATION=CONSOLE
```

MSG_LEVEL

The MSG_LEVEL keyword specifies whether to display only exception messages, display messages that concern a rule definition match, or display all messages.

Syntax



Notes:

1 Although you can specify both ALL and EXCEPTIONS, the two options contradict one another. If you specify ALL and EXCEPTIONS, only the EXCEPTIONS option is honored.

MSG_LEVEL=(EXCEPTIONS, ALL, RULEDEF)

Operands

ALL
Display exception messages and informational messages that are related to successful completion of certain functions by Advanced Allocation Management.

Note: MSG_LEVEL=ALL does not include the RULEDEF option. When only MSGLEVEL=ALL is specified, rule definition match messages are not displayed.

EXCEPTIONS
Display only exception messages.

RULEDEF
When you specify MSG_LEVEL=RULEDEF, Advanced Allocation Management displays a message that concerns a rule definition match each time that it checks for a matching rule definition and finds a match. Advanced Allocation Management does not display a message if it does not find a rule definition match.

Note: Checks for a matching rule definition can occur many times during the processing of a data set, and therefore can increase the number of messages that are output to the message destination.

Default

EXCEPTIONS

Usage and processing considerations

The MSG_LEVEL keyword is associated with the MESSAGES keyword. You must specify MESSAGES=ENABLE to implement this function.

Environments

ACS, ALLOCATE, CATCHECK, EXTEND, STOW, and UNALLOCATE.

Related keywords

MESSAGES and MSG_DESTINATION.

Example: MSG_LEVEL=RULEDEF

In the following example, MSG_LEVEL is set to RULEDEF:

```
INCLUDE JOBNAME=MSGLEVEL
        STEPNAME=CREATE0
        MSG_LEVEL=RULEDEF
```

The resulting job log shows the line number of the matching rule definition statements:

```
GLO2188I MATCHING INCLUDE RULEDEF LINE NUMBER = 17 - Q31V
GLO2188I MATCHING INCLUDE RULEDEF LINE NUMBER = 17 - Q31V
IEF236I ALLOC. FOR MSGLEVEL CRESTEP CREATE0
IGD100I 2105 ALLOCATED TO DDNAME DD1 DATACLAS ( )
GLO2188I MATCHING INCLUDE RULEDEF LINE NUMBER = 17 - Q31V
```

Next, MSG_LEVEL is set to EXCEPTIONS when an attempt is made to allocate a data set with the duplicate data set name (STEP CREATE2) and use the NOTCAT function to resolve the situation:

```
INCLUDE JOBNAME=MSGLEVEL
        STEPNAME=CREATE2
        MSG_LEVEL=EXCEPTIONS
        NOTCAT2=RE
        NOTCAT2_C=A
        NOTCAT2_Q=NSMS
        NOTCAT2_QUALIFIER_P=1
```

No RULEDEF messages are present in the job log.

Finally, when MSG_LEVEL=ALL is specified:

```
INCLUDE JOBNAME=MSGLEVEL
        STEPNAME=CREATE4
        MSG_LEVEL=ALL
        NOTCAT2=RE
        NOTCAT2_C=A
        NOTCAT2_Q=NSMS
        NOTCAT2_QUALIFIER_P=1
```

The resulting job log shows the following messages:

```
GLO2087I DATA SET ON VOLSER NSL200 RENAMED AT ALLOCATION TIME -Q31V
GLO2086I NEW NAME: NSMS.NSMS.RSTEST.GLO.ORL12.MSGALL -Q31V
GLO2123I MSGLEVEL,CREATE4,CRESTEP,DD1,NSMS.RSTEST.GLO.ORL12.MSGALL -Q31V
```

Note: When MSG_LEVEL=ALL is specified, no MSG_LEVEL messages are displayed in the job log (as they are when MSG_LEVEL=RULEDEF is specified).

Example: MSG_LEVEL=ALL

In the following example, MSG_LEVEL is set to ALL. Therefore, exception messages and informational messages that are related to successful completion of certain functions by Advanced Allocation Management, are displayed:

```
MSG_LEVEL=ALL
```

Example: MSG_LEVEL=(ALL, RULEDEF)

In the following example, MSG_LEVEL is set to ALL,RULEDEF:

```
INCLUDE JOBNAME=AAMTEST1
    MSG_LEVEL=(ALL,RULEDEF)
    SPACE_RELEASE=ENABLE
```

The resulting job log shows the following messages:

```
GLO2188I MATCHING INCLUDE RULEDEF LINE NUMBER = 17 - Q31V
GLO2188I MATCHING INCLUDE RULEDEF LINE NUMBER = 17 - Q31V
IEF236I ALLOC. FOR AAMTEST1 CRESTEP CREATE0
IGD100I 2105 ALLOCATED TO DDNAME SYSUT2 DATACLAS ( )
GLO2188I MATCHING INCLUDE RULEDEF LINE NUMBER = 17 – Q31V
GLO2125I SPACE RELEASE RECOVERED 9 TRACKS -Q31V
GLO2123I AAMTEST1,CREATE0,CRESTEP,SYSUT2,RSQA.AAMTEST1.GLO.TEST.DATA –Q31V
```

Process NOT CATLGD 2 errors (NOTCAT2)

The NOTCAT2 function detects and handles NOT CATLGD 2 errors. A NOT CATLGD 2 error is reported by the system message IEF287I at step termination when an attempt is made to catalog a newly created data set, but it is determined that the data set name is already cataloged to other volumes.

If the system default is set such that these errors do not cause job termination, allowing subsequent steps to continue might result in the usage of an incorrect data set. Usage of an incorrect data set might cause processing errors that can cause significant losses in employee and processing time spent recovering from the ensuing problems.

The NOTCAT2 keyword specifies how to handle NOT CATLGD 2 errors as reported by the system message IEF287I. Advanced Allocation Management handles NOT CATLGD 2 errors only when the data set in question is cataloged.

Syntax

NOTCAT2=DISABLE | JCLFAIL | RENAME | SCRATCH | UNCATALOG

Operands

Specify one of the following operands:

DISABLE
Do not perform the NOTCAT2 function.

JCLFAIL
Process NOT CATLGD 2 errors and set the job stream to fail with the following system message upon detecting the IEF287I NOT CATLGD 2 message for a data set:

```
IEF378I jobname jjjjjjjj - JOB FAILED - CATALOG DISPOSITION ERROR
```

Note: JCLFAIL does not apply to SMS-managed data sets because SMS performs the equivalent checks and fails a job when it encounters a duplicate data set name at allocation time.

RENAME
Rename the currently cataloged data set that has the duplicate data set name so that the newly created data set can be successfully cataloged.

SCRATCH
Uncatalog and delete the currently cataloged data set that has the duplicate data set name so that the newly created data set can be successfully cataloged.

Note: SCRATCH is valid for SMS and non-SMS-managed data sets with the NOTCAT2_CHECK=ALLOCATION option.

UNCATALOG
Uncatalog the currently cataloged data set that has the duplicate data set name so that the newly created data set can be successfully cataloged.

Notes:

1. UNCATALOG is ignored for SMS-managed data sets because SMS-managed data sets cannot be validly uncataloged.
2. UNCATALOG is valid with NOTCAT2_CHECK=ALLOCATION and NOTCAT2_CHECK=STEPTERM.

Notes:

1. This option is valid for SMS and non-SMS-managed data sets with the NOTCAT2_CHECK=ALLOCATION options.
2. If you use the NOTCAT2_QUALIFIER option, make sure that the high-level qualifier chosen maintains the status of the data set as SMS or non-SMS-managed. NOTCAT2_QUALIFIER or NOTCAT2_SUFFIX (or both) must be specified for this option to be effective.
3. If either, or both, of the NOTCAT2_QUALIFIER or NOTCAT2_SUFFIX options cause the length of the data set name to exceed 44 characters in length, the data set name is truncated. The NOTCAT2_QUALIFIER value is applied first (if present) followed by the NOTCAT2_SUFFIX (if present).
4. NOTCAT2 does not process dynamically allocated data sets. To process dynamically allocated data sets, use the NOTCAT2_DYNAL function. With the NOTCAT2_DYNAL function, you can selectively enable, or disable, processing of dynamically allocated data sets.

Usage and processing considerations

- NOTCAT2 is only applicable to newly created non-VSAM data sets.
- NOTCAT2 does not process dynamically allocated data sets. To process dynamically allocated data sets, use the NOTCAT2_DYNAL function. With the NOTCAT2_DYNAL function, you can selectively enable, or disable, processing of dynamically allocated data sets.

- When NOTCAT2 causes failure of the remainder of a job, system message IEF378I is issued.
- To allow the uncataloging, deletion, and renaming of data sets if the NOTCAT2 functions of UNCATALOG, SCRATCH, and RENAME are used, the Advanced Allocation Management started task must be authorized using your security authorization product.
- The NOTCAT2_CHECK STEPTERM option is ignored for SMS-managed data sets because SMS detects duplicate data set names during allocation and issues the IGD17101I message upon detection of the duplicate data set name, which then causes the current and subsequent steps to be flushed.
- Advanced Allocation Management handles NOT CATLGD 2 errors only when the data set in question is cataloged.

As shown in Table 9, all values of NOTCAT2 except DISABLE cause Advanced Allocation Management to intervene in NOTCAT2 situations for non-SMS-managed, non-VSAM data sets. NOTCAT2 processing does not apply to VSAM data sets.

For SMS-managed data sets, SMS causes the equivalent of `NOTCAT2=JCLFAIL` to occur without the assistance of Advanced Allocation Management (unless NOTCAT2_CHECK=ALLOCATE while NOTCAT2 specifies SCRATCH or RENAME).

Table 9. Effect of Advanced Allocation Management NOTCAT2 values on SMS- and non-SMS-managed data sets

NOTCAT2 value	NOTCAT2_CHECK (ALLOCATE)	NOTCAT2_CHECK (STEPTERM)
DISABLE	neither	neither
JCLFAIL	non-SMS*	non-SMS*
UNCATALOG	non-SMS	non-SMS
SCRATCH	both	non-SMS
RENAME	both	non-SMS

Note: * SMS causes the equivalent of `NOTCAT2=JCLFAIL` processing during allocation for SMS-managed data sets, without the assistance of Advanced Allocation Management.

Default

DISABLE

Environments

CATCHECK

Related keywords

NOTCAT2_CHECK, NOTCAT2_DYNAL, NOTCAT2_QUALIFIER, NOTCAT2_QUALIFIER_POSITION, and NOTCAT2_SUFFIX.

The following keywords are related to the NOTCAT2 function:

- The NOTCAT2_CHECK keyword specifies when the detection of the duplicate data set name occurs (during the initial allocation of the data set before the step

is performed, or at step termination time during data set disposition processing and after detection of the IEF287I NOT CATLGD 2 message for the data set).

- Use the NOTCAT2_QUALIFIER, NOTCAT2_QUALIFIER_POSITION, and NOTCAT2_SUFFIX keywords with the NOTCAT2 RENAME option to indicate that the currently cataloged data set with the duplicate data set name should be renamed so that the newly created data set can be successfully cataloged.

Example: NOTCAT2

The following shows an example of the NOTCAT2 function:

```
* RULE DEFINITIONS

DEFAULTS

*
INCLUDE JOBNAME=ABCDJOB * Item to include in processing
        DDNAME=DD1      * Item to include in processing
        NOTCAT2=JCLFAIL * Function to invoke
```

The INCLUDE keyword specifies the following selection criteria:

- JOBNAME=ABCDJOB specifies that jobs that have a job name of ABCDJOB are to be included in Advanced Allocation Management processing.
- DDNAME=DD1 specifies that jobs with a DDNAME of DD1 are to be included in Advanced Allocation Management processing.

NOTCAT2=JCLFAIL specifies that Advanced Allocation Management should fail the remainder of the job when it encounters an IEF287I NOT CATLGD 2 reported error.

Therefore, if Advanced Allocation Management encounters a job with a job name of ABCDJOB that has a DDNAME of DD1 that received a NOT CATLGD 2 error, it causes the remainder of the job to fail as reported by system message IEF378I.

A job that has a job name that does not match the job name selection criteria of ABCDJOB with the DDNAME selection criteria of DD1 does not process NOT CATLGD 2 errors in this manner.

Example: NOTCAT2 specified with NOTCAT2_CHECK

In the following example, NOTCAT2 is specified with the NOTCAT2_CHECK keyword:

```
* RULE DEFINITIONS

DEFAULTS NOTCAT2=JCLFAIL          * Default settings
         NOTCAT2_CHECK=STEPTERM
*
INCLUDE SMS_MANAGED=YES           * Item to include in processing
        NOTCAT2=SCRATCH           * Exception to default processing
```

NOTCAT2=JCLFAIL specifies that a job that encounters a NOT CATLGD 2 error is to fail at the step where the error is encountered and that subsequent steps should not be processed.

NOTCAT2_CHECK=STEPTERM specifies that the NOTCAT2 processing occurs at the end of the job step.

The INCLUDE keyword specifies selection criteria and exceptions to default processing. In the example, `SMS_MANAGED=YES` specifies that SMS-managed data sets are to be included in Advanced Allocation Management processing. `NOTCAT2=SCRATCH` specifies that the existing data set is to be deleted.

Therefore, if Advanced Allocation Management encounters a job whose attempt to allocate a new SMS-managed data set would fail because a data set with the same name is cataloged on another volume, Advanced Allocation Management deletes the old data set to allow the new data set to be allocated and cataloged.

NOTCAT2_CHECK

The NOTCAT2_CHECK keyword specifies when the detection of the duplicate data set name occurs.

Syntax

NOTCAT2_CHECK=ALLOCATION | STEPTERM

Operands

Specify one of the following operands:

ALLOCATION
The detection of the duplicate data set name occurs during the initial allocation of the data set before the step is performed.

Note: The ALLOCATION option is available for SMS and non-SMS-managed data sets.

STEPTERM
The detection of the duplicate data set name occurs at step termination (during data set disposition processing, and after detection of the IEF287I NOT CATLGD 2 message for the data set).

Notes:

1. The STEPTERM option is ignored for SMS-managed data sets because SMS detects duplicate data set names during allocation. SMS issues the IGD17101I message upon detection of the duplicate data set name, which causes the current and subsequent steps to be flushed.
2. If you use NOTCAT2 at step termination time, IEF253I and IGD17001I duplicate data set name messages are visible if the new data set is allocated to the same volser as the old data set. This situation occurs because you cannot have two data sets with the same name in a single VTOC. Therefore, the step never reaches the step termination phase because it fails during allocation. (This situation should occur only if you use an explicit volser for a non SMS-managed data set. If you allow the system choose a volser, it attempts to choose a volume that does not already have the data set on it.)

Default

STEPTERM

Environments

CATCHECK

Related keywords

NOTCAT2, NOTCAT2_DYNAL, NOTCAT2_QUALIFIER, NOTCAT2_QUALIFIER_POSITION, and NOTCAT2_SUFFIX.

Example

In the following example, the detection of the duplicate data set name occurs during the initial allocation of the data set before the step is performed:

```
NOTCAT2_CHECK=ALLOCATION
```

NOTCAT2_DYNAL

In some cases, NOTCAT2 failures for dynamic allocations are intercepted and processed by the calling program (for example, TSO). However, some applications do not automatically handle such situations and for which NOTCAT2 processing might be beneficial. In these cases, you can use the NOTCAT2_DYNAL function to selectively enable, or disable, processing of dynamically allocated data sets.

Syntax

NOTCAT2_DYNAL=ENABLE | DISABLE

Operands

ENABLE
Allow NOTCAT2 processing for dynamically allocated data sets.

DISABLE
Bypass NOTCAT2 processing for dynamically allocated data sets.

Usage and processing considerations

Advanced Allocation Management does not automatically handle NOTCAT2 for dynamically allocated data sets. The default for the NOTCAT2_DYNAL keyword is DISABLE. Without knowledge of the specific actions that are performed by a calling program, unexpected problems might occur.

Default

DISABLE

Environments

CATCHECK

Related keywords

NOTCAT2, NOTCAT2_CHECK, NOTCAT2_QUALIFIER, NOTCAT2_QUALIFIER_POSITION, and NOTCAT2_SUFFIX.

Example

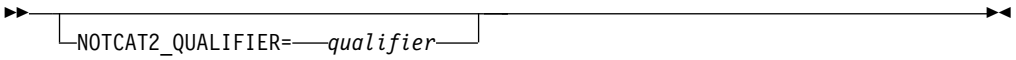
In the following example, dynamically allocated data sets are eligible for NOTCAT2 processing:

```
NOTCAT2_DYNAL=ENABLE
```

NOTCAT2_QUALIFIER

The NOTCAT2_QUALIFIER keyword specifies a high-level qualifier.

Syntax



NOTCAT2_QUALIFIER=*qualifier*

Where:

qualifier
is a qualifier (up to eight characters in length) to add to the existing data set name in the position that is specified by the NOTCAT2_QUALIFIER_POSITION keyword. The value that you specify must meet normal data set naming conventions for valid characters. The following rules apply for a data set qualifier:

- The first character must be an alphabetic or national ($, #, @) character.
- Characters 2 - 8 must be an alphanumeric or national character.

If needed, a period is automatically added to the end of the value when the new data set name is formed. If the addition of the value to the data set name causes it to exceed 44 characters in length, the data set name is truncated at 44 characters.

If you specify NOTCAT2_QUALIFIER and NOTCAT2_SUFFIX, NOTCAT2_QUALIFIER is applied first.

Note: For the NOTCAT2_QUALIFIER keyword to be effective, the NOTCAT2_QUALIFIER_POSITION keyword must have a nonzero value.

Default

None.

Environments

CATCHECK

Related keywords

NOTCAT2, NOTCAT2_CHECK, NOTCAT2_DYNAL, NOTCAT2_QUALIFIER_POSITION, and NOTCAT2_SUFFIX.

Example

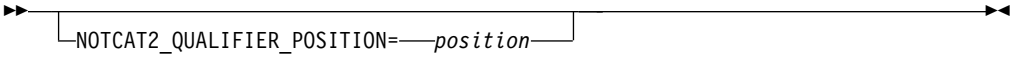
In the following example, the resulting data set name has a qualifier of abc as the second qualifier in the data set name. For example: tst.abc.data:

```
NOTCAT2_QUALIFIER=abc
NOTCAT2_QUALIFIER_POSITION=2
```

NOTCAT2_QUALIFIER_POSITION

The NOTCAT2_QUALIFIER_POSITION keyword specifies the position of the data set name qualifier that is defined by the NOTCAT2_QUALIFIER keyword.

Syntax



NOTCAT2_QUALIFIER_POSITION=*position*

Where:

position
is a numeric value that specifies the position of the data set name qualifier. The maximum value is 22. A value of 1 indicates that the qualifier will be the highest level qualifier in the data set name. A value that exceeds the current number of qualifiers in the data set name results in the qualifier being added to the end of the data set name.

Default

The default is 0, which is the same as not specifying NOTCAT2_QUALIFIER_POSITION.

Environments

CATCHECK

Related keywords

NOTCAT2, NOTCAT2_CHECK, NOTCAT2_DYNAL, NOTCAT2_QUALIFIER, and NOTCAT2_SUFFIX.

Example

In the following example, the resulting data set name begins with abc. For example, abc.test.data:

```
NOTCAT2_QUALIFIER=abc
NOTCAT2_QUALIFIER_POSITION=1
```

NOTCAT2_SUFFIX

The NOTCAT2_SUFFIX keyword specifies a low-level qualifier.

Syntax

NOTCAT2_SUFFIX=*suffix*

Where:

suffix
is a low-level qualifier (up to eight characters in length) to add to the end of the existing data set name. The value must meet normal data set naming conventions for valid characters. If the addition of the value to the end of the data set name causes it to exceed 44 characters in length, the data set name is truncated at 44 characters. If NOTCAT2_QUALIFIER and NOTCAT2_SUFFIX are specified, NOTCAT2_SUFFIX is applied last.

Default

None.

Environments

CATCHECK

Related keywords

NOTCAT2, NOTCAT2_CHECK, NOTCAT2_DYNAL, NOTCAT2_QUALIFIER, and NOTCAT2_QUALIFIER_POSITION.

Example

In the following example, the resulting data set name ends with myjcl. For example, `abc.test.myjcl`:

```
NOTCAT2_SUFFIX=myjcl
```

Dynamic PDS directory increase (PDS_DIR_INC)

The PDS_DIR_INC function detects and recovers DIRECTORY FULL conditions by dynamically increasing the directory space within a partitioned data set when a DIRECTORY FULL condition occurs.

DIRECTORY FULL conditions occur when an attempt is made to store directory information for a new member in a PDS using the MVS STOW service and all of the directory blocks are full. Currently, when an attempt is made to store directory information for a new member in a partitioned data set and no space is available in the directory, the STOW service is terminated. If the STOW service was invoked from CLOSE processing, the system issues a B14-0C abend. When this situation occurs, the only way to add the new member is to perform the following steps:

1. Copy the current member data to a temporary location.
2. Allocate a new data set with more directory blocks.
3. Copy the old data set to the new data set.
4. Copy the current member data from the temporary location to the new data set.
5. Delete the old data set.
6. Rename the new data set using the original data set name.

This process can be time-consuming and prone to error.

With the PDS_DIR_INC function, the DIRECTORY FULL condition can be detected and dynamically recovered. When Advanced Allocation Management detects a

DIRECTORY FULL condition, it determines whether it is possible to add another block to the directory. If so, the function adds a single directory block to the directory to provide more space to allow the current STOW operation to complete successfully. It also allows for the future addition of 6 to 20 members to the data set, depending on the type of information that is being stored in the directory entries (such as ISPF statistics).

The PDS_DIR_INC keyword specifies whether to automatically increase the directory space within a partitioned data set when a DIRECTORY FULL condition occurs.

Syntax

PDS_DIR_INC=ENABLE | DISABLE

Operands

ENABLE
Automatically increase the directory space.

DISABLE
Do not increase the directory space.

Default

DISABLE

Usage and processing considerations

MVS requires that all directory blocks in a PDS directory be contiguous. It also requires that the entire directory reside in the first extent of the data set. When the PDS_DIR_INC function detects a DIRECTORY FULL condition, it verifies that a directory block can be added without violating these MVS requirements. Therefore, if the current directory completely occupies the first extent of the data set, it is not possible for Advanced Allocation Management to add a directory block. If member data follows the last directory block on the track, Advanced Allocation Management attempts to move necessary members to free the remaining space on the track for the new directory block. If not enough space is available in the data set to move the required members, Advanced Allocation Management cannot complete the recovery.

Notes:

1. To use the PDS_DIR_INC function, the S1_INTERCEPT subsystem option must be set to ENABLE (the default setting).
2. PDS/Es do not have the same restrictions regarding the fixed size of the directory and therefore do not encounter DIRECTORY FULL conditions. Thus, the PDS_DIR_INC function does not attempt to operate on PDS/Es.
3. Advanced Allocation Management obtains control during the MVS STOW service to determine when it is necessary to perform recovery processing for DIRECTORY FULL conditions. Therefore, processes that

rewrite the PDS directory without using STOW processing are not eligible for dynamic PDS directory increase processing.

Depending upon the type of operation that is being performed and the type of data sets that are being processed, IEBCOPY might use STOW (for example, when one or more members of a PDS or PDS/E are being copied to another, standard PDS data set). If the input data set is a standard PDS, IEBCOPY does not use STOW to update the output PDS. If the input data set is a PDS/E, IEBCOPY uses STOW to update the output PDS.

Advanced Allocation Management attempts to perform dynamic PDS directory increase for IEBCOPY only in cases where it uses STOW to add member information to the directory of a PDS.

Environments

STOW

Example

In the following example, the PDS_DIR_INC function applies and is processed for all jobs except for the jobs where the job name is ABCJOB1:

```
* RULE DEFINITIONS
DEFAULTS PDS_DIR_INC=ENABLE       * Default settings
*
INCLUDE JOBNAME=ABCJOB1           * item to include in processing
PDS_DIR_INC=DISABLE
```

Note the following items in the example:

- `PDS_DIR_INC=ENABLE` indicates the PDS_DIR_INC function is enabled.
- The INCLUDE keyword specifies the selection criteria to which the function applies. `JOBNAME=ABCJOB1` specifies that jobs that specify a job name of ABCJOB1 are to be included in Advanced Allocation Management processing.
- `PDS_DIR_INC=DISABLED` indicates that the PDS_DIR_INC function should be disabled.

Therefore, if Advanced Allocation Management encounters a job where the job name matches ABCJOB1 and a DIRECTORY FULL condition occurs, Advanced Allocation Management does *not* increase the directory space as defined by the subsequent PDS_DIR_INC statement.

If Advanced Allocation Management encounters a job where the job name does *not* match ABCJOB1 and a DIRECTORY FULL condition occurs, Advanced Allocation Management increases the directory space as defined by the PDS_DIR_INC keyword in the DEFAULTS statement.

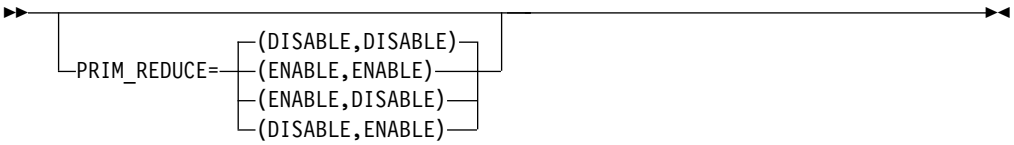
Controlled reduction of primary space (PRIM_REDUCE)

The PRIM_REDUCE function helps avoid JOB FAILED-JCL errors that occur due to insufficient space for a primary allocation.

Out-of-space conditions can occur for primary allocations when a job requests allocation for a new data set and the primary space is not available on the requested volume. When an out-of-space condition occurs for a primary allocation, the PRIM_REDUCE function reduces the original allocation by a user-specified amount, up to a maximum value, until the allocation is successful.

The PRIM_REDUCE keyword specifies whether to reduce the primary allocation when an out-of-space condition occurs.

Syntax



PRIM_REDUCE=(ENABLE | DISABLE, ENABLE | DISABLE)

Operands

The first operand applies to the primary allocation on the first volume. The following values are valid:

ENABLE
Reduce the primary allocation when an out-of-space condition occurs.

DISABLE
Do not reduce the primary allocation when an out-of-space condition occurs.

The second operand applies to the primary allocation on the second and subsequent volumes. The following values are valid:

ENABLE
Reduce the primary allocation when an out-of-space condition occurs.

DISABLE
Do not reduce the primary allocation when an out-of-space condition occurs.

Default

DISABLE, DISABLE

Environments

ALLOCATE and EXTEND.

Usage and processing considerations

- You must set PRIM_REDUCE_INC to a nonzero value to implement the function.
- For the PRIM_REDUCE function initial allocation (the first set of PRIM_REDUCE operands), Advanced Allocation Management uses the amount specified by PRIM_REDUCE_INC to continue reducing the allocation until it is successful or the limit that is specified by PRIM_REDUCE_LIM is reached.

  For PRIM_REDUCE non-initial allocations (the second set of PRIM_REDUCE operands), Advanced Allocation Management evaluates the amount of available freespace and reduces the allocation to that value or the PRIM_REDUCE_LIM value, whichever is greater.
- The PRIM_REDUCE, PRIM_REDUCE_INC, and PRIM_REDUCE_LIM keywords are supported for VSAM and non-VSAM data sets. The PRIM_REDUCE function and related keywords work in the same manner for both types of data sets.

- For non-VSAM data sets, the first allocation on volumes other than the first volume, is not technically a primary allocation. The allocation quantity value that is used to extend non-VSAM volumes to additional volumes is the secondary quantity.
- For SMS VSAM and non-VSAM data sets, because the volume serial number is not yet chosen by the system at PRIM_REDUCE time, if VOLSER_CURRENT is specified in the selection criteria with a value other than $NULL, the rule definition statement is considered not to match for the PRIM_REDUCE function on the first volume.
- When specifying explicit volsers for a data set and one of the non-first volumes contains no free space when an extend to a new volume occurs, end of volume processing normally produces an E37 abend. If you want Advanced Allocation Management to skip the volume with no free space and continue processing by attempting the next explicitly specified volume or by adding a new volume using the VOL_ADD function, you must specify `PRIM_REDUCE=(,ENABLE)`.
- If you specify more than 65535 tracks of data for a primary allocation for non-VSAM, non-LARGE format, non-extended format data sets, Advanced Allocation Management reduces the specified value to 65535 tracks or less, as part of PRIM_REDUCE processing.
- PRIM_REDUCE is not performed for DFDSS dynamically allocated data sets.
- If a data set has the Guaranteed Space attribute defined in the SMS Storage Class, PRIM_REDUCE processing does not occur for the data set.
- If DSORG or RECORG is not explicitly specified in the JCL or if RECORG is not specified in the associated Data Class definition, the data set organization is not known at PRIM_REDUCE time. Therefore, if DSORG is specified in the selection criteria for a rule, the rule definition statement is considered not to match.
- The smaller the value that you specify on the PRIM_REDUCE_LIM keyword, the more successful the PRIM_REDUCE function will be in reducing the allocation to a value that prevents the failure of the allocation. The smaller the value that you specify for PRIM_REDUCE_INC, the closer the allocation quantity will be to the minimum size that is needed to make the allocation succeed, which therefore reduces the unused space for the allocation.

  Note: Specifying a smaller value for PRIM_REDUCE_INC can increase the number of allocation attempts.

Related keywords

PRIM_REDUCE_INC and PRIM_REDUCE_LIM.

Example: PRIM_REDUCE (ENABLE, DISABLE)

In the following example, when an out-of-space condition occurs, the primary allocation is reduced on the first volume, but not on the second, or subsequent volumes:

```
PRIM_REDUCE=(ENABLE, DISABLE)
```

Figure 9 on page 88 illustrates the example:

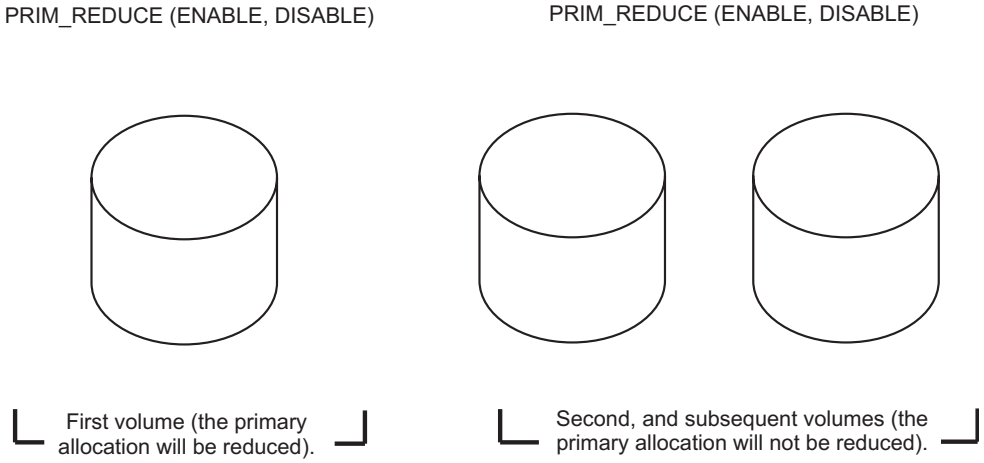


Figure 9. PRIM_REDUCE (ENABLE, DISABLE)

Example: PRIM_REDUCE=ENABLE

In the following example, `PRIM_REDUCE=ENABLE` specifies that the PRIM_REDUCE function is enabled:

```
* RULE DEFINITIONS
DEFAULTS PRIM_REDUCE=ENABLE        * Default settings
         PRIM_REDUCE_LIM=50
         PRIM_REDUCE_INC=10
*
INCLUDE  JOBNAME=ABCJOB2           * Item to include in processing
INCLUDE  JOBNAME=ABCJOB3           * Item to include in processing
         PRIM_REDUCE_INC=05        * Function to invoke
```

Note the following items in the example:

- `PRIM_REDUCE_LIM=50` specifies that the primary allocation amount should not be reduced below 50 percent of the original allocation amount.
- `PRIM_REDUCE_INC=10` specifies that the original allocation value should be reduced by 10 percent of the current allocation amount (on each allocation attempt) until the allocation is successful.

The INCLUDE keywords that follow the PRIM_REDUCE keywords specify the selection criteria:

- The first INCLUDE keyword specifies that jobs that have a job name of ABCJOB2 are to be included in Advanced Allocation Management processing.
- The second INCLUDE keyword specifies that jobs that have a job name of ABCJOB3 are to be included in Advanced Allocation Management processing. The PRIM_REDUCE_INC keyword that follows specifies that for these jobs, the original allocation is to be reduced by 5 percent of the current allocation amount instead of 10 percent.

Therefore, if Advanced Allocation Management encounters a job that has a job name of ABCJOB2 and an out-of-space condition occurs for a primary allocation, Advanced Allocation Management reduces the original allocation by 10 percent of the current allocation amount (not to exceed 50 percent of the original allocation amount) until the allocation is successful.

If Advanced Allocation Management encounters a job that has a job name of ABCJOB3 and an out-of-space condition occurs for a primary allocation, Advanced

Allocation Management reduces the original allocation by 5 percent of the current allocation amount (not to exceed 50 percent of the original allocation amount) until the allocation is successful.

Jobs that do not have job names of ABCJOB2 or ABCJOB3 are included in Advanced Allocation Management processing. When an out-of-space condition occurs for these jobs, the jobs abend.

PRIM_REDUCE_INC

The PRIM_REDUCE_INC keyword specifies the increment to use (a percentage value of the current allocation amount) when reducing the primary allocation as enabled by the PRIM_REDUCE keyword.

Syntax

PRIM_REDUCE_INC=*xx*,yy

Where:

xx is the increment (0 - 99) to apply to the primary allocation on the first volume.

yy is the increment (0 - 99) to apply to the primary allocation on the second and subsequent volumes.

Note: Specifying 0 disables PRIM_REDUCE.

Usage and processing considerations

When an out-of-space condition occurs for a primary allocation, the original allocation is reduced by the amount that is specified by PRIM_REDUCE_INC, up to the value specified for the PRIM_REDUCE_LIM keyword, until the allocation is successful.

For example, if enabled by the PRIM_REDUCE keyword, when an out-of-space condition occurs for a primary allocation on the first volume, Advanced Allocation Management reduces the original allocation by the amount that is specified by PRIM_REDUCE_INC until the allocation is successful or the limit that is specified by the PRIM_REDUCE_LIM keyword is reached.

For example, if `PRIM_REDUCE_INC=5` and `PRIM_REDUCE_LIM=80` Advanced Allocation Management reduces the allocation amount in 5 percent increments, not to exceed 80 percent of the original allocation amount.

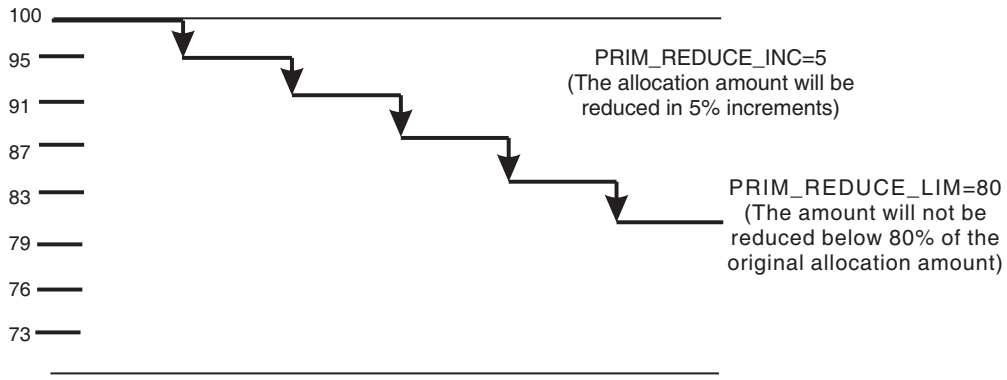


Figure 10. PRIM_REDUCE_INC (primary allocation on the first volume)

Note: The PRIM_REDUCE_INC keyword is associated with the PRIM_REDUCE and PRIM_REDUCE_LIM keywords. You must specify `PRIM_REDUCE=ENABLE` to implement this function and PRIM_REDUCE_LIM to specify the reduction limit.

Default

The default is 0,0.

Environments

ALLOCATE and EXTEND.

Related keywords

PRIM_REDUCE and PRIM_REDUCE_LIM.

Example

In the following example, when an out-of-space condition occurs for a primary allocation on the *first volume*, the original allocation is reduced by 40 percent of the current allocation amount, up to the value that is specified for the PRIM_REDUCE_LIM keyword, until the allocation is successful:

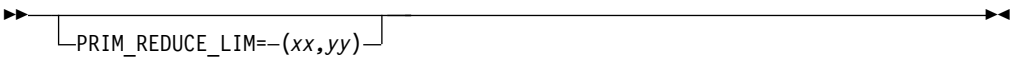
```
PRIM_REDUCE_INC=(40,40)
```

When an out-of-space condition occurs for a primary allocation on the *second and subsequent volumes*, the original allocation is also reduced by 40 percent of the current allocation amount, up to the value that is specified for the PRIM_REDUCE_LIM keyword, until the allocation is successful.

PRIM_REDUCE_LIM

The PRIM_REDUCE_LIM keyword specifies the reduction limit as a percentage value of the original allocation amount.

Syntax



PRIM_REDUCE_LIM=(*xx*,*yy*)

Where:

xx is the limit (0 - 99) to apply to the primary allocation on the first volume.

yy is the limit (0 - 99) to apply to the primary allocation on the second and subsequent volumes.

Usage and processing considerations

- The PRIM_REDUCE_LIM keyword is associated with the PRIM_REDUCE keyword. PRIM_REDUCE_LIM is effective only when the corresponding PRIM_REDUCE operand is set to ENABLE. When an out-of-space condition occurs, the size of the primary allocation is not reduced below the amount that is specified by the PRIM_REDUCE_LIM keyword.
- When the PRIM_REDUCE_LIM value for the initial volume of a non-VSAM data set is set to zero, the primary quantity for a data set is set to zero (if necessary) by the PRIM_REDUCE function. This allows the secondary quantity, if specified, to be used to allocate space on any explicitly specified volumes or on volumes that are added by Advanced Allocation Management through the VOL_ADD function. If the PRIM_REDUCE_LIM value is nonzero, the lowest value to which the primary quantity is reduced is 1. For VSAM data sets, the lowest possible primary quantity value is one allocation unit, regardless of the setting of PRIM_REDUCE_LIM.

Default

The default is 0,0.

Environments

ALLOCATE and EXTEND.

Related keywords

PRIM_REDUCE_INC and PRIM_REDUCE.

Example

In the following example, when an out-of-space condition occurs for a primary allocation on the *first volume*, Advanced Allocation Management reduces the original allocation by the amount that is specified by PRIM_REDUCE_INC until the allocation is successful, or the limit that is specified by the PRIM_REDUCE_LIM keyword (20 percent of the original allocation amount) is reached:

```
PRIM_REDUCE_LIM=(20,40)
```

When an out-of-space condition occurs for a primary allocation on the *second and subsequent volumes*, Advanced Allocation Management evaluates the amount of freespace and reduce the allocation to that value, or to the value specified by PRIM_REDUCE_LIM (40 percent of the original allocation amount), whichever is greater.

Test a rule definition (RULEDEF_TEST)

Use the RULEDEF_TEST function to verify that selection criteria specified on a rule definition INCLUDE or EXCLUDE statement are set up properly to match intended data set allocations.

The RULEDEF_TEST function generates information that indicates whether Advanced Allocation Management considers selection criteria that is specified on a rule definition INCLUDE or EXCLUDE statement to match a data set that is being allocated or extended.

Syntax

RULEDEF_TEST=ENABLE | DISABLE

Operands

ENABLE
Enable testing for rule definitions.

DISABLE
Disable testing for rule definitions (perform processing).

Usage and processing considerations

The RULEDEF_TEST keyword specifies whether entities that match a specific rule definition are to be tested or processed.

When RULEDEF_TEST is set to ENABLE on an INCLUDE or EXCLUDE statement and a data set allocation or extend is performed that matches the selection criteria on that INCLUDE or EXCLUDE statement, no preventive or recovery action is taken. Instead, a message is written to indicate that the match occurred. If SMF recording for Advanced Allocation Management is active, an SMF record that indicates RULEDEF_TEST processing for the data set is also generated. This behavior allows you to test one or more rule definition INCLUDE or EXCLUDE statements before you use them in processing.

Default

DISABLE

Environments

ACS, ALLOCATE, CATCHECK, EXTEND, STOW, and UNALLOCATE.

Example

In the following example, when jobs that have a job name of ABC pass though allocation or end-of-volume processing, test messages are generated that indicate that Advanced Allocation Management recognizes them as matches for the rule definition:

```
RULE DEFINITIONS

DEFAULTS
*
INCLUDE JOBNAME=ABC
  PRIM_REDUCE=ENABLE
  SEC_REDUCE=ENABLE
  RULEDEF_TEST=ENABLE
```

After you run the ABC job and observe the test messages, you can remove RULEDEF_TEST (or set it to DISABLE) and be confident that the rule definition will perform the indicated processing (PRIM_REDUCE and SEC_REDUCE) for jobs named ABC.

Add a secondary allocation (SEC_ALLOC)

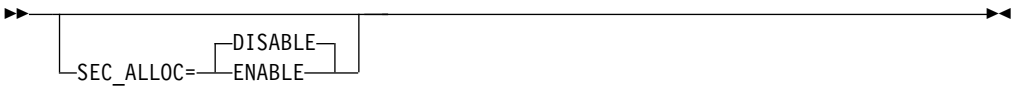
The SEC_ALLOC function helps avoid job failures that are caused by D37 abends.

Data sets that are created with only a primary allocation are not extended when the primary allocation is full and therefore, a D37 abend occurs. During D37 processing, the SEC_ALLOC function obtains control and adds a secondary space value that is based on a percentage of the original primary allocation amount (provided no secondary allocation was specified).

The SEC_ALLOC keyword specifies whether to add a secondary allocation during D37 processing when one does not exist in order to prevent an out-of-space condition.

Note: The SEC_ALLOC function will not work with PDSEs. Advanced Allocation Management secondary allocation functions are not compatible with PDSEs.

Syntax



SEC_ALLOC=ENABLE | DISABLE

Operands

ENABLE
Add a secondary allocation.

DISABLE
Do not add a secondary allocation.

Usage and processing considerations

The SEC_ALLOC keyword is associated with the SEC_ALLOC_AMT keyword. You must set SEC_ALLOC_AMT to a nonzero value to implement the SEC_ALLOC function.

The SEC_ALLOC and SEC_ALLOC_AMT keywords are supported for VSAM and non-VSAM data sets. The SEC_ALLOC function and related keyword work in the same manner for both types of data sets.

The following is the processing order of the SEC_* functions:

1. SEC_ALLOC
2. SEC_INC
3. SEC_BEST
4. SEC_REDUCE

Default

DISABLE

Environments

EXTEND

Related keywords

SEC_ALLOC_AMT

Example

In the following example, the PRIM_REDUCE function is specified within the default settings:

```
* RULE DEFINITIONS
DEFAULTS PRIM_REDUCE=ENABLE        * Default settings
         PRIM_REDUCE_LIM=50
         PRIM_REDUCE_INC=10
*
INCLUDE  JOBNAME=ABCDJOB           * Item to include in processing
         STEPNAME=CREATE           * Item to include in processing
         SEC_ALLOC=ENABLE          * Function to invoke
         SEC_ALLOC_AMT=50
```

Note the following items:

The INCLUDE keyword that follows the PRIM_REDUCE function keywords specify the selection criteria:

- `JOBNAME=ABCDJOB` specifies that only those jobs that have a job name of ABCDJOB are to be included in Advanced Allocation Management processing.
- `STEPNAME=CREATE` specifies that only those jobs that have a step name of CREATE are to be included in Advanced Allocation Management processing.

The SEC_ALLOC function keywords that follow the INCLUDE keyword and selection criteria, specify that Advanced Allocation Management should add a secondary allocation if an out-of-space condition occurs:

- `SEC_ALLOC=ENABLE` specifies that the SEC_ALLOC function is enabled.
- `SEC_ALLOC_AMT= 50` specifies that the secondary allocation amount is 50 percent of the original primary allocation amount.

Therefore, if Advanced Allocation Management encounters an out-of-space condition for a job that has a job name of ABCDJOB and a stepname of CREATE, it adds a secondary allocation. Because SEC_ALLOC_AMT is set to 50, the secondary allocation amount is 50 percent of the original primary allocation amount. If Advanced Allocation Management encounters a job that does not match all of the specified selection criteria and an out-of-space condition occurs, the job is allowed to abend.

SEC_ALLOC_AMT

The SEC_ALLOC_AMT keyword specifies the secondary allocation amount as a percentage of the original primary allocation amount.

Syntax

SEC_ALLOC_AMT=*xxxx*

Where:

xxxx
is an amount that is specified as a percentage of the original primary allocation amount. The valid range is 0 - 1000. Specifying 0 disables SEC_ALLOC.

Default

The default is 0.

Environments

EXTEND

Related keywords

SEC_ALLOC

Note: The SEC_ALLOC_AMT keyword is associated with the SEC_ALLOC keyword. You must specify SEC_ALLOC=ENABLE to implement this function.

Example

In the following example, SEC_ALLOC_AMT is set to 10; therefore, the secondary allocation amount is 10 percent of the original primary allocation amount:

```
SEC_ALLOC_AMT=10
```

Reduce secondary allocation to largest available free extent (SEC_BEST)

The SEC_BEST function helps to increase the chances for successful job completion when the operating system fails to acquire the secondary allocation amount that is needed to successfully complete a job.

When the operating system fails to acquire the secondary allocation amount that is needed to successfully complete a job, a B37 abend can result. The SEC_BEST function occurs before the attempt to allocate space to prevent the abend from occurring. It decreases the secondary allocation to fit in the largest single extent on the volume and then re-drives the space acquisition processing.

The SEC_BEST keyword specifies whether to reduce the secondary allocation amount to the largest extent available on a volume when not enough space exists to allocate the original secondary allocation amount.

Note: The SEC_BEST function will not work with PDSEs. Advanced Allocation Management secondary allocation functions are not compatible with PDSEs.

Syntax

SEC_BEST=ENABLE | DISABLE

Operands

ENABLE
Reduce the secondary extent.

DISABLE
Do not reduce the secondary extent.

Default

DISABLE

Environments

EXTEND

Usage and processing considerations

The purpose of the SEC_BEST function is to prevent volume fragmentation by reducing (if needed) a secondary allocation quantity to the size of the largest free extent. If the SEC_BEST function is not enabled, an extent that is too large to fit into the largest single extent on the volume is split into a maximum of five extents, which can cause fragmentation on the volume. To adjust the size of the extents and the amount of distribution of extents on the volume, use SEC_REDUCE in place of, or with the SEC_BEST function.

The SEC_BEST and SEC_BEST_LIM keywords are supported for VSAM and non-VSAM data sets. The SEC_BEST function and related keyword work in the same manner for both types of data sets.

The following is the processing order of the SEC_* functions:

1. SEC_ALLOC
2. SEC_INC
3. SEC_BEST
4. SEC_REDUCE

The SEC_BEST function always occurs before a secondary allocation attempt and SEC_REDUCE occurs only when the secondary allocation attempt fails.

Related keywords

SEC_BEST_LIM

Example

In the following example, the PRIM_REDUCE function is specified within the default settings:

```
* RULE DEFINITIONS
DEFAULTS PRIM_REDUCE=ENABLE        * Default settings
         PRIM_REDUCE_LIM=50
         PRIM_REDUCE_INC=10
*
INCLUDE  JOBNAME=ABCDJOB           * Item to include in processing
         SEC_BEST=ENABLE           * Function to invoke
```

Note the following items:

The INCLUDE keyword that follows the PRIM_REDUCE function keywords specify the selection criteria. JOBNAME=ABCDJOB specifies that only jobs that have a job name of ABCDJOB are to be included in Advanced Allocation Management processing.

The SEC_BEST function keyword that follows the INCLUDE keyword and selection criteria, specify that Advanced Allocation Management is to reduce the secondary allocation to fit in the largest single extent on the volume.

SEC_BEST_LIM

The SEC_BEST_LIM keyword specifies a limit as a percentage of the original allocation amount, below which the SEC_BEST function does not reduce the original allocation.

Syntax

SEC_BEST_LIM=*xx*

Where:

xx is a limit that is specified as a percentage of the original allocation amount below which the SEC_BEST function does not reduce the original allocation. The valid range is 0 - 99.

Note: The SEC_BEST_LIM keyword is associated with the SEC_BEST keyword. You must specify SEC_BEST=ENABLE to implement this function.

Default

The default is 0.

Environments

EXTEND

Related keywords

SEC_BEST

Example

In the following example, SEC_BEST_LIM is set to 50; therefore, Advanced Allocation Management evaluates the amount of freespace and reduces the allocation to that value, or to the value that is specified by SEC_BEST_LIM (50 percent of the original allocation amount), whichever is greater:

```
SEC_BEST_LIM=50
```

Increase the size of secondary allocation (SEC_INC)

The SEC_INC function helps avoid abends that can occur due to insufficient secondary allocations. The function helps prevent x37 abends for data sets that unexpectedly increase in size or that do so at infrequent intervals, by increasing the size of each successive secondary allocation at a specified point.

The SEC_INC keyword specifies whether to increase secondary allocations after a specific extent number is reached as defined by the SEC_INC_XTNT keyword.

Note: The SEC_INC function will not work with PDSEs. Advanced Allocation Management secondary allocation functions are not compatible with PDSEs.

Syntax

SEC_INC=ENABLE | DISABLE

Operands

ENABLE
Increase secondary allocations.

DISABLE
Do not increase secondary allocations.

Default

DISABLE

Environments

EXTEND

Usage and processing considerations

- The SEC_INC keyword is associated with the SEC_INC_AMT keyword. You must set SEC_INC_AMT to a nonzero value to implement this function.
- If the newly increased size does not fit in a single extent, you can reduce the size by also specifying the SEC_BEST function, or recover from it failing by using the SEC_REDUCE function.
- The SEC_INC function supports only non-VSAM data sets.

The following is the processing order of the SEC_* functions:

1. SEC_ALLOC
2. SEC_INC

3. SEC_BEST
4. SEC_REDUCE

Related keywords

SEC_INC_AMT, SEC_INC_VOLNO, and SEC_INC_XTNT.

Example: SEC_INC=DISABLE

In the following example, SEC_INC is set to DISABLE; therefore, Advanced Allocation Management does not increase secondary allocations:

```
SEC_INC=DISABLE
```

Example: SEC_INC=ENABLE

In the following example, SEC_INC=ENABLE enables the SEC_INC function:

```
* RULE DEFINITIONS
DEFAULTS  SEC_INC=ENABLE          * Default settings
          SEC_INC_AMT=100
          SEC_INC_XTNT=1
*
INCLUDE  JOBNAME=ABCDJOB          * Item to include in processing
```

Note the following items:

- SEC_INC_AMT=100 specifies, as a percentage of the previous allocation amount, the amount by which secondary allocations are to be increased. Because SEC_INC_AMT is set to 100, Advanced Allocation Management increases secondary allocations by 100 percent of the previous allocation amount.
- SEC_INC_XTNT=1 specifies the extent number at which secondary allocations are to be increased as defined by the SEC_INC and SEC_INC_AMT keywords.

The INCLUDE keyword that follows the SEC_INC keywords specify the selection criteria. JOBNAME=ABCDJOB specifies that only jobs that have a job name of ABCDJOB are to be included in Advanced Allocation Management processing.

Therefore, if Advanced Allocation Management encounters a situation where the specified extent (extent 1) is reached for a job that has a job name of ABCDJOB, it increases the secondary allocations by 100 percent of the previous allocation amount, at extent 1.

A job that has a job name that does not match the job name selection criteria of ABCDJOB is allowed to abend if there is an insufficient secondary allocation.

SEC_INC_AMT

The SEC_INC_AMT keyword specifies, as a percentage of the original or previous allocation amount, the amount by which to increase secondary allocations.

Syntax

SEC_INC_AMT=*xxxx*

Where:

xxxx
is a percentage of the original or previous allocation amount, the amount by which to increase secondary allocations. The valid range is 0 - 1000. Specifying 0 disables SEC_INC.

Note: The SEC_INC_AMT keyword is associated with the SEC_INC keyword. You must specify `SEC_INC=ENABLE` to implement this function.

Default

The default is 0.

Environments

EXTEND

Related keywords

SEC_INC, SEC_INC_VOLNO, and SEC_INC_XTNT.

Example

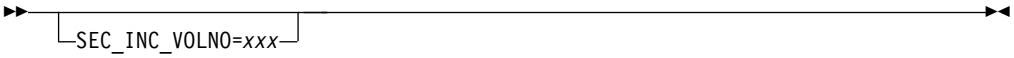
In the following example, SEC_INC_AMT is set to 20; therefore, Advanced Allocation Management increases secondary allocations by 20 percent of the original, or previous, allocation amount:

```
SEC_INC_AMT=20
```

SEC_INC_VOLNO

The SEC_INC_VOLNO keyword specifies the volume sequence number at which to increase secondary allocations as defined by the SEC_INC, SEC_INC_AMT, and SEC_INC_XTNT keywords.

Syntax



SEC_INC_VOLNO=*xxx*

Where:

xxx
is the volume sequence number at which secondary allocations are increased as defined by the SEC_INC, SEC_INC_AMT, and SEC_INC_XTNT keywords. The valid range is 0 - 255. If you specify 0, the SEC_INC_XTNT value is assumed to apply to the first volume of a data set.

Usage and processing considerations

The SEC_INC_VOLNO keyword is associated with the SEC_INC keyword. You must specify `SEC_INC=ENABLE` to implement this function.

Default

The default is 0.

Environments

EXTEND

Related keywords

SEC_INC_AMT, SEC_INC, and SEC_INC_XTNT.

Example

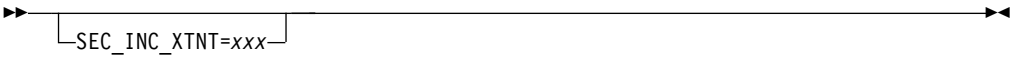
In the following example, SEC_INC_VOLNO is set to 5; therefore, at volume sequence number 5, Advanced Allocation Management increases secondary allocations as defined by the SEC_INC, SEC_INC_AMT and SEC_INC_XTNT keywords:

```
SEC_INC_VOLNO=5
```

SEC_INC_XTNT

The SEC_INC_XTNT keyword specifies the extent number on the volume that is specified by the SEC_INC_VOLNO keyword.

Syntax



SEC_INC_XTNT=*xxx*

Where:

xxx
: is the extent number on the volume that is specified by the SEC_INC_VOLNO keyword. The valid range is 0 - 255.

  If SEC_INC_VOLNO is not specified, *xxx* is the first volume that secondary allocations are to be increased as defined by the SEC_INC and SEC_INC_AMT keywords.

Usage and processing considerations

The SEC_INC_XTNT keyword is associated with the SEC_INC keyword. You must specify `SEC_INC=ENABLE` to implement this function.

Default

The default is 0.

Environments

EXTEND

Related keywords

SEC_INC, SEC_INC_AMT, and SEC_INC_VOLNO.

Example

In the following example, SEC_INC_XTNT is set to 9; therefore, at extent 9, Advanced Allocation Management increases secondary allocations as defined by the SEC_INC and SEC_INC_AMT keywords:

```
SEC_INC_XTNT=9
```

Recover from secondary allocation space failures (SEC_REDUCE)

The SEC_REDUCE function recovers from secondary allocation space failures. The function reduces the allocation amount to a value that is consistent with the current free space on the volume.

The SEC_REDUCE keyword specifies whether to reduce a secondary allocation on the current volume in response to an out-of-space condition.

Note: The SEC_REDUCE function will not work with PDSEs. Advanced Allocation Management secondary allocation functions are not compatible with PDSEs.

Syntax

SEC_REDUCE=ENABLE | DISABLE

Operands

ENABLE
Reduce the secondary allocation.

DISABLE
Do not reduce the secondary allocation.

Usage and processing considerations

- The SEC_REDUCE keyword is associated with the SEC_REDUCE_INC keyword. You must set SEC_REDUCE_INC to a nonzero value to implement this function.
- When you use SEC_REDUCE with multi-striped SMS data sets, the secondary allocation amount is reduced by a user-specified percentage that is specified through the SEC_REDUCE_INC keyword until the allocation is successful.
- The SEC_REDUCE_INC keyword applies to SMS multi-striped data sets only. Therefore, the 5 percent that is specified by the SEC_REDUCE_INC keyword in "Example: SEC_REDUCE=ENABLE" on page 103 applies to SMS data sets only. For non-SMS data sets, the allocation is reduced to the largest available extent that is not below the SEC_REDUCE_LIM value.
- The SEC_REDUCE, SEC_REDUCE_INC, and SEC_REDUCE_LIM keywords are supported for VSAM and non-VSAM data sets. The SEC_REDUCE function and related keywords work in the same manner for both types of data sets. For VSAM data sets, Advanced Allocation Management does not reduce the secondary space below the size of a control area (CA).

The following is the processing order of the SEC_* functions:

1. SEC_ALLOC

2. SEC_INC
3. SEC_BEST
4. SEC_REDUCE

The SEC_BEST function always occurs before a secondary allocation attempt and SEC_REDUCE occurs only when the secondary allocation attempt fails.

Default

DISABLE

Environments

EXTEND

Related keywords

SEC_REDUCE_INC and SEC_REDUCE_LIM.

Example: SEC_REDUCE=DISABLE

In the following example, SEC_REDUCE is set to DISABLE; therefore, Advanced Allocation Management does not reduce the secondary allocation on the current volume in response to an out-of-space condition:

```
SEC_REDUCE=DISABLE
```

Example: SEC_REDUCE=ENABLE

In the following example, SEC_REDUCE=ENABLE specifies that the SEC_REDUCE function is to be enabled:

```
* RULE DEFINITIONS
DEFAULTS SEC_REDUCE=ENABLE        * Default settings
         SEC_REDUCE_LIM=20
         SEC_REDUCE_INC=5
*
INCLUDE  JOBNAME=ABCJOB*          * Item to include in processing
         STEPNAME=CREATE          * Item to include in processing
         DDNAME=DD1               * Item to include in processing
```

Note the following items:

- SEC_REDUCE_INC= 5 specifies that Advanced Allocation Management is to reduce a secondary allocation by 5 percent of the original secondary allocation amount.
- SEC_REDUCE_LIM=20 specifies the limit (20 percent of the original secondary allocation amount) that Advanced Allocation Management should not exceed during the reduction process.

The INCLUDE keyword specifies the selection criteria:

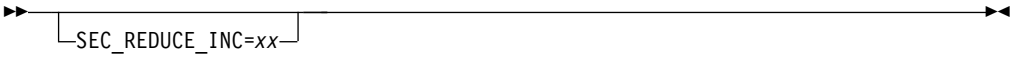
- JOBNAME=ABCJOB* specifies that only jobs that have a job name that begins with ABCJOB are to be included in Advanced Allocation Management processing.
- STEPNAME=CREATE specifies that only jobs that have a step name of CREATE are to be included in Advanced Allocation Management processing.
- DDNAME=DD1 specifies that only jobs that have a ddname of DD1 are to be included in Advanced Allocation Management processing.

Therefore, if Advanced Allocation Management encounters an out-of-space condition for a data set where the job has a job name that begins with ABCJOB (for example, ABCJOB5) with a STEPNAME of CREATE and a DDNAME of DD1, it reduces the secondary allocation by 5 percent of the original secondary allocation amount, not to exceed 20 percent of the original secondary allocation amount. If Advanced Allocation Management encounters a job that does not match all of the specified selection criteria and an out-of-space condition occurs, the job is allowed to abend.

SEC_REDUCE_INC

The SEC_REDUCE_INC keyword specifies a percentage of the original secondary allocation amount by which a secondary allocation should be reduced with the SEC_REDUCE keyword.

Syntax



SEC_REDUCE_INC=*xx*

Where:

xx is a percentage of the original secondary allocation amount by which a secondary allocation should be reduced with the SEC_REDUCE keyword. The valid range is 0 - 99. Specifying 0 disables SEC_REDUCE.

Usage and processing considerations

- The SEC_REDUCE_INC keyword is associated with the SEC_REDUCE keyword. You must specify `SEC_REDUCE=ENABLE` to implement this function. Secondary reduction is limited by the SEC_REDUCE_LIM keyword.
- The SEC_REDUCE_INC value is only used for the following data sets:
  1. SMS multi-striped data sets, both VSAM and non-VSAM, when a secondary space reduction is performed.
  2. SMS non-VSAM data sets when the secondary allocation quantity is used during an extend to a new volume.

Default

The default is 0.

Environments

EXTEND

Related keywords

SEC_REDUCE and SEC_REDUCE_LIM.

Example

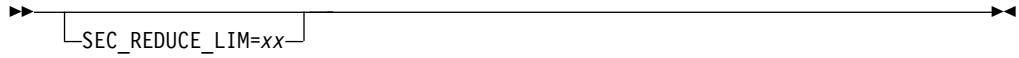
In the following example, SEC_REDUCE_INC is set to 20; therefore, for multi-striped data sets defined by SMS, Advanced Allocation Management reduces a secondary allocation by 20 percent of the original secondary allocation amount:

```
SEC_REDUCE_INC=20
```

SEC_REDUCE_LIM

The SEC_REDUCE_LIM keyword specifies the secondary allocation reduction limit as a percentage of the original secondary allocation amount below which the secondary allocation should not be reduced.

Syntax



SEC_REDUCE_LIM=*xx*

Where:

xx is the secondary allocation reduction limit as a percentage of the original secondary allocation amount below which the secondary allocation should not be reduced. The valid range is 0 - 99.

Usage and processing considerations

The SEC_REDUCE_LIM keyword is associated with the SEC_REDUCE keyword. You must specify `SEC_REDUCE=ENABLE` to implement this function.

Default

The default is 0.

Environments

EXTEND

Related keywords

SEC_REDUCE and SEC_REDUCE_INC.

Example

In the following example, SEC_REDUCE_LIM is set to 10; therefore, Advanced Allocation Management evaluates the amount of freespace and reduces the allocation to that value, or to the value specified by SEC_REDUCE_LIM (10 percent of the original allocation amount), whichever is greater:

```
SEC_REDUCE_LIM=10
```

Recover unused disk space (SPACE_RELEASE)

The SPACE_RELEASE function increases available space for other tasks in the system by automatically releasing unused space when a data set is closed.

The SPACE_RELEASE keyword specifies whether to release unused space for a data set during unallocation and, if so, whether to release it only after a secondary extent was allocated.

Syntax

SPACE_RELEASE=ENABLE | DISABLE | SECONDARY

Operands

ENABLE
Release unused space.

DISABLE
Do not release unused space.

SECONDARY
Release unused space only if a secondary extent was allocated. (Space is released if a secondary extent exists.)

Default

DISABLE

Environments

UNALLOCATE

Usage and processing considerations

- SPACE_RELEASE is only applicable to new or existing physical sequential data sets.
- The SPACE_RELEASE function is independent of the RLSE subparameter. It does not matter if you specify RLSE in the JCL. Advanced Allocation Management performs the SPACE_RELEASE function, if specified to do so within the rule definitions.
- Because Advanced Allocation Management causes space release to occur, OS/390® or z/OS restrictions apply.

Example: SPACE_RELEASE=DISABLE

In the following example, SPACE_RELEASE is set to DISABLE; therefore, Advanced Allocation Management does not release space:

```
SPACE_RELEASE=DISABLE
```

Example: SPACE_RELEASE=ENABLE

In the following example, SPACE_RELEASE=ENABLE specifies that the SPACE_RELEASE function is enabled:

```
 * RULE DEFINITIONS
 DEFAULTS
 *
INCLUDE  JOBNAME=ABCDJOB            * Item to include in processing
         DDNAME=DD1                 * Item to include in processing
         SPACE_RELEASE=ENABLE       * Function to invoke
```

Note the following items:

The INCLUDE keyword specifies the following selection criteria:

- `JOBNAME=ABCDJOB` specifies that jobs that have a job name of ABCDJOB are to be included in Advanced Allocation Management processing.
- `DDNAME=DD1` specifies that jobs with a DDNAME of DD1 are to be included in Advanced Allocation Management processing.

`SPACE_RELEASE=ENABLE` specifies that Advanced Allocation Management should automatically release unused space when a data set is closed.

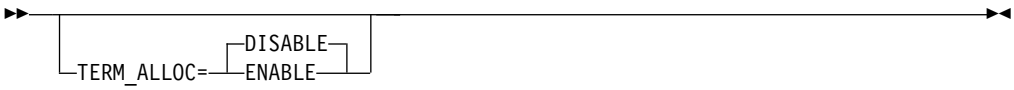
Therefore, if Advanced Allocation Management encounters a job that has a job name of ABCDJOB and a DDNAME of DD1, it automatically releases unused space when a data set is closed.

A job that has a job name that does not match the job name selection criteria of ABCDJOB does not have space that is released in this manner.

Terminate an allocation (TERM_ALLOC)

The TERM_ALLOC keyword specifies that an allocation should be forced to fail when the selection criteria in a rule definition are met. For example, you can use the TERM_ALLOC function to enforce site installation standards by forcing an allocation to fail when the selection criteria in a rule definition are met.

Syntax



TERM_ALLOC=ENABLE | DISABLE

Operands

ENABLE
Force the current allocation to fail. For more information, see "Usage and processing considerations."

DISABLE
Process the current allocation normally.

Usage and processing considerations

When set to ENABLE, Advanced Allocation Management fails the allocation by simulating an authorization failure, which varies based on the SMS management status of the data set, and whether the data set is a VSAM data set:

- For a non-SMS, non-VSAM data set, the allocation fails with the message:
  `IEF720I JJHTESTZ ALLOCATE ddname - USER NOT AUTHORIZED TO DEFINE THIS DATA SET`

  For a batch job, this results in a JCL error and the failure of the job.
- For a non-SMS, VSAM data set, the DEFINE fails with the following messages:

  `IDC3036I DATA SET COULD NOT BE DEFINED TO SECURITY SUBSYSTEM`

  `IDC3007I ** VSAM CATALOG RETURN CODE IS 118`
- For an SMS, non-VSAM data set, the allocation fails with the message:

```
IEF344I JJHTESTZ ALLOCATE ddname - ALLOCATION FAILED DUE TO DATA FACILITY
SYSTEM ERROR
```

- For an SMS, VSAM data set, the allocation fails with the message:

```
IDC3932I STORAGE MANAGEMENT SUBSYSTEM CALL FAILED. REASON CODE WAS 17061
```

In all cases, regardless of the setting of the MSG_LEVEL option, Advanced Allocation Management displays the following message when an allocation is failed:

```
GLO2999I ALLOCATION TERMINATED AS REQUESTED BY RULEDEF LINE nnnnn - ssid
```

Where *nnnnn* is the rule definition line number that is associated with the TERM_ALLOC function and *ssid* is the Advanced Allocation Management subsystem identification. The message is followed by message GLO2123I, which identifies the data set. If SMF recording is enabled, an SMF record is created to document the failure.

Default

DISABLE

Environments

ACS, ALLOCATE, CATCHECK, STOW, and UNALLOCATE.

Example

In the following example, if the job name is PRODJOB and the SMS Data Class is specified as TEST, the allocation fails:

```
DEFAULTS
*
INCLUDE JOBNAME=PRODJOB
           DATACLAS=TEST
           TERM_ALLOC=ENABLE
```

Dynamically add volumes (VOL_ADD)

The VOL_ADD function helps to avoid out-of-space conditions that occur during end-of-volume processing. During end-of-volume processing, it dynamically adds another volume when the secondary allocation space request cannot be satisfied on the current volume, and properly handles requisite catalog updates.

The VOL_ADD keyword specifies whether to add additional volumes when an out-of-space condition occurs and no more volumes are available as specified in the original allocation.

Syntax

VOL_ADD=ENABLE |DISABLE

Operands

ENABLE
Add additional volumes.

DISABLE
Do not add additional volumes.

To locate a suitable volume, Advanced Allocation Management searches the volume groups that are specified using the VOL_ADD_UNIT, VOL_ADD_VSER, VOL_ADD_VGRP, and VOL_ADD_SGRP keywords. Advanced Allocation Management searches all of these volume groups until it finds the best candidate.

VOL_ADD searches for a default volume group in the following circumstances:

- when you explicitly specify a volume group, or
- when you specify VOL_ADD and you do not specify values for the VOL_ADD_UNIT, VOL_ADD_VSER, and VOL_ADD_VGRP keywords.

VOL_ADD will find a default volume group that is based on the first, or current, volser.

Figure 11 on page 110 shows the processing logic Advanced Allocation Management uses to determine the best candidate:

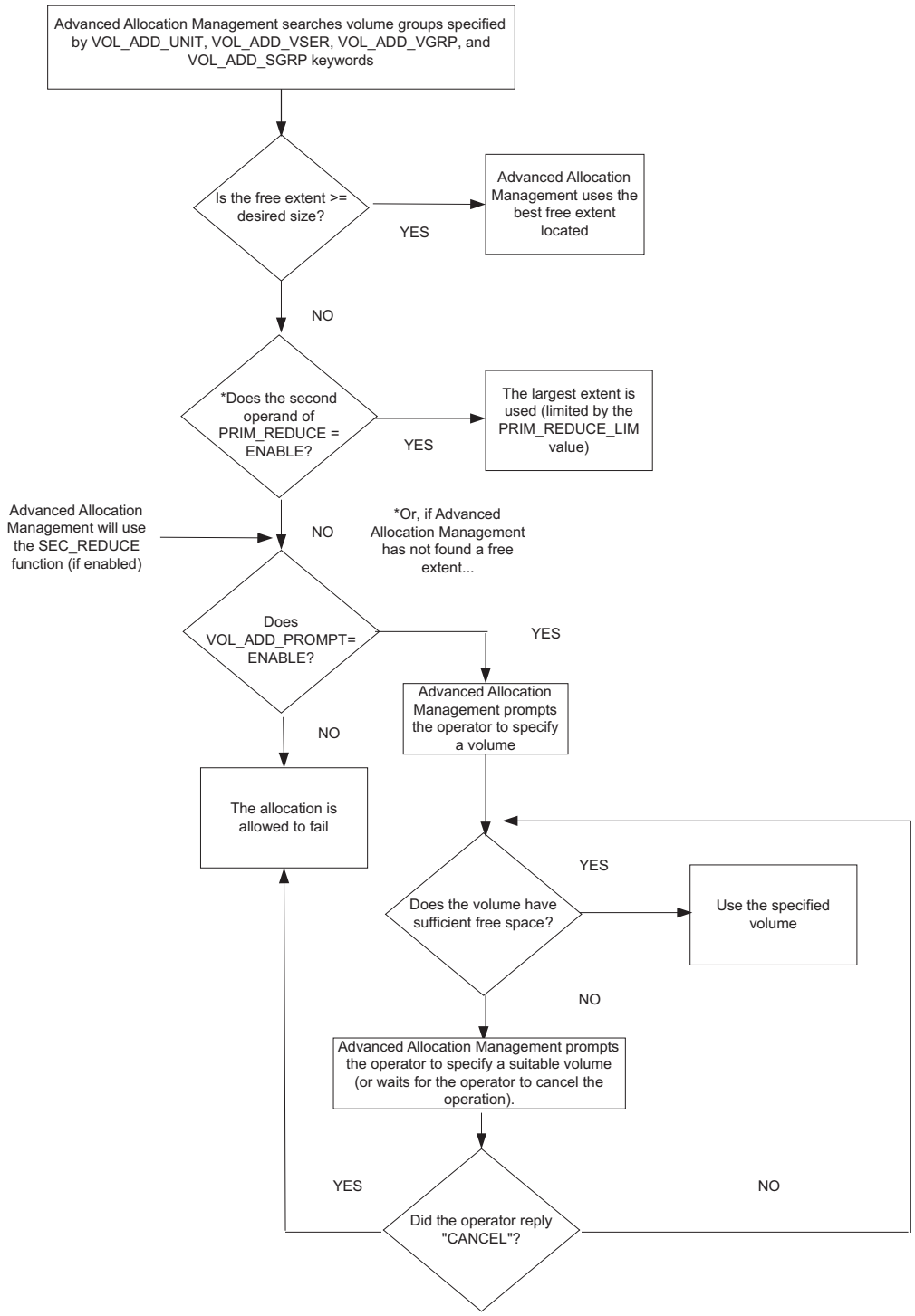


Figure 11. VOL_ADD keyword and associated processing logic

Default

DISABLE

Environments

EXTEND

Usage and processing considerations

Data set support:

- The VOL_ADD, VOL_ADD_LIM, VOL_ADD_PROMPT, VOL_ADD_SGRP, VOL_ADD_UNIT, VOL_ADD_VGRP, and VOL_ADD_VSER keywords are supported for VSAM and non-VSAM data sets. The VOL_ADD function and related keywords work in the same manner for both types of data sets.
- MVS data management does not support multiple volumes for the following data sets, and therefore the VOL_ADD function is not supported for these data sets:
  - PDSes (PDS and PDS/E)
  - Sequential (non-VSAM) multi-striped data sets
  - VSAM temporary data sets
  - VIO data sets
- The VOL_ADD function does not support data sets with a DSORG value of DA.
- The VOL_ADD function cannot be specified for SORTWK data sets. You cannot extend SORTWK data sets to another volume. SORTWK data sets can be on a single volume only. If you need more space, use the SEC_INC function or add more SORTWK data sets.

SMS:

- For SMS data sets (VSAM and non-VSAM), SMS selects additional volumes based on the SMS constructs that apply for the data set (Advanced Allocation Management has no influence on the volumes that are used). Therefore, volume groups are not used and the VOL_ADD_LIM, VOL_ADD_PROMPT, VOL_ADD_SGRP, VOL_ADD_UNIT, VOL_ADD_VGRP, VOL_ADD_VSER, VOL_ADD_ALGORITHM, VOL_ADD_MOUNT_USAGE, and VOL_ADD_VGRP_SEARCH keywords do not apply.
- When SMS extends a VSAM data set that has in its associated data class the ADD'L VOLUME AMOUNT attribute set to (S) Secondary, the initial allocation on the new volume is set to the value of the secondary allocation quantity. If the secondary allocation quantity is zero, the extend does not occur even if the VOL_ADD function is enabled and (or) candidate volumes exist. For such an extend to be successful, enable the Advanced Allocation Management SEC_ALLOC function or set the SMS ADD'L VOLUME AMOUNT attribute to (P) Primary.

Processing:

- Because of integrity considerations, Advanced Allocation Management does not perform a VOL_ADD if more than one DCB or ACB is opened against a DDNAME or if more than one DDNAME is assigned to the same data set name.
- When you specify explicit volsers for a data set and one of the non-first volumes contains no free space when an extend to a new volume occurs, end of volume processing normally produces an E37 abend. If you want Advanced Allocation Management to skip the volume with no free space and continue processing by trying the next explicitly specified volume or by adding a new volume using the VOL_ADD function, you must specify `PRIM_REDUCE=(,ENABLE)`.

- When Advanced Allocation Management processes a VSAM cluster, all alternate indexes within the upgrade set for the cluster are also handled when end of volume is reached for the AIX®.

Related keywords

VOL_ADD_ALGORITHM, VOL_ADD_LIM, VOL_ADD_MOUNT_USAGE, VOL_ADD_PROMPT, VOL_ADD_SGRP, VOL_ADD_UNIT, VOL_ADD_VGRP, VOL_ADD_VSER, VOL_ADD_UPG_VOL, and VOL_ADD_VGRP_SEARCH.

Example

In the following example, `VOL_ADD=ENABLE` specifies that the VOL_ADD function is enabled:

```
* RULE DEFINITIONS
*
EXCLUDE JOBNAME=ABCJOB1 * Item to exclude from processing
INCLUDE JOBNAME=ABCJOB* * Item to include in processing
         VOL_ADD=ENABLE * Default settings
         VOL_ADD_LIM=10
         VOL_ADD_VSER=(DASD01,DASD02,
         DASD03,DASD04)
```

Note the following items:

- `VOL_ADD_LIM =10` specifies that Advanced Allocation Management is to add a maximum of 10 volumes, if needed.
- The VOL_ADD_VSER keyword lists the volume serial numbers Advanced Allocation Management should use to determine where it should obtain the additional volumes.

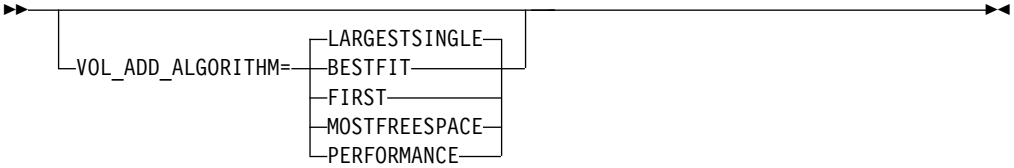
The EXCLUDE keyword specifies the selection criteria that are to be *excluded* from Advanced Allocation Management processing. In the example, jobs that have a job name of ABCJOB1 are excluded from Advanced Allocation Management processing. If Advanced Allocation Management encounters a job that has a job name of ABCJOB1 and a secondary allocation space request cannot be satisfied on the current volume, it does not add additional volumes.

However, if Advanced Allocation Management encounters a job that has a job name other than ABCJOB1, it adds additional volumes as directed by the VOL_ADD keywords because the INCLUDE statement specifies that jobs with job names that begin with ABCJOB are to be included in Advanced Allocation Management processing.

VOL_ADD_ALGORITHM

The VOL_ADD_ALGORITHM keyword specifies a volume selection algorithm to use when extending to a new volume.

Syntax



VOL_ADD_ALGORITHM=BESTFIT | FIRST | LARGESTSINGLE | MOSTFREESPACE | PERFORMANCE

Operands

BESTFIT
Select the volume (or volumes) with a single extent in which the requested amount is closest to the requested primary allocation quantity (for VSAM data sets) or the secondary allocation quantity (for non-VSAM data sets) without being less than the requested primary allocation quantity or secondary allocation quantity, as appropriate.

FIRST
Select the first volume (or volumes) with enough total freespace to satisfy the requested allocation amount.

Note: The allocation amount might not be able to be satisfied because the freespace needed might be spread out across more than five extents, which is the normal limit within allocation for satisfying a space request. Therefore, you might need to specify PRIM_REDUCE to help make the allocation successful.

LARGESTSINGLE
Select the volume (or volumes) with the largest single free extent.

Note: Advanced Allocation Management selects a volume with the largest single extent. This does not mean that the volume has enough space to satisfy the requested amount, it simply indicates that it is the volume with the largest single extent. Therefore, you might need to use the PRIM_REDUCE function to handle those situations where the selected volume does not have adequate space available.

MOSTFREESPACE
Select the volume with the most available freespace. The most freespace is defined as the total number of free cylinders or tracks on the volume, depending upon the allocation unit that is used. If cylinder allocation is specified, the total number of free cylinders is considered. Otherwise, the total number of free tracks is considered.

PERFORMANCE
Select the volume with the lowest fragmentation index, giving preference to those volumes that have enough available freespace to satisfy the allocation request.

Default

LARGESTSINGLE

Environments

EXTEND

Related keywords

VOL_ADD, VOL_ADD_LIM, VOL_ADD_MOUNT_USAGE, VOL_ADD_PROMPT, VOL_ADD_SGRP, VOL_ADD_UNIT, VOL_ADD_VGRP, VOL_ADD_VGRP_SEARCH, VOL_ADD_UPG_VOL, and VOL_ADD_VSER.

Example

In the following example, the volume with most available freespace is selected:

```
VOL_ADD_ALGORITHM=MOSTFREESPACE
```

VOL_ADD_LIM

The VOL_ADD_LIM keyword specifies the maximum number of volumes a data set is allowed to have when the VOL_ADD function is enabled.

Syntax

VOL_ADD_LIM=*xxx*

Where:

xxx
is the maximum number of volumes a data set is allowed to have when the VOL_ADD function is enabled. The valid range is 0 - 255.

Note: The VOL_ADD_LIM keyword is associated with the VOL_ADD keyword. You must specify `VOL_ADD=ENABLE` to implement this function.

Default

The default is 0. Specifying 0 indicates there is no limit to the maximum number of volumes except the limit that is implemented by z/OS.

Environments

EXTEND

Related keywords

VOL_ADD, VOL_ADD_ALGORITHM, VOL_ADD_MOUNT_USAGE, VOL_ADD_PROMPT, VOL_ADD_SGRP, VOL_ADD_UNIT, VOL_ADD_VGRP, VOL_ADD_VGRP_SEARCH, VOL_ADD_UPG_VOL, and VOL_ADD_VSER.

Example

In the following example, VOL_ADD_LIM is set to 10; therefore, Advanced Allocation Management does not allow the data set to exist on more than ten volumes. Thus, when VOL_ADD_LIM is set to 10, the data set is limited to a total of ten volumes, which means you can extend only to an additional nine volumes before Advanced Allocation Management allows the job to abend.

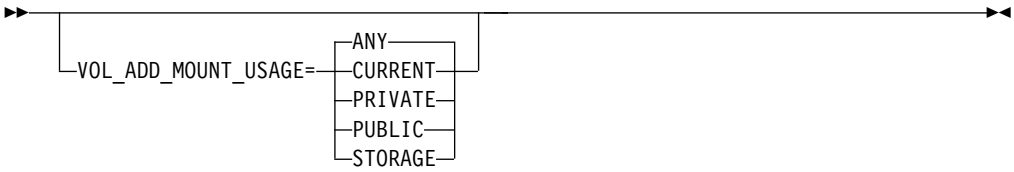
```
VOL_ADD_LIM=10
```

VOL_ADD_MOUNT_USAGE

The VOL_ADD_MOUNT_USAGE keyword acts as a filter for the volumes that are selected as part of the normal volume selection process that is used by Advanced Allocation Management. That is, Advanced Allocation Management selects a group of candidate volumes that are based on the VOL_ADD_VGRP, VOL_ADD_UNIT, and VOL_ADD_VSER keyword settings, and the default volume group. The

VOL_ADD_MOUNT_USAGE keyword value is then applied to that set of volumes to create a subset of eligible candidate volumes for the VOL_ADD function.

Syntax



VOL_ADD_MOUNT_USAGE=ANY | CURRENT | PRIVATE | PUBLIC | STORAGE

Operands

ANY
Any mount usage attribute is acceptable.

CURRENT
The mount usage attribute of the selected volume should be the same as the mount usage attribute of the current volume of the data set. The current volume of a data set is the volume that caused end-of-volume processing to occur, resulting in the invocation of the VOL_ADD function.

PRIVATE, PUBLIC, and STORAGE
The PRIVATE, PUBLIC, and STORAGE keyword values are equivalent to the mount usage attribute for a volume as defined in the Unit Control Block for a device, and as defined by z/OS.

Only one keyword value can be specified. For example:

Valid: VOL_ADD_MOUNT_USAGE=STORAGE

Not valid: VOL_ADD_MOUNT_USAGE=STORAGE,PUBLIC

Default

ANY

Environments

EXTEND

Related keywords

VOL_ADD, VOL_ADD_ALGORITHM, VOL_ADD_LIM, VOL_ADD_PROMPT, VOL_ADD_SGRP, VOL_ADD_UNIT, VOL_ADD_VGRP, VOL_ADD_VGRP_SEARCH, VOL_ADD_UPG_VOL, and VOL_ADD_VSER.

Example

In the following example, when a VOL_ADD is to be performed for a non-SMS-managed data set for job JJHTEST01, the volumes that are considered by Advanced Allocation Management are all of the volumes that have a mount usage attribute of STORAGE in the volume group that is named GROUP01 and the default volume group:

```
INCLUDE JOBNAME=JJHTEST01
                VOL_ADD=ENABLE
                VOL_ADD_VGRP=GROUP01
                VOL_ADD_MOUNT_USAGE=STORAGE
```

VOL_ADD_PROMPT

The VOL_ADD_PROMPT keyword specifies whether to notify the operator by using a WTOR (message GLO2137A) when a volume cannot be found to satisfy the VOL_ADD function.

Syntax

VOL_ADD_PROMPT=ENABLE | DISABLE

Operands

ENABLE
Prompt the operator.

DISABLE
Do not prompt the operator.

Usage and processing considerations

1. The VOL_ADD_PROMPT keyword is associated with the VOL_ADD keyword. You must specify `VOL_ADD=ENABLE` to implement this function.
2. The VOL_ADD_PROMPT keyword does not apply to SMS data sets.

Default

DISABLE

Environments

EXTEND

Related keywords

VOL_ADD, VOL_ADD_ALGORITHM, VOL_ADD_LIM, VOL_ADD_MOUNT_USAGE, VOL_ADD_SGRP, VOL_ADD_UNIT, VOL_ADD_VGRP, VOL_ADD_VGRP_SEARCH, VOL_ADD_UPG_VOL, and VOL_ADD_VSER.

Example

In the following example, VOL_ADD_PROMPT is set to ENABLE; therefore, a WTOR notifies the operator when a volume cannot be found to satisfy the VOL_ADD function:

```
VOL_ADD_PROMPT=ENABLE
```

VOL_ADD_SGRP

The VOL_ADD_SGRP keyword determines the default volume group Advanced Allocation Management should use for VOL_ADD.

Syntax

VOL_ADD_SGRP=FIRST | CURRENT

Operands

Specify one of the following operands:

CURRENT
Use the volume group that is associated with the current (last) volume.

If you specify VOL_ADD_SGRP=CURRENT, Advanced Allocation Management uses the current volume of the data set (which might be different than the first volume if the data set has more than one volume) to perform the search for the default volume group.

When you specify VOL_ADD_SGRP=CURRENT, you do not need to define a volume group using the VOL_ADD_VGRP keyword on the INCLUDE statement.

FIRST
Use the first volume that is assigned to the data set to search the VGRPDEFS to locate a volume group to use. The first volume group that is found (which includes the first volume of the data set) is used as the default volume group.

Usage and processing considerations

When searching for a volume for VOL_ADD, Advanced Allocation Management considers all volumes in the default volume group, and other volumes that are explicitly specified on the INCLUDE statement by the VOL_ADD_VGRP, VOL_ADD_UNIT, and VOL_ADD_VSER keywords before it selects a volume. Advanced Allocation Management uses the volume with the largest amount of freespace for the VOL_ADD. Advanced Allocation Management does not stop searching volume groups when it locates a volume with a suitable amount of space (it continues to search all volumes from all volume groups and other sources for the volume with the largest amount of freespace).

Notes:

1. The VOL_ADD_SGRP keyword is associated with the VOL_ADD keyword. You must specify VOL_ADD=ENABLE to implement this function.
2. The VOL_ADD_SGRP keyword does not apply to SMS data sets.

Default

FIRST

Environments

EXTEND

Related keywords

VOL_ADD, VOL_ADD_ALGORITHM, VOL_ADD_LIM, VOL_ADD_MOUNT_USAGE, VOL_ADD_PROMPT, VOL_ADD_UNIT,

VOL_ADD_VGRP, VOL_ADD_VGRP_SEARCH, VOL_ADD_UPG_VOL, and VOL_ADD_VSER.

Example

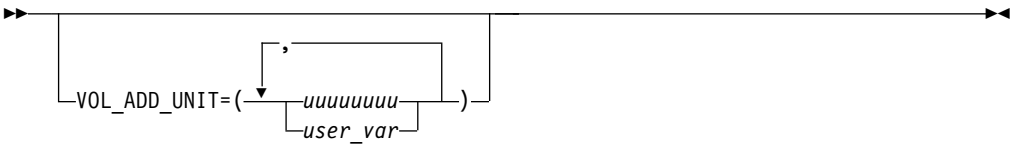
In the following example, the volume group that is associated with the current (last) volume is used:

```
VOL_ADD_SGRP=CURRENT
```

VOL_ADD_UNIT

The VOL_ADD_UNIT keyword specifies esoteric unit names or generic names to use with the VOL_ADD keyword.

Syntax



VOL_ADD_UNIT=(*uuuuuuuu* | *user_var*, *uuuuuuuu* | *user_var*,.....)

Where:

uuuuuuuu
is a unit name. Each name (*uuuuuuuu*) can be a maximum of eight characters in length.

Usage and processing considerations

1. The VOL_ADD_UNIT keyword is associated with the VOL_ADD keyword. You must specify `VOL_ADD=ENABLE` to implement this function.
2. The VOL_ADD_UNIT keyword does not apply to SMS data sets.

Default

None.

Environments

EXTEND

Related keywords

VOL_ADD, VOL_ADD_ALGORITHM, VOL_ADD_LIM, VOL_ADD_MOUNT_USAGE, VOL_ADD_PROMPT, VOL_ADD_SGRP, VOL_ADD_VGRP, VOL_ADD_VGRP_SEARCH, VOL_ADD_UPG_VOL, and VOL_ADD_VSER.

Example: VOL_ADD_UNIT set to RDVOLS

In the following example, VOL_ADD_UNIT is set to RDVOLS. Therefore, Advanced Allocation Management uses the unit name RDVOLS with the VOL_ADD keyword to determine where to obtain additional volumes.

```
VOL_ADD_UNIT=RDVOLS
```

Example: VOL_ADD_UNIT set to a user-defined variable

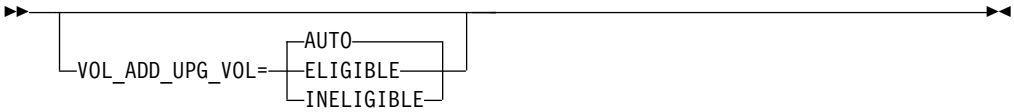
In the following example, VOL_ADD_UNIT is set to the user-defined variable &VNAMES. Therefore, Advanced Allocation Management uses the unit name as defined by &VNAMES with the VOL_ADD keyword to determine where to obtain additional volumes.

```
VOL_ADD_UNIT=&VNAMES
```

VOL_ADD_UPG_VOL

The VOL_ADD_UPG_VOL keyword specifies whether to include or exclude volumes that are allocated to the base cluster and upgrade set for a non-SMS VSAM cluster as eligible for VOL_ADD processing for individual components. The purpose of the VOL_ADD_UPG_VOL keyword is to help improve performance when accessing the components of a base cluster and its upgrade set by extending components outside of the set of volumes that are already allocated, therefore minimizing contention among components.

Syntax



VOL_ADD_UPG_VOL=AUTO | ELIGIBLE | INELIGIBLE

Operands

Specify one of the following operands:

AUTO
When the VOL_ADD function searches for a candidate volume for a component that is allocated to a non-SMS VSAM cluster or a member of its upgrade set during VOL_ADD processing, it first attempts to locate a suitable candidate volume within the set of volumes that are defined by the VOL_ADD_VSER, VOL_ADD_UNIT, and VOL_ADD_VGRP keywords (it excludes volumes that are allocated to the base cluster or its upgrade set). If the VOL_ADD function cannot find a suitable volume within the set of volumes, it expands the set of eligible volumes to include all of the volumes that are allocated to the base cluster and its upgrade set.

ELIGIBLE
When VOL_ADD searches for a suitable candidate volume to extend a component of the base cluster or its upgrade set, all volumes that are allocated to the base cluster and upgrade set for a non-SMS VSAM cluster are eligible candidate volumes for VOL_ADD processing along with the volumes that are defined by the VOL_ADD_VSER, VOL_ADD_UNIT, and VOL_ADD_VGRP keywords.

INELIGIBLE
When VOL_ADD searches for a suitable candidate volume to extend a component of the base cluster or its upgrade set, all volumes that are allocated to the base cluster and upgrade set for a non-SMS VSAM cluster are considered ineligible as candidate volumes for VOL_ADD processing. Only those volumes that are defined by the VOL_ADD_VSER, VOL_ADD_UNIT, and VOL_ADD_VGRP keywords are considered to be eligible.

Default

AUTO

Environments

EXTEND

Usage and processing considerations

- The VOL_ADD_UPG_VOL keyword is associated with the VOL_ADD keyword. You must specify `VOL_ADD=ENABLE` to implement the VOL_ADD_UPG_VOL function.
- The VOL_ADD_UPG_VOL keyword does not apply to SMS data sets.
- When Advanced Allocation Management considers the volumes that are allocated to a component of a VSAM data set, it includes primary and candidate volumes regardless of whether data exists on the volumes.

  For example, consider a base cluster with its index and data components on volume VOL001. The base cluster has one alternate index that is associated with it in the upgrade set, with its index and data components on VOL002. Presume that the data component of the base cluster must extend to a new volume. Also presume that the volume group that is associated with the extend consists of the volumes VOL001, VOL002, and VOL003. If you set VOL_ADD_UPG_VOL to `INELIGIBLE` even if space is available on VOL002, only VOL003 is considered as a candidate for the extend because the data set exists on VOL001, and VOL002 is part of the set of volumes that are allocated to the base cluster and the upgrade set. If you set VOL_ADD_UPG_VOL to `ELIGIBLE`, VOL002 and VOL003 are considered as candidate volumes for the extend.

Related keywords

VOL_ADD, VOL_ADD_ALGORITHM, VOL_ADD_LIM, VOL_ADD_MOUNT_USAGE, VOL_ADD_PROMPT, VOL_ADD_SGRP, VOL_ADD_UNIT, VOL_ADD_VGRP, VOL_ADD_VGRP_SEARCH, and VOL_ADD_VSER.

Example

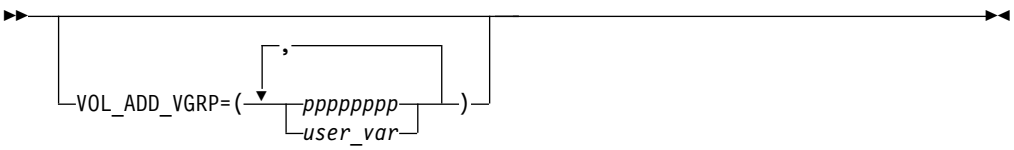
In the following example, the set of eligible volumes to be considered by VOL_ADD processing when a component of base cluster and its upgrade set are extended, includes all volumes that are allocated to the base cluster and its upgrade set.

```
VOL_ADD_UPG_VOL=ELIGIBLE
```

VOL_ADD_VGRP

The VOL_ADD_VGRP keyword specifies a name of the volume groups to use with the VOL_ADD keyword.

Syntax



VOL_ADD_VGRP=(*ppppppppp* | *user_var*, *ppppppppp* | *user_var*,.....)

Where:

ppppppppp
is a name of a volume group. Each volume group name (*ppppppppp*) can be a maximum of eight characters in length.

Usage and processing considerations

1. The VOL_ADD_VGRP keyword is associated with the VOL_ADD keyword. You must specify `VOL_ADD=ENABLE` to implement this function.
2. The VOL_ADD_VGRP keyword does not apply to SMS data sets.

Default

None.

Note: When no volume group is defined, Advanced Allocation Management uses a default volume group. The default volume group is chosen using the first, or current, volser for a data set as specified, or defaulted to in the VOL_ADD_SGRP keyword. Advanced Allocation Management uses the particular volume to search the volume group definitions in order (from first to last) looking for a volume group that contains the volser. That volume group is used as the default volume group from which a volume is selected for volume addition based on the selected or defaulted VOL_ADD_ALGORITHM.

Environments

EXTEND

Related keywords

VOL_ADD, VOL_ADD_ALGORITHM, VOL_ADD_LIM, VOL_ADD_MOUNT_USAGE, VOL_ADD_PROMPT, VOL_ADD_SGRP, VOL_ADD_UNIT, VOL_ADD_VGRP_SEARCH, VOL_ADD_UPG_VOL, and VOL_ADD_VSER.

Example: Multiple volume groups

In the following example, VOL_ADD_VGRP lists the volume groups VGRP0001 and VGRP0002. Therefore, the VGRP0001 and VGRP0002 volume groups are used with the VOL_ADD keyword to determine which volume groups Advanced Allocation Management uses to obtain additional volumes.

```
VOL_ADD_VGRP=(VGRP0001,VGRP0002)
```

Example: Volume groups defined by user-defined variables

In the following example, VOL_ADD_VGRP lists the user-defined variables &GRPA, &GRPB, and &GRPC. The volume groups that are defined by these variables are used with the VOL_ADD keyword to determine which volume groups Advanced Allocation Management uses to obtain additional volumes.

```
VOL_ADD_VGRP=(&GRPA,&GRPB,&GRPC)
```

VOL_ADD_VGRP_SEARCH

The VOL__ADD_VGRP_SEARCH keyword specifies the volume selection volume group search method when extending to a new volume.

Syntax

VOL__ADD_VGRP_SEARCH=ALL | FIRST

Operands

Specify one of the following operands:

ALL
Search all volume groups for a suitable volume.

FIRST
Stop searching for volumes after the volume group that was just searched contains a volume that meets the criterion that is established by the VOL_ADD_VGRP_ALGORITHM keyword.

Note: Other than BESTFIT (when all volumes requested can be satisfied using the BESTFIT algorithm) there is no guarantee that the space that was requested is available. For example, if you specify `VOL_ADD_VGRP_ALGORITHM= LARGESTSINGLE` and Advanced Allocation Management selects a volume with the largest single extent in a volume group, it does not mean that the volume has enough space to satisfy the requested amount, it simply means that it is the volume with the largest single extent. Therefore, you might need to use the PRIM_REDUCE function to handle situations where the selected volume does not have adequate space available.

Default

ALL

Environments

EXTEND

Related keywords

VOL_ADD, VOL_ADD_ALGORITHM, VOL_ADD_LIM, VOL_ADD_MOUNT_USAGE, VOL_ADD_PROMPT, VOL_ADD_SGRP, VOL_ADD_UNIT, VOL_ADD_VGRP, VOL_ADD_UPG_VOL, and VOL_ADD_VSER.

Example

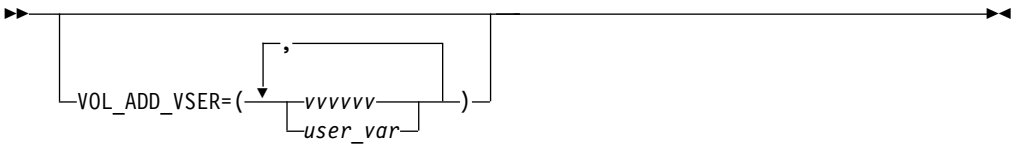
In the following example, Advanced Allocation Management stops searching for volumes if a suitable volume exists in the volume group in which it is searching:

```
VOL__ADD_VGRP_SEARCH=FIRST
```

VOL_ADD_VSER

The VOL_ADD_VSER keyword specifies the volume serial numbers to use with the VOL_ADD keyword.

Syntax



VOL_ADD_VSER=(*vvvvvv* | *user_var*, *vvvvvv* | *user_var*,.....)

Where:

vvvvvv
: is a serial number. Each serial number (*vvvvvv*) can be a maximum of six characters in length.

Usage and processing considerations

1. The VOL_ADD_VSER keyword is associated with the VOL_ADD keyword. You must specify `VOL_ADD=ENABLE` to implement this function.
2. The VOL_ADD_VSER keyword does not apply to SMS data sets.

Default

None.

Environments

EXTEND

Related keywords

VOL_ADD, VOL_ADD_ALGORITHM, VOL_ADD_LIM, VOL_ADD_MOUNT_USAGE, VOL_ADD_PROMPT, VOL_ADD_SGRP, VOL_ADD_UNIT, VOL_ADD_VGRP, VOL_ADD_UPG_VOL, and VOL_ADD_VGRP_SEARCH.

Example: Multiple volume serial numbers

In the following example, VOL_ADD_VSER lists the volume serial numbers DASD01, DASD02. Therefore, Advanced Allocation Management uses these volume serial numbers with the VOL_ADD keyword to determine where to obtain additional volumes.

```
VOL_ADD_VSER=(DASD01, DASD02)
```

Note: The order in which you specify the volume serial numbers does not affect processing.

Example: Volume serial numbers defined by user-defined variables

In the following example, VOL_ADD_VSER lists the user-defined variable &DASDA. Therefore, Advanced Allocation Management uses the volume serial

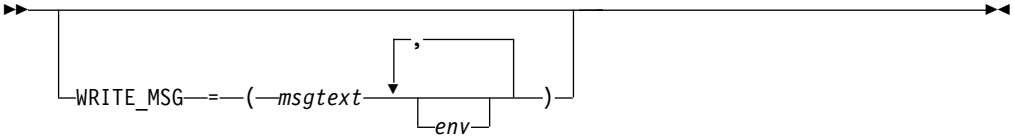
numbers as defined by the variable &DASDA with the VOL_ADD keyword to determine where to obtain additional volumes.

```
VOL_ADD_VSER=(&DASDA)
```

Output a message (WRITE_MSG)

Specify the WRITE_MSG keyword to output a message upon execution of the actions for a rule definition.

Syntax



WRITE_MSG=(*msgtext*,*env*...)

Where:

msgtext
is the message to write to the destination that is specified in the MSG_DESTINATION function. If blanks are included in the message, surround the message text with double quotation marks. The maximum length of the message, including quotation marks, is 120 characters.

env
(Optional.) Specifies the environments in which to display the message. You can specify 0 - 7 environment keywords. The following options are valid:

ACS
The point in SMS processing directly following the assignment of one or more of the SMS constructs of Data Class, Storage Class, Management Class, or Storage Group.

ALL
(Default.) All points where Advanced Allocation Management intercepts processing.

ALLOCATE
The point in processing where a new data set is allocated (including VSAM and non-VSAM, SMS, and non-SMS data sets).

CATCHECK
The point in allocation processing where the determination of the prior existence of a data set is made. For SMS data sets, this point is just before the allocation of a data set. For non-SMS data sets, this point can be just before the allocation of a data set (or at unallocation processing time during catalog disposition processing).

EXTEND
The point in processing where a new extent is added to a data set, either on the current volume or on a new volume (including volume addition).

STOW
The point in PDS processing when a new member is being added to a data set and the directory is being updated.

UNALLOCATE
The point in processing where a data set is unallocated.

Note:

- The message text can contain the following symbolic variables, which are replaced with the appropriate text when the message is generated. The total length of the message, after the symbolic variables are expanded, cannot exceed 120 characters. Text that exceeds the maximum length is truncated.

  &JB Jobname.

  &ST Stepname.

  &PS Procedure stepname.

  &DS Data set name.

  &DD DDNAME.

  &EV Environment. This variable displays the same values for Environment as described for *env*.

  &LN The rule definition line number that is associated with the rule definition that contains the WRITE_MSG keyword for which the selection criteria matched.

  &SS The Advanced Allocation Management subsystem identification.

  When the symbolic variables are replaced with actual values, trailing blanks are removed. If a symbolic variable does not have an associated value (such as Procedure stepname), a single asterisk is displayed.

  Specifying two ampersands in a row displays a single ampersand character (&) in the output message.
- The specification of the first eight characters can use the same rules that are used for other Advanced Allocation Management messages. If the first three characters are GLO and the next four characters are numeric characters, the eighth character can be used to define the circumstances under which the message is displayed:
  - If the eighth character is an E, the message is always displayed.
  - If the eighth character is an I, the message is displayed only if the MSG_LEVEL function is set to ALL.
  - If the eighth character is set to S, the message is always displayed and the S is replaced with an I.

  If the first three characters are not GLO and the next four characters are not numeric, the message is always displayed.
- To continue the message text to another line, surround the text with double quotation marks. Continue the text to (and including) column 72. On the next line, continue the message text at column one.

  For example:

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
--------------------------------------------------------------------------------

INCLUDE JOBNAME=JJHTEST1
      WRITE_MSG="THIS VERY LONG SAMPLE MESSAGE TEXT IS GOING TO BE CONTINUED ON
THE NEXT LINE"
```

Default

ALL

Environments

ACS, ALLOCATE, CATCHECK, EXTEND, STOW, and UNALLOCATE.

Example

In the following example, the message: `Do not allocate TEST data sets in VGRP=PROD` is generated:

```
WRITE_MSG=("Do not allocate TEST data sets in VGRP=PROD")
```

Enhanced Allocation Management functions

To set allocation attributes to specific values, use Enhanced Allocation Management function keywords.

Note:

- The underlined portion of the keyword indicates the minimally acceptable abbreviation.
- $NULL indicates that a particular attribute value should be set such that it appears that it was not specified by the user.
- The default that is listed is the default for the Advanced Allocation Management keyword, it is *not* the default that is assigned to the attribute by the operating system (unless otherwise specifically noted).
- Where applicable, the JCL, IDCAMS, data, management, and storage class equivalents are provided. For the data, management, or storage class equivalents, the field name from the appropriate ISMF panel that is used to define the class is listed.
- The use of numeric suffixes is permitted.
- When you use the Enhanced Allocation Management functions, it is your responsibility to ensure that the values are consistent and valid to the operating system.

$DCO

The $DCO keyword specifies whether the data class override feature should be active for all Advanced Allocation Management supported data class variables. It specifies whether the data class values for various attributes should override specified JCL or DEFINE specified values.

Syntax

$DCO=CONDITIONAL | DISABLE | UNCONDITIONAL

Operands

Specify one of the following operands:

CONDITIONAL
Override JCL or DEFINE values only if the corresponding data class value is not null.

DISABLE
Turn off data class override unless explicitly specified at the individual variable level.

UNCONDITIONAL
Override the JCL or DEFINE values even if the corresponding data class value is null, in which case the variable in the JCL or DEFINE is also set (as if it had not been specified).

$DCO supports the following values within the data class:

- Access Bias
- Control Area Freespace
- Control Interval Freespace
- Data Control Interval Size
- Data Set Organization
- Directory Blocks
- DSNTYPE
- EATTR
- Expiration Date
- Extended Addressability
- FRLOG
- Key Length
- Key Offset
- Logstream ID
- Primary Quantity
- Record Format
- Record Length
- Record Organization
- Retention Period
- Reuse
- Secondary Quantity
- Space Allocation Units (AVGREC)
- Spanned
- Speed or Recovery
- Tape Compaction
- Volume Count
- Xrgn Shareoption
- Xsys Shareoption

Default

None.

Environments

ALLOCATE

JCL equivalent

Refer to each individual attribute supported by Advanced Allocation Management.

Example

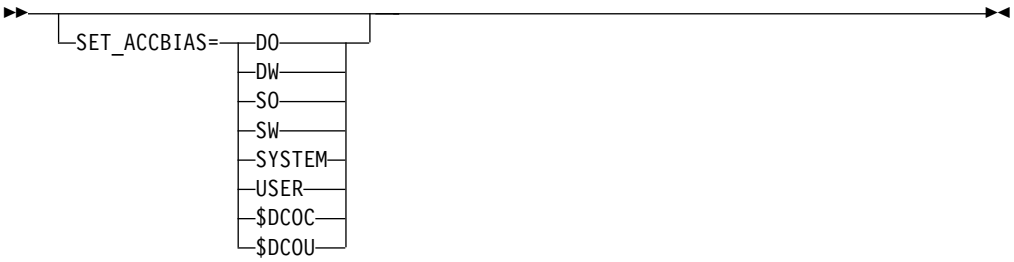
In the following example, only JCL or DEFINE values are overridden if the corresponding data class value is not null:

```
$DCO=C
```

SET_ACCBIAS

(VSAM only.) The SET_ACCBIAS keyword specifies the assigned VSAM System Managed Buffering specification.

Syntax



SET_ACCBIAS=DO | DW | SO | SW | SYSTEM | USER | $DCOC | $DCOU

Operands

Specify one of the following operands:

DO System-managed buffering with direct optimization.

DW System-managed buffering that is weighted for direct processing.

SO System-managed buffering with sequential optimization.

SW System-managed buffering that is weighted for sequential processing.

SYSTEM
System-managed buffering chooses the buffering technique.

USER
System-managed buffering is not used.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD AMP=('ACCBIAS=USER')
```

Data class equivalent

Record Access Bias.

Related selection criteria

ACCBIAS

Example

The following example specifies system-managed buffering that is weighted for direct processing:

```
SET_ACCBIAS=DW
```

SET_ADISP

The SET_ADISP keyword specifies the abnormal termination disposition for a data set.

Syntax

SET_ADISP=CATLG | DELETE | KEEP | UNCATLG

Operands

Specify one of the following operands:

CATLG
The specified abnormal disposition is DISP=(,,CATLG).

DELETE
The specified abnormal disposition is DISP=(,,DELETE).

KEEP
The specified abnormal disposition is DISP=(,,KEEP).

UNCATLG
The specified abnormal disposition is DISP=(,,UNCATLG).

Note:

1. Wildcard characters are not permitted.
2. The SET_ADISP keyword is not applicable to VSAM IDCAMS DEFINE statements.
3. The system takes the following defaults with the abnormal termination disposition parameter:
   - If you omit the abnormal termination disposition, the default is the disposition that is specified, or defaulted to, by the normal termination disposition. If the normal termination disposition is PASS, the default abnormal termination disposition is DELETE for a NEW data set, or KEEP for an existing data set.

- If you omit the DISP parameter, the default disposition is DELETE for the abnormal termination disposition.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD DISP=(,,DELETE)
```

Related selection criteria

ADISP

Example

In the following example, the specified abnormal disposition is DISP=(,,DELETE):

```
SET_ADISP=DELETE
```

SET_BLKSIZE

(Non-VSAM only.) The SET_BLKSIZE keyword specifies the block size for a data set.

Syntax

SET_BLKSIZE=*nnnnnnnnnn*

Where:

nnnnnnnnnn is an integer (0 - 2147483647).

Note: If you set this value to 0, you can cause the system to calculate the system-determined blocksize as determined under the circumstances.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD DD01 BLKSIZE=256000
```

Related selection criteria

BLKSIZE

Example

In the following example, the specified blocksize is 256000:

```
SET_BLKSIZE=256000
```

SET_BUFSPC

(VSAM only.) The SET_BUFSPC keyword specifies the buffer space specification for a data set.

Syntax

SET_BUFSPC=*nnnnnnnnnn*

Where:

nnnnnnnnnn is an integer (0 - 2147483647).

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD DD01 AMP=('BUFSP=128000')
```

IDCAMS equivalent

```
DEFINE CLUSTER(BUFFERSPACE(256000) ...)
```

Related selection criteria

BUFSPC

Example

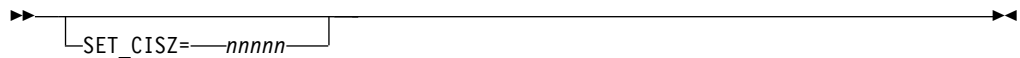
In the following example, the buffer space specification is 256000:

```
SET_BUFSPC=256000
```

SET_CISZ

(VSAM only.) The SET_CISZ keyword specifies the control interval size for a VSAM cluster.

Syntax



SET_CISZ=*nnnnn*

Where:

nnnnn is an integer (0 - 32768).

Default

None.

Environments

ALLOCATE

AMS equivalent

```
DEFINE CLUSTER (CONTROLINTERVALSIZE(8192) ....)
```

Related selection criteria

CISZ

Example

In the following example, the specified control interval size is 8192:

```
SET_CISZ=8192
```

SET_CISZ_DATA

(VSAM only.) The SET_CISZ_DATA keyword specifies the control interval size for the data component of a VSAM cluster.

Syntax

SET_CISZ_DATA=*nnnnn* | $DCOC | $DCOU

Where:

nnnnn is an integer (0 - 32768).

Default

None.

Environments

ALLOCATE.

AMS equivalent

```
DEFINE CLUSTER (....) DATA (CONTROLINTERVALSIZE(8192) ....)
```

Data class equivalent

CIsize Data.

Related selection criteria

CISZ_DATA.

Example

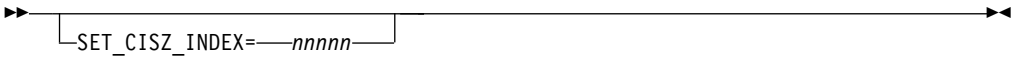
In the following example, the specified control interval size for the data component is 8192:

```
SET_CISZ_DATA=8192
```

SET_CISZ_INDEX

(VSAM only.) The SET_CISZ_INDEX keyword specifies the control interval size for the index component of a VSAM cluster.

Syntax



SET_CISZ_INDEX=*nnnnn*

Where:

nnnnn is an integer (0 - 32768).

Default

None.

Environments

ALLOCATE

AMS equivalent

```
DEFINE CLUSTER (....) DATA(....) INDEX(CONTROLINTERVALSIZE(8192) ....)
```

Data class equivalent

CIsize Data.

Related selection criteria

CISZ_DATA

Example

In the following example, the specified control interval size for the index component is 8192:

```
SET_CISZ_INDEX=8192
```

SET_CONTIGUOUS

(Non-VSAM disk data sets only.) The SET_CONTIGUOUS keyword specifies whether the CONTIG option should be specified when allocating a new data set.

Syntax

SET_CONTIGUOUS=YES|NO

Operands

YES
Specify CONTIG.

NO Do not specify CONTIG.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD SPACE=(CYL,(1,1),,CONTIG)
```

Related selection criteria

CONTIGUOUS

Example

In the following example, CONTIG should be specified when allocating a new data set:

```
SET_CONTIG=Y
```

SET_DATACLAS

The SET_DATACLAS keyword specifies the SMS data class to be assigned to a data set.

Syntax

SET_DATACLAS=*dataclas*|$NULL

Where:

dataclas is the 1 - 8 character SMS data class name. Wildcard characters are not permitted.

Note:

- As permitted in SMS, you can assign a data class to a non-SMS data set.

- This keyword requires additional calls to SMS Configuration Services and its presence anywhere in a rule definitions member can cause a measurable increase in CPU time for all tasks in the system, especially tasks that perform large numbers of allocations. Therefore, carefully consider the specification of SET_DATACLAS in your rule definitions member.

Default

None.

Environments

ACS

JCL equivalent

```
DD01 DD DATACLAS=SMALL
```

IDCAMS equivalent

```
DEFINE CLUSTER (NAME(...) DATACLASS(SMALL) ...)
```

Related selection criteria

DATACLAS

Example

In the following example, "SMALL" is the SMS data class name to be assigned to the data set:

```
SET_DATACLAS=SMALL
```

SET_DEFERMOUNT

The SET_DEFERMOUNT keyword specifies whether the DEFER option should be specified for the UNIT keyword.

Syntax

SET_DEFERMOUNT=YES|NO

Operands

Specify one of the following operands:

YES
DEFER should be specified.

NO DEFER should not be specified.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD UNIT=(3490,,DEFER)
```

Related selection criteria

DEFERMOUNT

Example

In the following example, the DEFER option should be specified for the UNIT keyword:

```
SET_DEFER=Y
```

SET_DIRBLKS

(PDS and PDS/E data sets only.) The SET_DIRBLKS keyword specifies the number of directory blocks that are assigned to a PDS or PDS/E data set.

Syntax

SET_DIRBLKS=*nnnnn* | $DCOC | $DCOU

Where:

nnnnn is an integer (0 - 65535) that specifies the number of directory blocks.

| **Note:** The $DCOC and $DCOU keywords are only valid for a SPACEUNITS value
| of AVGRECU, AVGRECK, or AVGRECM.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD SPACE=(,(,,120))
```

Data class equivalent

Space Directory

Related selection criteria

DIRBLKS

Example

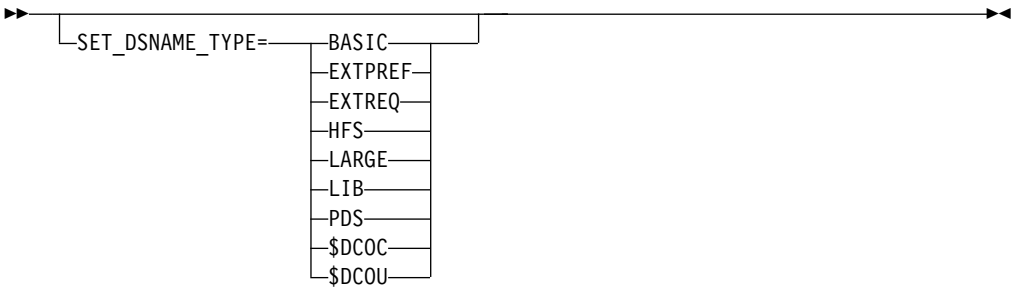
In the following example, the number of directory blocks that are assigned is 120:

```
SET_DIRBLKS=120
```

SET_DSNAME_TYPE

The SET_DSNAME_TYPE keyword specifies the data set type.

Syntax



SET_DSNAME_TYPE=BASIC | EXTPREF | EXTREQ | HFS | LARGE | LIB | PDS | $DCOC | $DCOU

Operands

Specify one of the following operands:

BASIC
Allows the specification of less than 65535 tracks for the primary and secondary allocation for non-VSAM data sets.

EXTPREF
An extended format is preferred if the data set is VSAM, sequential, or if DSORG is omitted from all sources.

EXTREQ
Extended format is required if the data set is VSAM, sequential, or if DSORG is omitted from all sources.

HFS
Hierarchical file system data set (HFS data set).

LARGE
Allows the specification of more than 65535 tracks for the primary and secondary allocation for non-VSAM data sets.

LIB
PDSE data set.

PDS
Partitioned data set (PDS).

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD DSNTYPE=HFS
```

Data class equivalent

Data Set Name Type and If Ext parameters.

Related selection criteria

DSNAME_TYPE

Example

In the following example, the data set type is PDSE:

```
SET_DSNAME_TYPE=LIB
```

SET_DSORG

The SET_DSORG keyword specifies the data set organization for non-VSAM data sets and the record organization for VSAM data sets.

Syntax

SET_DSORG=DA | ESDS | KSDS | LDS | PO | PS | RRDS | $DCOC | $DCOU

Operands

Specify one of the following operands:

DA Direct access.

ESDS
VSAM entry-sequenced data set.

KSDS
VSAM key-sequenced data set.

LDS
VSAM linear data set.

PO Partitioned organized.

PS Physical sequential.

RRDS
A VSAM variable-length or fixed-length relative-record data set.

Notes:

1. Wildcard characters are not permitted.
2. The $DCOU and $DCOC keywords are only valid for KSDS, ESDS, RRDS, and LDS VSAM data sets.
3. The Enhanced Allocation Management function keywords that pertain to VSAM record organization (KSDS, ESDS, RRDS, LDS) do not apply to VSAM IDCAMS DEFINE; however, they are applicable to VSAM data sets that are created through JCL.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD DSORG=PS
DD01 DD RECORG=KS
```

Data class equivalent

Recorg

Related selection criteria

DSORG

Example

In the following example, the specified organization is physical sequential:

```
SET_DSORG=PS
```

SET_EATTR

The SET_EATTR keyword specifies the Extended Attributes for a VSAM cluster or non-VSAM data set.

Syntax

SET_EATTR=NO | OPT | $NULL | $DCOU | $DCOC

Note: For non-VSAM data sets, the $NULL and $DCOU options are not supported. If you specify $NULL and $DCOU for non-VSAM data sets, they are ignored.

Operands

NO Set Extended Attributes to NO for the data set.

OPT
Set Extended Attributes to OPT (Optional) for the data set.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD EATTR=OPT,...
```

IDCAMS equivalent

```
DEFINE CLUSTER(EATTR(OPT)...)
```

Data class equivalent

EATTR

Related selection criteria

EATTR

Example

In the following example, Extended Attributes is set to OPT:

```
SET_EATTR=OPT
```

SET_ERASE

(VSAM data sets only.) The SET_ERASE keyword specifies whether the VSAM ERASE or NOERASE option should be specified.

Syntax

SET_ERASE=YES | NO

Operands

Specify one of the following operands:

YES
Specify ERASE.

NO Specify NOERASE.

Default

None.

Environments

ALLOCATE

IDCAMS equivalent

```
DEFINE CL (ERASE ....)
DEFINE CL (NOERASE....)
```

Data class equivalent

None.

Related selection criteria

ERASE

Example

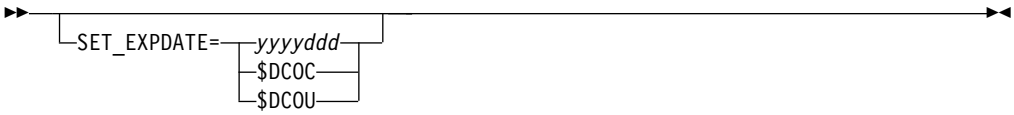
In the following example, the VSAM ERASE option should be specified:

```
SET_ERASE=YES
```

SET_EXPDATE

The SET_EXPDATE keyword specifies an expiration date for a data set.

Syntax



SET_EXPDATE=*yyyyddd* | $DCOC | $DCOU

Where:

yyyyddd is a Julian date and *yyyy* is a four-digit year and *ddd* is a three-digit day.

Note: If you specify SET_EXPDATE and SET_RETPD, SET_EXPDATE takes precedence.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD EXPDT=2009/003
```

IDCAMS equivalent

```
DEFINE CLUSTER(TO(2009003) ..)
```

Data class equivalent

Retpd or Expdt.

Related selection criteria

EXPDATE

Example

In the following example, the expiration date is the third day of 2009:

```
SET_EXPDATE=2009003
```

SET_EXTADDR

(VSAM data sets only.) The SET_EXTADDR keyword specifies the Extended Addressability attribute for a VSAM cluster.

Syntax

SET_EXTADDR=YES | NO | $DCOC | $DCOU

Note: For this VSAM option, the ISMF Data Class Define panel always defaults the value for this field to N if a blank is specified. Therefore, because a value is always defined, the $DCOC option acts the same as the $DCOU option.

Operands

YES
Define Extended addressability for the data set.

NO Do *not* define Extended addressability for the data set.

Note: The system honors EXTADDR=YES only if the cluster is also defined as an extended format data set.

Default

SMS defaults to NO for Extended Addressability.

Environments

ALLOCATE

Data class equivalent

Extended Addressability.

Related selection criteria

EXTADDR

Example

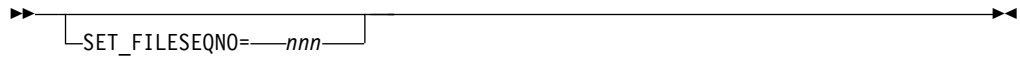
In the following example, Extended addressability should not be defined for the data set:

```
SET_EXTADDR=N
```

SET_FILESEQNO

(Tape data sets only.) The SET_FILESEQNO keyword specifies a file sequence number of a data set.

Syntax



SET_FILESEQNO=*nnn*

Where:

nnn is the sequence number (0 - 255) of the data set on the tape.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD LABEL=(4,)
```

Related selection criteria

FILESEQNO

Example

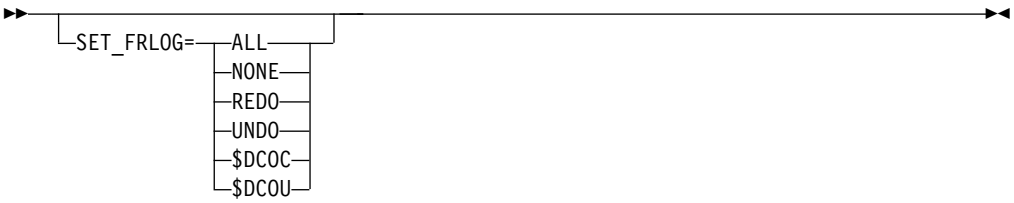
In the following example, the file sequence number is 4:

```
SET_FILESEQNO=4
```

SET_FRLOG

(VSAM data sets only.) The SET_FRLOG keyword specifies whether VSAM batch logging should be specified for a VSAM data set.

Syntax



SET_FRLOG=ALL | NONE | REDO | UNDO | $DCOC | $DCOU

Operands

Specify one of the following operands:

ALL
Both backward and forward recovery should be enabled.

NONE
The VSAM batch logging option should not be performed.

REDO
Forward recovery should be enabled.

UNDO
Backward recovery should be enabled.

Default

None. (This keyword has no default.)

Environments

ALLOCATE

IDCAMS equivalent

```
DEFINE CL (FRLOG(REDO) ...)
```

Data class equivalent

FRlog

Related selection criteria

FRLOG

Example

In the following example, forward recovery should be enabled:

```
SET_FRLOG=REDO
```

SET_FSPC_CA

(VSAM data sets only.) The SET_FSPC_CA keyword specifies the freespace percentage for the control area of a VSAM data set.

Syntax

SET_FSPC_CA=*nnn* | $DCOC | $DCOU

Where:

nnn is an integer that specifies the free space percentage (0 - 100).

Default

None.

Environments

ALLOCATE

IDCAMS equivalent

```
DEFINE CL (FREESPACE(0 10) ...)
```

Data class equivalent

% Freespace CA

Related selection criteria

FSPC_CA

Example

In the following example, the specified free space percentage is 10 (10%):

```
SET_FSPC_CA=10
```

SET_FSPC_CI

(VSAM data sets only.) The SET_FSPC_CI keyword specifies the freespace percentage for the control interval of a VSAM data set.

Syntax

SET_FSPC_CI=*nnn* | $DCOC | $DCOU

Where:

nnn is an integer that specifies the free space percentage (0 - 100).

Default

None.

Environments

ALLOCATE

IDCAMS equivalent

```
DEFINE CL (FREESPACE(10 0) ...)
```

Data class equivalent

% Freespace CI

Related selection criteria

FSPC_CI

Example

In the following example, the specified free space percentage is 10 (10%):

```
SET_FSPC_CI=10
```

SET_ILOAD

(VSAM data sets only.) The SET_ILOAD keyword specifies the initial load option to be specified for a VSAM data set.

Syntax

SET_ILOAD=SPEED | RECOVERY | $DCOC | $DCOU

Note: For this VSAM option, the ISMF Data Class Define panel always defaults the value for this field to R (Recovery) if a blank is specified. Therefore, because a value is always defined, the $DCOC option acts the same as the $DCOU option.

Operands

Specify one of the following operands:

RECOVERY
The RECOVERY option should be specified.

SPEED
The SPEED option should be specified.

Default

None.

Environments

ALLOCATE

IDCAMS equivalent

```
DEFINE CLUSTER (RECOVERY ...)
DEFINE CLUSTER (SPEED ...)
```

Data class equivalent

Initial Load.

Related selection criteria

ILOAD

Example

In the following example, the initial load option to be specified is the SPEED option:

```
SET_ILOAD=S
```

SET_KEY_LEN

(Key-sequenced VSAM data sets only.) The SET_KEY_LEN keyword specifies the key length of a key-sequenced VSAM data set.

Syntax

SET_KEY_LEN=*nnn* | $DCOC | $DCOU

Where:

nnn is the length of a key in the range of 1 - 255.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD KEYLEN=6,...
```

IDCAMS equivalent

```
DEFINE CLUSTER(KEYS(8 ..) ...)
```

Data class equivalent

Keylen

Related selection criteria

KEY_LEN

Example

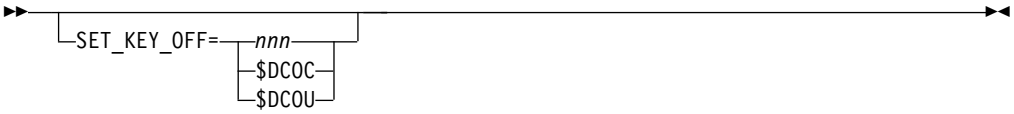
In the following example, the specified key length is 8:

```
SET_KEY_L=8
```

SET_KEY_OFF

(Key-sequenced VSAM data sets only.) The SET_KEY_OFF keyword specifies the offset of the key in a key-sequenced VSAM data set.

Syntax



SET_KEY_OFF=*nnnnn* | $DCOC | $DCOU

Where:

nnnnn is the offset of the key (0 - 32767) within a data record.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD KEYOFF=0,...
```

IDCAMS equivalent

```
DEFINE CLUSTER(KEYS(... 12) ...)
```

Data class equivalent

Keyoff.

Related selection criteria

KEY_OFF

Example

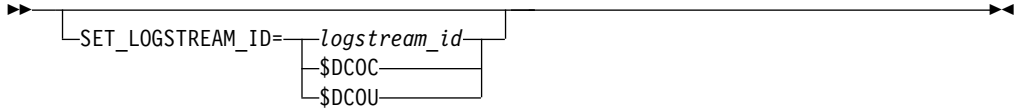
In the following example, the specified offset of the key is 12:

```
SET_KEY_OFF=12
```

SET_LOGSTREAM_ID

(SMS-managed VSAM data sets only.) The SET_LOGSTREAM_ID keyword specifies the name of a logstream data set for an SMS-managed VSAM data set.

Syntax



SET_LOGSTREAM_ID=*logstream_id* | $DCOC | $DCOU

Where:

logstream_id is a string of 1 - 26 characters that name the logstream. Wildcard characters are not permitted.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD LGSTREAM=PAY.STREAM
```

IDCAMS equivalent

```
DEFINE CLUSTER(LOGSTREAMID(PAY.STREAM) ...)
```

Data class equivalent

Logstream ID

Related selection criteria

LOGSTREAM_ID

Example

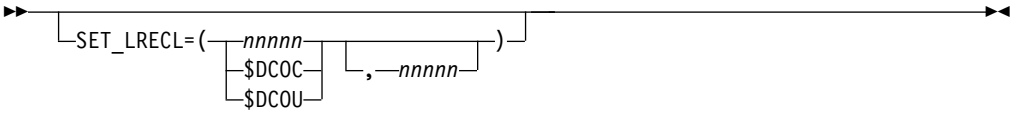
In the following example, the name of the logstream data set is PAY.STREAM:

```
SET_LOGSTREAM_ID=PAY.STREAM
```

SET_LRECL

The SET_LRECL keyword specifies the record length of a data set.

Syntax



SET_LRECL=(*nnnnn* | $DCOC | $DCOU,*nnnnn*)

Where:

nnnnn is an integer (1 - 32767) that defines the length of a non-VSAM record or the average and maximum record sizes of a VSAM data set.

Notes:

- For non-VSAM data sets, the second length is ignored.
- If $DCOC or $DCOU is specified for the first operand, the second operand becomes irrelevant.
- The second LRECL set of parameters in the SET_LRECL function pertain only to IDCAMS-based VSAM DEFINEs. Therefore, they are ignored for JCL-based definition of VSAM data sets.
- The $DCOC and $DCOU keywords are only valid for a SPACEUNITS value of AVGRECU, AVGRECK, or AVGRECM.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD LRECL=80,....
```

IDCAMS equivalent

```
DEFINE CL (RECORDSIZE(80 120) ...)
```

Data class equivalent

Lrecl

Related selection criteria

LRECL

Example

```
SET_LRECL=(80,,120)
```

SET_MGMTCLAS

(SMS-managed data sets only.) The SET_MGMTCLAS keyword specifies the SMS management class to be assigned to an SMS data set.

Syntax

SET_MGMTCLAS=*mgmtclas* | $NULL

Where:

mgmtclas is the 1 - 8 character name that is assigned to the SMS management class. Wildcard characters are not permitted.

Note: The SET_MGMTCLAS keyword requires additional calls to SMS Configuration Services and its presence in the RULEDEFS member can cause a measurable increase in CPU time for all tasks in the system, especially tasks that perform large numbers of allocations. Therefore, carefully consider the specification of the SET_MGMTCLAS keyword in your rule definitions member.

Default

None.

Environments

ACS

JCL equivalent

```
DD01 DD MGMTCLAS=STANDARD
```

IDCAMS equivalent

```
DEFINE CLUSTER(MANAGEMENTCLASS(LONGTERM) ...)
```

Related selection criteria

MGMTCLAS

Example

In the following example, the SMS management class to be assigned is TSO:

```
SET_MGMTCLAS=TSO
```

SET_NDISP

The SET_NDISP keyword specifies the normal termination disposition for a data set.

Syntax

SET_NDISP=CATLG | DELETE | KEEP | PASS | UNCATLG

Operands

Specify one of the following operands:

CATLG
The specified or defaulted normal disposition is DISP=(,CATLG).

DELETE
The specified or defaulted normal disposition is DISP=(,DELETE).

KEEP
The specified or defaulted normal disposition is DISP=(,KEEP).

PASS
The specified or defaulted normal disposition is DISP=(,PASS).

UNCATLG
The specified or defaulted normal disposition is DISP=(,UNCATLG).

Notes:

1. The SET_NDISP keyword does not apply to VSAM IDCAMS DEFINE statements.
2. The system takes the following defaults with the normal termination disposition parameter:
   - If you omit the normal termination disposition, the default is DELETE for a NEW data set and KEEP for an existing data set.
   - If you omit the DISP parameter, the default disposition is DELETE for the normal termination disposition.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD DISP=(,DELETE,)
```

Related selection criteria

NDISP

Example

In the following example, the specified or defaulted normal disposition is DISP=(,CATLG):

```
SET_NDISP=CATLG
```

SET_NUMVOLS

(This keyword applies to VSAM DEFINES only.) The SET_NUMVOLS keyword specifies the number of volumes to define at the cluster level for a VSAM DEFINE.

Syntax

SET_NUMVOLS=*nn* | $DCOC | $DCOU

Where:

nnn is an integer (1 - 59) that specifies the number of volumes.

Notes:

- If you specify SET_NUMVOLS=$DCOU and the DATACLAS Volume Count value is null, the override is not performed.
- For non-SMS managed data sets, SET_NUMVOLS is only effective if AVS is set to ENABLE.
- For non-SMS managed data sets, IDCAMS requires the specification of the VOLUMES parameter when a data set is defined. When you use SET_NUMVOLS, this parameter must still be defined, but any VOLSER can be specified because it is overridden by AVS processing.

Default

None.

Environments

ALLOCATE

IDCAMS equivalent

```
DEFINE CLUSTER(VOLUMES(VOL001 VOL002) ...)
DEFINE CLUSTER(VOLUMES(* * * *) ...)
```

Data class equivalent

Volume Count

Related selection criteria

NUMVOLS

Example

In the following example, the number of volumes to be defined is 2:

```
SET_NUMVOLS=2
```

SET_NUMVOLS_DATA

(Applies to VSAM DEFINES only.) The SET_NUMVOLS_DATA keyword specifies the number of volumes to be defined for the data component of a VSAM cluster.

Syntax

SET_NUMVOLS_DATA=*nn*

Where:

nn is an integer (1 - 59) that specifies the number of volumes.

Default

None.

Environments

ALLOCATE

IDCAMS equivalent

```
DEFINE CLUSTER(...) DATA(VOLUMES(VOL001 VOL002) ...)
DEFINE CLUSTER(...) DATA(VOLUMES(* * * *) ...)
```

Related selection criteria

NUMVOLS_DATA

Example

In the following example, the number of volumes to be defined is 2:

```
SET_NUMVOLS_DATA=2
```

SET_NUMVOLS_INDEX

(VSAM DEFINES only.) The SET_NUMVOLS_INDEX keyword specifies the number of volumes to be defined for the index component of a VSAM cluster. For instances other than VSAM DEFINES, use SET_VOLCT instead.

Syntax

SET_NUMVOLS_INDEX=*nn*

Where:

nn is an integer (1 - 59) that specifies the number of volumes.

Default

None.

Environments

ALLOCATE

IDCAMS equivalent

```
DEFINE CLUSTER(...) INDEX(VOLUMES(VOL001 VOL002) ...)
DEFINE CLUSTER(...) INDEX(VOLUMES(* * * *) ...)
```

Related selection criteria

NUMVOLS_INDEX

Example

In the following example, the number of volumes to be defined is 2:

```
SET_NUMVOLS_INDEX=2
```

SET_PQTY

The SET_PQTY keyword specifies the primary quantity value that is assigned to the data set. For VSAM data sets, it defines the primary quantity at the cluster level.

Syntax

SET_PQTY=*nnnnnnnn* | $DCOC | $DCOU | $NULL

Where:

nnnnnnnn is an integer (0 - 16777215) that specifies the primary quantity for a data set. For non-VSAM new data sets, *nnnnnnnn* is the actual amount requested. For VSAM data sets, *nnnnnnnn* is the requested primary quantity (the value is represented in space allocation units).

Note: The $DCOC and $DCOU keywords are only valid for a SPACEUNITS value of AVGRECU, AVGRECK, or AVGRECM.

See “Space parameter considerations” on page 264 for information about specifying space-related keywords.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD SPACE=(...,(1))
```

IDCAMS equivalent

```
DEFINE CLUSTER(CYLINDERS(1) ....)
DEFINE CLUSTER(TRACKS(1) ....)
DEFINE CLUSTER(RECORDS(1000) ....)
DEFINE CLUSTER(MEGABYTES(2) ....)
DEFINE CLUSTER(KILOBYTES(2048) ...)
```

Data class equivalent

Space Primary

Related selection criteria

PQTY

Example

In the following example, the primary quantity value is 2048:

```
SET_PQTY=2048
```

SET_PQTY_DATA

(VSAM data sets only.) The SET_PQTY_DATA keyword specifies the primary quantity value to be assigned to the data component of a VSAM data set.

Syntax

SET_PQTY_DATA=*nnnnnnnn* | $NULL

Where:

nnnnnnnn is an integer (0 - 16777215) that specifies the requested primary allocation quantity for the data component in the requested allocation units.

Default

None.

Environments

ALLOCATE

IDCAMS equivalent

```
DEFINE CLUSTER(...) DATA(CYLINDERS(1) ....)
DEFINE CLUSTER(...) DATA(TRACKS(1) ....)
DEFINE CLUSTER(...) DATA(RECORDS(1000) ....)
DEFINE CLUSTER(...) DATA(MEGABYTES(2) ....)
DEFINE CLUSTER(...) DATA(KILOBYTES(2048) ...)
```

Related selection criteria

PQTY_DATA

Example

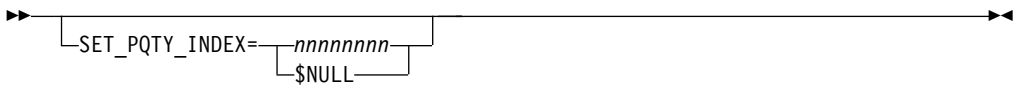
In the following example, the primary quantity value is 2048:

```
SET_PQTY_DATA=2048
```

SET_PQTY_INDEX

(VSAM data sets only.) The SET_PQTY_INDEX keyword specifies the primary quantity value to be assigned to the index component of a VSAM data set.

Syntax



SET_PQTY_INDEX=*nnnnnnnn* | $NULL

Where:

nnnnnnnn is an integer (0 - 16777215) that specifies the requested primary allocation quantity for the index component in the requested allocation units.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DEFINE CLUSTER(...) INDEX(CYLINDERS(1) ....)
DEFINE CLUSTER(...) INDEX(TRACKS(1) ....)
DEFINE CLUSTER(...) INDEX(RECORDS(1000) ....)
DEFINE CLUSTER(...) INDEX(MEGABYTES(2) ....)
DEFINE CLUSTER(...) INDEX(KILOBYTES(2048) ...)
```

Related selection criteria

PQTY_INDEX

Example

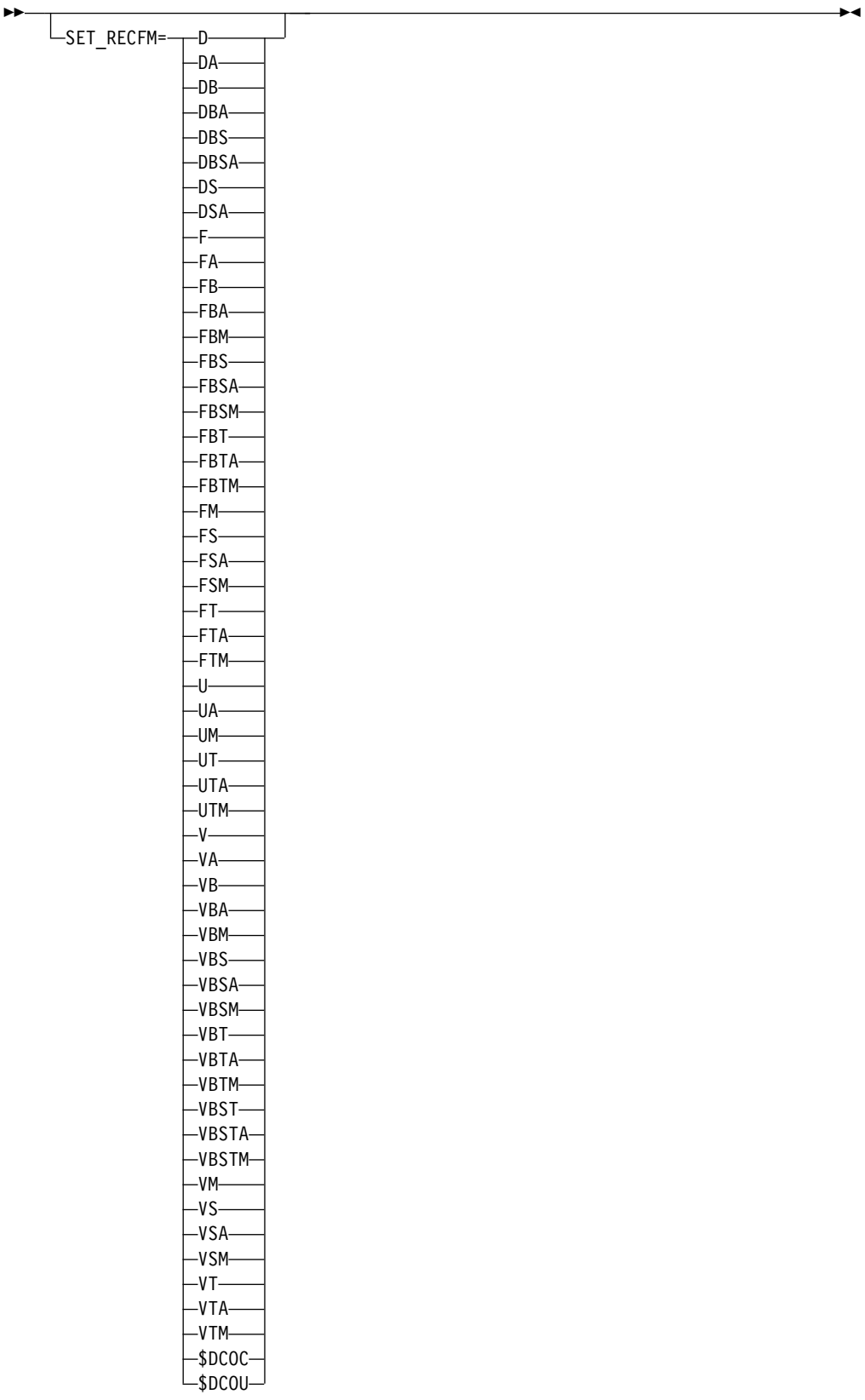
In the following example, the primary quantity value is 2048:

```
SET_PQTY_INDEX=2048
```

SET_RECFM

(Non-VSAM data sets only.) The SET_RECFM keyword specifies the record format for a non-VSAM data set.

Syntax



SET_RECFM=D | DA | DB | DBA | DBS | DBSA | DS | DSA | F | FA | FB | FBA | FBM | FBS | FBSA | FBSM | FBT | FBTA | FBTM | FM | FT | FTA | FTM | FS | FSA | FSM | U | UA | UM | UT | UTA | UTM | V | VA | VB | VBA | VBM | VM | VBS | VBSA | VBSM |

VBT | VBTA | VBTM | VBST | VBSTA | VBSTM | VS | VSA | VSM | VT | VTA | VTM | $DCOC | $DCOU

Operands

D ASCII variable-length.

DA ASCII variable-length (ANSI format).

DB ASCII variable-length (blocked).

DBA
ASCII variable-length (blocked, ANSI format).

DBS
ASCII variable-length (blocked, spanned).

DBSA
ASCII variable-length (blocked, spanned, ANSI format).

DS ASCII variable-length (spanned).

DSA
ASCII variable-length (spanned, ANSI format).

F Fixed length.

FA Fixed length (ANSI format).

FB Fixed length (blocked).

FBA
Fixed length (blocked, ANSI format).

FBM
Fixed length (blocked, machine code format).

FBS
Fixed length (blocked, standard blocks).

FBSA
Fixed length (blocked, standard blocks, ANSI format).

FBSM
Fixed length (blocked, standard blocks, machine code format).

FBT
Fixed length (blocked, track overflow).

FBTA
Fixed length (blocked, track overflow, ANSI format).

FBTM
Fixed length (blocked, track overflow, machine code format).

FM Machine code format.

FS Fixed length (standard blocks).

FSA
Fixed length (standard blocks, ANSI format).

FSM
Fixed length (standard blocks, machine code format).

FT Fixed length (track overflow).

FTA
Fixed length (track overflow, ANSI format).

FTM
Fixed length (track overflow, machine code format).

U Undefined-length.

UA Undefined-length (ANSI format).

UM Undefined-length (machine code format).

UT Undefined-length (track overflow).

UTA
Undefined-length (track overflow, ANSI format).

UTM
Undefined-length (track overflow, machine code format).

V Variable length.

VA
Variable length (ANSI format).

VB Variable length (blocked).

VBA
Variable length (blocked, ANSI format).

VBM
Variable length (blocked, machine code format).

VBS
Variable length (blocked, spanned).

VBSA
Variable length (blocked, spanned, ANSI format).

VBSM
Variable length (blocked, spanned, machine code format).

VBST
Variable length (blocked, spanned, track overflow).

VBSTA
Variable length (blocked, spanned, track overflow, ANSI format).

VBSTM
Variable length (blocked, spanned, track overflow, machine code format).

VBT
Variable length (blocked, track overflow).

VBTA
Variable length (blocked, track overflow, ANSI format).

VBTM
Variable length (blocked, track overflow, machine code format).

VM
Variable length (machine code format).

VS
Variable length (spanned).

VSA
Variable length (spanned, ANSI format).

VSM
Variable length (spanned, machine code format).

VT
Variable length (track overflow).

VTA
Variable length (track overflow, ANSI format).

VTM
Variable length (track overflow, machine code format).

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD RECFM=FB
```

Data class equivalent

Recfm

Related selection criteria

RECFM

Example

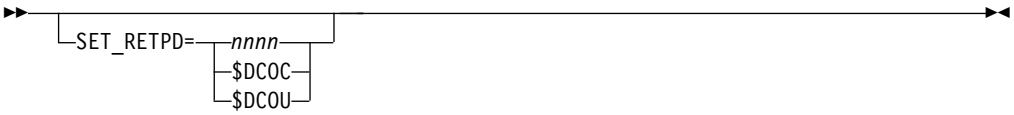
In the following example, the record format is fixed length (blocked):

```
SET_RECFM=FB
```

SET_RETPD

The SET_RETPD keyword specifies the retention period for a data set.

Syntax



SET_RETPD=*nnnn* | $DCOC | $DCOU

Where:

nnnn is an integer (0 - 9999) that specifies the number of days to retain the data set.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD RETPD=100
```

IDCAMS equivalent

```
DEFINE CLUSTER(FOR(200) ...)
```

Data class equivalent

Retpd or Expdt.

Related selection criteria

RETPD

Example

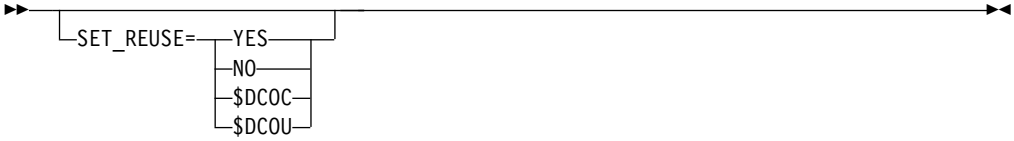
In the following example, the retention period is 100 days:

```
SET_RETPD=100
```

SET_REUSE

(VSAM data sets only.) The SET_REUSE keyword specifies whether the VSAM REUSE or NOREUSE option should be specified.

Syntax



SET_REUSE=YES | NO | $DCOC | $DCOU

Note: For this VSAM option, the ISMF Data Class Define panel always requires a value of Y or N. Therefore, because a value is always defined, the $DCOC option acts the same as the $DCOU option.

Operands

YES
REUSE should be specified.

NO NOREUSE should be specified.

Default

None.

Environments

ALLOCATE

IDCAMS equivalent

```
DEFINE CL (REUSE ....)
DEFINE CL (NOREUSE....)
```

Data class equivalent

Reuse

Related selection criteria

REUSE

Example

In the following example, the VSAM REUSE option should be specified:

```
SET_REUSE=YES
```

SET_SHROPT_XRGN

(VSAM data sets only.) The SET_SHROPT_XGRN keyword specifies the shareoptions for cross-region sharing.

Syntax

SET_SHROPT_XRGN=*n* | $DCOC | $DCOU

Where:

n is an integer (1 - 4) that specifies the VSAM cross-region shareoption.

Default

None.

Environments

ALLOCATE

IDCAMS equivalent

```
DEFINE CL (SHAREOPTIONS(1) ...)
```

Data class equivalent

Shareoptions Xregion

Related selection criteria

SHROPT_XRGN

Example

In the following example, 1 is the specified VSAM cross-region shareoption:

```
SET_SHROPT_XRGN=1
```

SET_SHROPT_XSYS

(VSAM data sets only.) The SET_SHROPT_XSYS keyword specifies the shareoptions for cross-system sharing.

Syntax

SET_SHROPT_XSYS=*n* | $DCOC | $DCOU

Where:

n is an integer (3 or 4) that specifies the VSAM cross-system shareoption.

Default

None.

Environments

ALLOCATE

IDCAMS equivalent

```
DEFINE CL (SHAREOPTIONS(. 3) ...)
```

Data class equivalent

Shareoptions Xsystem

Related selection criteria

SHROPT_XSYS

Example

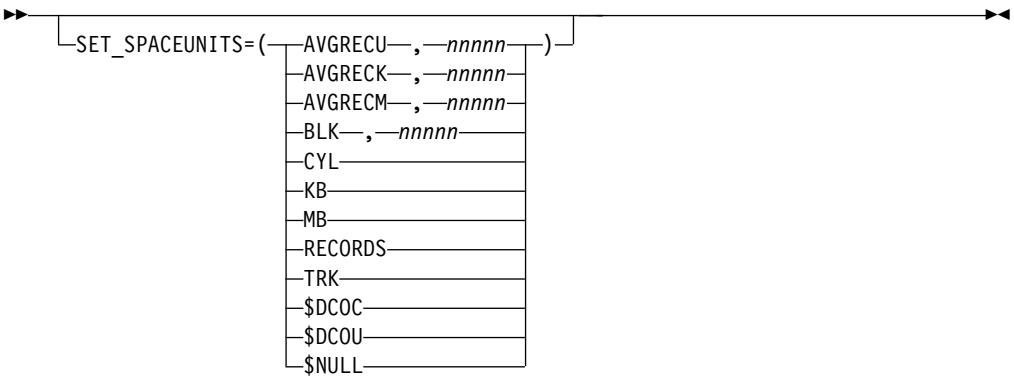
In the following example, 3 is the specified VSAM cross-system shareoption:

```
SET_SHROPT_XSYS=3
```

SET_SPACEUNITS

The SET_SPACEUNITS keyword specifies the allocation unit type for a data set.

Syntax



SET_SPACEUNITS=(AVGRECU | AVGRECK | AVGRECM | BLK | CYL | KB | MB | TRK | RECORDS | $DCOC | $DCOU | $NULL, *nnnnn*)

Operands

AVGRECU
Space is allocated in records with AVGREC=U specified.

AVGRECK
Space is allocated in records with AVGREC=K specified.

AVGRECM
Space is allocated in records with AVGREC=M specified.

Note: You can use the AVGRECU, AVGRECK, and AVGRECM operands with a VSAM DEFINE and it will function in an identical manner as the processing that occurs when AVGREC is specified in the Data Class.

BLK
Space is allocated in blocks.

nnnnn
An integer (0 - 65535) that specifies the block size or record size when allocations are made with the allocation space units of BLK or AVGRECU, AVGRECK, or AVGRECM respectively.

CYL
Space is allocated in cylinders.

KB Space is allocated in kilobytes.

Note: This value applies to IDCAMS DEFINEs only.

MB Space is allocated in megabytes.

Note: This value applies to IDCAMS DEFINEs only.

RECORDS
Space is allocated using the IDCAMS DEFINE CLUSTER RECORDS keyword.

TRK
Space is allocated in tracks.

See "Space parameter considerations" on page 264 for information about specifying space-related keywords.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD SPACE=(8000,(2000,200))
```

IDCAMS equivalent

```
DEFINE CLUSTER(CYLINDERS(...) ....)
DEFINE CLUSTER(TRACKS(...) ....)
DEFINE CLUSTER(RECORDS(...) ....)
DEFINE CLUSTER(MEGABYTES(...) ....)
DEFINE CLUSTER(KILOBYTES(...) ...)
```

Data class equivalent

Space Avgrec

Related selection criteria

SPACEUNITS

Example

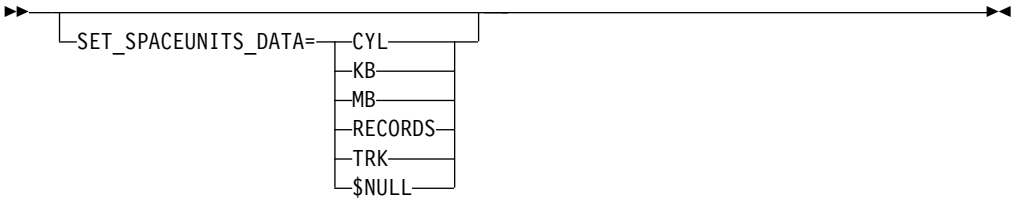
In the following example, space is allocated in cylinders:

```
SET_SPACEU=CYL
```

SET_SPACEUNITS_DATA

(VSAM data sets only.) The SET_SPACEUNITS_DATA keyword specifies the allocation units for the data component of a VSAM data set.

Syntax



SET_SPACEUNITS_DATA=CYL | KB | MB | RECORDS | TRK | $NULL

Operands

CYL
Space is allocated in cylinders.

KB Space is allocated in kilobytes.

MB Space is allocated in megabytes.

RECORDS
Space is allocated in records.

TRK
Space is allocated in tracks.

Default

None.

Environments

ALLOCATE

IDCAMS equivalent

```
DEFINE CLUSTER(...) DATA(CYLINDERS(...) ....)
DEFINE CLUSTER(...) DATA(TRACKS(...) ....)
DEFINE CLUSTER(...) DATA(RECORDS(...) ....)
DEFINE CLUSTER(...) DATA(MEGABYTES(...) ....)
DEFINE CLUSTER(...) DATA(KILOBYTES(...) ...)
```

Related selection criteria

SPACEUNITS_DATA

Example

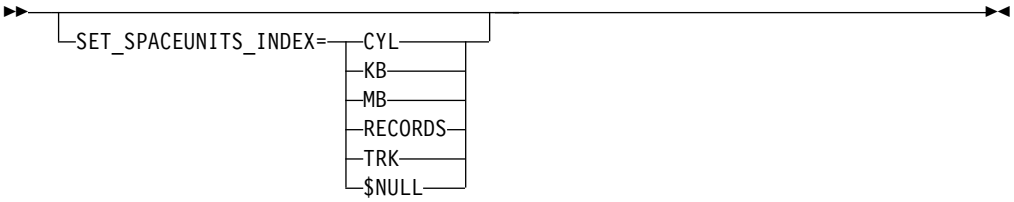
In the following example, space is allocated in kilobytes:

```
SET_SPACEUNITS_D=KB
```

SET_SPACEUNITS_INDEX

(VSAM data sets only.) The SET_SPACEUNITS_INDEX keyword specifies the allocation units for the index component of a VSAM data set.

Syntax



SET_SPACEUNITS_INDEX=CYL | KB | MB | RECORDS | TRK | $NULL

Operands

CYL
Space is allocated in cylinders.

KB Space is allocated in kilobytes.

MB Space is allocated in megabytes.

RECORDS
Space is allocated in records.

TRK
Space is allocated in tracks.

Default

None.

Environments

ALLOCATE

IDCAMS equivalent

```
DEFINE CLUSTER(...) INDEX(CYLINDERS(...) ....)
DEFINE CLUSTER(...) INDEX(TRACKS(...) ....)
DEFINE CLUSTER(...) INDEX(RECORDS(...) ....)
DEFINE CLUSTER(...) INDEX(MEGABYTES(...) ....)
DEFINE CLUSTER(...) INDEX(KILOBYTES(...) ...)
```

Related selection criteria

SPACEUNITS_INDEX

Example

In the following example, space is allocated in tracks:

```
SET_SPACEUNITS_INDEX=TRK
```

SET_SPANNED

(VSAM data sets only.) The SET_SPANNED keyword specifies whether the SPANNED option should be defined for a VSAM data set.

Syntax

SET_SPANNED=YES | NO | $DCOC | $DCOU

Operands

YES
The SPANNED option should be defined.

NO The NOSPANNED option should be defined.

Default

None.

Environments

ALLOCATE

IDCAMS equivalent

```
DEFINE CLUSTER (SPANNED ...)
DEFINE CLUSTER (NOSPANNED ...)
```

Data class equivalent

Spanned / Nonspanned

Related selection criteria

SPANNED

Example

In the following example, the SPANNED option should be defined:

```
SET_SPAN=Y
```

SET_SQTY

The SET_SQTY keyword specifies the secondary quantity value to be assigned to the data set. For VSAM data sets, the SET_SQTY keyword specifies the secondary quantity at the cluster level.

Syntax

SET_SQTY=*nnnnnnnn* | $DCOC | $DCOU | $NULL

Where:

nnnnnnnn is an integer (0 - 16777215) that specifies the requested secondary quantity for a data set in the requested allocation units.

| **Note:** The $DCOC and $DCOU keywords are only valid for a SPACEUNITS value
| of AVGRECU, AVGRECK, or AVGRECM.

See “Space parameter considerations” on page 264 for information about specifying space-related keywords.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD SPACE=(...,(,2))
```

IDCAMS equivalent

```
DEFINE CLUSTER(CYLINDERS(... 1) ....)
DEFINE CLUSTER(TRACKS(... 1) ....)
DEFINE CLUSTER(RECORDS(... 1000) ....)
DEFINE CLUSTER(MEGABYTES(... 2) ....)
DEFINE CLUSTER(KILOBYTES(... 2048) ...)
```

Data class equivalent

Space Secondary

Related selection criteria

SQTY

Example

In the following example, the secondary quantity value to be assigned is 2048:

```
SET_SQTY=2048
```

SET_SQTY_DATA

(VSAM data sets only.) The SET_SQTY_DATA keyword specifies the secondary quantity value to be assigned to the data component of a VSAM data set.

Syntax

SET_SQTY_DATA=*nnnnnnnn* | $NULL

Where:

nnnnnnnn is an integer (0 - 16777215) that specifies the requested secondary allocation quantity for the data component in the requested allocation units.

Default

None.

Environments

ALLOCATE

IDCAMS equivalent

```
DEFINE CLUSTER(...) DATA(CYLINDERS(... 1) ....)
DEFINE CLUSTER(...) DATA(TRACKS(... 1) ....)
DEFINE CLUSTER(...) DATA(RECORDS(... 1000) ....)
DEFINE CLUSTER(...) DATA(MEGABYTES(... 2) ....)
DEFINE CLUSTER(...) DATA(KILOBYTES(... 2048) ...)
```

Related selection criteria

SQTY_DATA

Example

In the following example, the secondary quantity value to be assigned is 2048:

```
SET_SQTY_DATA=2048
```

SET_SQTY_INDEX

(VSAM data sets only.) The SET_SQTY_INDEX keyword specifies the secondary quantity value to be assigned to the index component of a VSAM data set.

Syntax

SET_SQTY_INDEX=*nnnnnnnn* | $NULL

Where:

nnnnnnnn is an integer (0 - 16777215) that specifies the requested secondary allocation quantity for the index component in the requested allocation units.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DEFINE CL ... INDEX(TRACKS(1) ...)
```

IDCAMS equivalent

```
DEFINE CLUSTER(...) INDEX(CYLINDERS(... 1) ....)
DEFINE CLUSTER(...) INDEX(TRACKS(... 1) ....)
DEFINE CLUSTER(...) INDEX(RECORDS(... 1000) ....)
DEFINE CLUSTER(...) INDEX(MEGABYTES(... 2) ....)
DEFINE CLUSTER(...) INDEX(KILOBYTES(... 2048) ...)
```

Related selection criteria

SQTY_INDEX

Example

In the following example, the secondary quantity value to be assigned is 2048:

```
SET_SQTY_INDEX=2048
```

SET_STORCLAS

The SET_STORCLAS keyword specifies an SMS storage class to be assigned to an SMS data set.

Syntax

SET_STORCLAS=*storclas* | $NULL

Where:

storclas is the 1 - 8 character string that specifies the SMS storage class. Wildcard characters are not permitted.

Notes:

1. If you remove the storage class, you effectively make the data set non SMS-managed. To remove the storage class that is assigned to a data set by an SMS ACS routine, specify SET_STORCLAS=$NULL.
2. If you add a storage class to a data set that originally did not have one assigned to it, it causes the data set to become SMS-managed.
3. SET_STORCLAS requires additional calls to SMS Configuration Services and its presence in the RULEDEFS member can cause a measurable increase in CPU time for all of the tasks in the system, especially tasks that perform many allocations. Therefore, carefully consider the specification of SET_STORCLAS in your rule definitions member.

Default

None.

Environments

ACS

JCL equivalent

```
DD01 DD STORCLAS=PAYRCLAS
```

IDCAMS equivalent

```
DEFINE CLUSTER(STORAGECLASS(PAYRCLAS) ...)
```

Related selection criteria

STORCLAS

Example

In the following example, the SMS storage class to be assigned is PAYRCLAS:

```
SET_STORCLAS=PAYRCLAS
```

SET_STORGRP

The SET_STORGRP keyword specifies an SMS storage group that is assigned to the data set.

Syntax

SET_STORGRP=*storgrp* | $NULL

Where:

storgrp is a 1 - 8 character string that specifies the SMS storage group.

Note: The SET_STORGRP keyword requires additional calls to SMS Configuration Services and its presence in the RULEDEFS member can cause a measurable increase in CPU time for all tasks in the system, especially tasks that perform large numbers of allocations. Therefore, carefully consider the specification of the SET_STORGRP keyword in your rule definitions member.

Default

None.

Environments

ACS

Related selection criteria

STORGR.

Example

In the following example, the SMS storage group is STGLARGE:

```
SET_STORGRP=STGLARGE
```

SET_TAPE_COMP

(Tape data sets only.) The SET_TAPE_COMP keyword specifies the data compaction setting for a tape data set.

Syntax

SET_TAPE_COMP=YES | NO | $DCOC | $DCOU

Operands

YES
Specify TRTCH=COMP.

NO Do not specify TRTCH=NOCOMP.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD DCB=('TRTCH=COMP')
```

Data class equivalent

Compaction

Related selection criteria

TAPE_COMP

Example

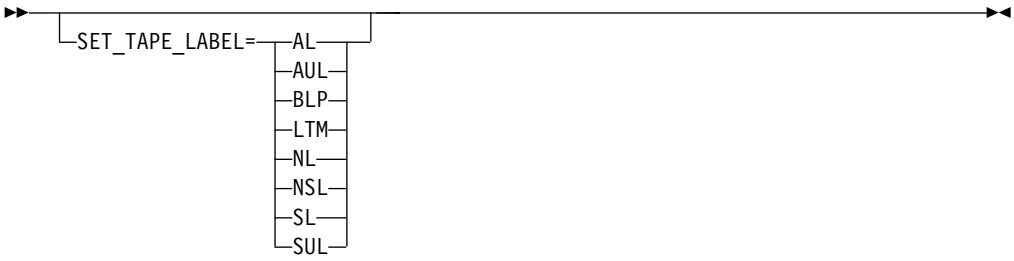
In the following example, TRTCH=COMP should be specified:

```
SET_TAPE_C=Y
```

SET_TAPE_LABEL

(Tape data sets only.) The SET_TAPE_LABEL keyword specifies the type of tape label that is used for a data set.

Syntax



SET_TAPE_LABEL=AL | AUL | BLP | LTM | NL | NSL | SL | SUL

Operands

AL ISO/ANSI Version 1 or ISO/ANSI/FIPS Version 3 labels.

AUL
User labels and ISO/ANSI Version 1 or ISO/ANSI/FIPS Version 3 labels.

BLP
Bypass label processing.

LTM
Leading tapemarks.

NL Unlabeled tapes.

NSL
Non-standard labels.

SL Standard labels.

SUL
Standard and user labels.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD LABEL=(,SL),...
```

Related selection criteria

TAPE_LABEL

Example

In the following example, standard labels are used:

```
SET_TAPE_L=SL
```

SET_UNIT_AFFINITY

The SET_UNIT_AFFINITY keyword specifies whether UNIT affinity should be defined for a data set.

Syntax

SET_UNIT_AFFINTY=YES|NO

Operands

Specify one of the following operands:

YES
UNIT affinity should be specified (allow unit affinity if it is specified).

NO UNIT affinity should not be specified (remove unit affinity if it is specified).

Note: The SET_UNIT_AFFINITY keyword does not apply to VSAM DEFINEs or SMS data sets.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD UNIT=AFF=DD02
```

Related selection criteria

UNIT_AFFINITY

Example

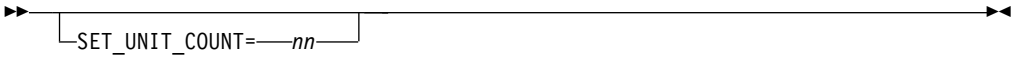
In the following example, UNIT affinity should be specified:

```
SET_UNIT_A=Y
```

SET_UNIT_COUNT

The SET_UNIT_COUNT keyword specifies the number of units that are assigned to the data set.

Syntax



SET_UNIT_COUNT=*nn*

Where:

nn specifies the number of units (1 - 59) assigned to the data set.

Note: Do not use SET_UNIT_COUNT for VSAM DEFINEs. SET_UNIT_COUNT is valid for JCL or dynamically allocated data sets (including SMS and non-SMS VSAM and non-VSAM data sets). For IDCAMS DEFINEs, use the SET_NUMVOLS keyword instead.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD UNIT=(SYSDA,3)
```

Related selection criteria

UNIT_COUNT

Example

In the following example, the number of units that are assigned to the data set is 3:

```
SET_UNIT_C=3
```

SET_UNIT_NAME

The SET_UNIT_NAME keyword specifies the generic or esoteric unit name to be assigned.

Syntax

SET_UNIT_NAME=*uuuuuuuu* | $NULL

Note: SET_UNIT_NAME is only applicable to non-SMS data sets. The SET_UNIT_NAME keyword does not apply to VSAM DEFINEs or SMS data sets.

Where:

uuuuuuuu is the 1 - 8 character unit name. Wildcard characters are not permitted.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD UNIT=SYSALLDA
```

Related selection criteria

UNIT_NAME

Examples

In the following example, the unit name to be assigned is SYSALLDA:

```
SET_UNIT_NAME=SYSALLDA
```

In the following example, when the specified retention period is fewer than five days, the unit name is set to TEMPTAPE.

```
INCLUDE
     RETPD=(5,LT)
     SET_UNIT_NAME=TEMPTAPE
```

SET_VOLCT

The SET_VOLCT keyword specifies the number of volumes that are defined to the data set.

Syntax

SET_VOLCT=*nnn* | $DCOC | $DCOU

Notes:

- For DASD data sets, the maximum number of volumes is limited to 59. For tape data sets, the maximum number of volumes is limited to 255.
- Do not use SET_VOLCT for VSAM DEFINEs. SET_VOLCT is valid for JCL or dynamically allocated data sets (including SMS and non-SMS VSAM and non-VSAM data sets). For IDCAMS DEFINEs, use the SET_NUMVOLS keyword instead.

Where:

nnn is an integer (1 - 255) that specifies the volume count.

Default

None.

Environments

ALLOCATE

JCL equivalent

```
DD01 DD VOL=(,,,4,SER=VOL001)
```

Data class equivalent

Volume Count

Related selection criteria

VOLCT

Example

In the following example, the number of volumes that are defined is 4:

```
SET_VOLCT=4
```

SET_VSAM_SEPARATE

(VSAM data sets only.) SET_VSAM_SEPARATE specifies whether the index and data components of a VSAM data set should be on separate volumes.

Syntax

SET_VSAM_SEPARATE=YES|NO

Operands

Specify one of the following operands:

YES
Index and data components should be on separate volumes.

NO Index and data components should *not* be on separate volumes.

Note: For non-SMS VSAM data sets, to separate the index and the data components, Advanced Allocation Management must have a potential list of volumes to choose from. Therefore, you can specify SET_VSAM_SEPARATE only if you are also using the advanced volume selection (AVS) function.

Default

None.

Environments

ALLOCATE

IDCAMS equivalent

```
DEFINE CL (...) DATA(VOL(VOL001) ...) INDEX(VOL(VOL002) ...)
```

Related selection criteria

VSAM_SEPARATE

Example

In the following example, index and data components should be on separate volumes:

```
SET_VSAM_SEP=Y
```

Selection criteria keywords

Use selection criteria keywords with the INCLUDE or EXCLUDE keywords to specify the items to include or exclude from Advanced Allocation Management processing.

Note the following items in the syntax descriptions:

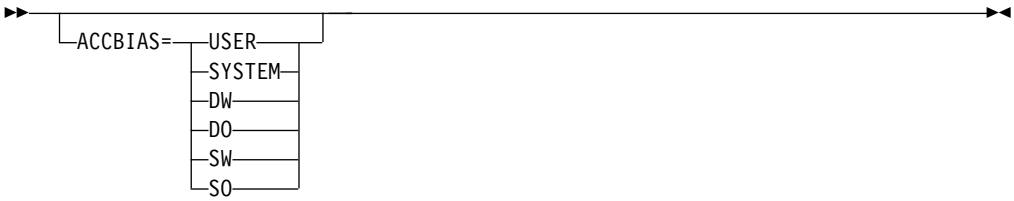
1. The underlined portion of the keyword indicates the minimally acceptable abbreviation.
2. $NULL indicates that the particular selection criterion was not specified.
3. *op* represents the comparison operands (EQ, NE, LT, GT, LE, and GE).

4. The default that is listed is the default for the Advanced Allocation Management keyword, *not* the default that is assigned to the particular attribute by the operating system, unless otherwise specifically noted.
5. Where applicable, the JCL, IDCAMS, and Data, Management, and Storage Class equivalents are listed. For the Data, Management, or Storage Class equivalents, the field name from the appropriate ISMF panel that is used to define the class is listed.
6. *user_var* indicates a user-defined variable.
7. The use of numeric suffixes is permitted.

ACCBIAS

(VSAM only.) The ACCBIAS keyword specifies the assigned VSAM System Managed Buffering specification.

Syntax



ACCBIAS=USER | SYSTEM | DW | DO | SO | SW

Operands

Specify one of the following operands:

USER
Bypass system-managed buffering.

SYSTEM
System-managed buffering.

DW Direct weighted.

DO Sequential optimized.

SO Sequential optimized.

SW Sequential weighted.

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

JCL equivalent

```
DD01 DD AMP=('ACCBIAS=USER')
```

Data class equivalent

Record Access Bias

Related Enhanced Allocation Management function

SET_ACCBIAS

Example

```
ACCBIAS=DO
```

ADISP

The ADISP keyword specifies the abnormal termination disposition for a data set.

Syntax

ADISP=KEEP | CATLG | UNCATLG | DELETE | $NULL

Operands

Specify one of the following operands:

KEEP
The specified abnormal disposition is DISP=(,,KEEP).

CATLG
The specified abnormal disposition is DISP=(,,CATLG).

UNCATLG
The specified abnormal disposition is DISP=(,,UNCATLG).

DELETE
The specified abnormal disposition is DISP=(,,DELETE).

Note:

1. Wildcard characters are not permitted.
2. The ADISP keyword is not applicable to VSAM IDCAMS DEFINE statements.
3. The system takes the following defaults with the abnormal termination disposition parameter:
   - If you omit the abnormal termination disposition, the default is the disposition that is specified, or defaulted to, by the normal termination disposition. If the normal termination disposition is PASS, the default abnormal termination disposition is DELETE for a NEW data set (or KEEP for an existing data set).
   - If you omit the DISP parameter, the default disposition is DELETE for the abnormal termination disposition.

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
DD01 DD DISP=(,,DELETE)
```

Related Enhanced Allocation Management function

SET_ADISP

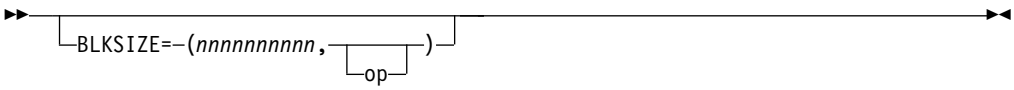
Example

```
ADISP=KEEP
```

BLKSIZE

(Non-VSAM only.) The BLKSIZE keyword specifies the defined block size for a data set.

Syntax



BLKSIZE=(*nnnnnnnnn*,op)

Where:

nnnnnnnnn is an integer (0 - 2147483647).

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
DD DD01 BLKSIZE=256000
```

Related Enhanced Allocation Management function

SET_BLKSIZE

Example

```
BLKSIZE=256000
```

BUFSPC

(VSAM only.) The BUFSPC keyword specifies the defined buffer space specification for a data set.

Syntax

BUFSPC=(*nnnnnnnnnn*,op)

Where:

nnnnnnnnnn is an integer (0 - 2147483647).

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

JCL equivalent

```
DD DD01 AMP=('BUFSP=128000')
```

IDCAMS equivalent

```
DEFINE CLUSTER(BUFFERSPACE(256000) ...)
```

Related Enhanced Allocation Management function

SET_BUFSPC

Example

```
BUFSPC=128000
```

CISZ

(VSAM only.) The CISZ keyword specifies the defined control interval size for a VSAM cluster.

Syntax

CISZ=(*nnnnn*,op)

Where:

nnnnn is an integer (0 - 32768).

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

AMS equivalent

```
DEFINE CLUSTER (CONTROLINTERVALSIZE(8192) ....)
```

Related Enhanced Allocation Management function

SET_CISZ

Example

```
CISZ=8192
```

CISZ_DATA

(VSAM only.) The CISZ_DATA keyword specifies the defined control interval size for the data component of a VSAM cluster.

Syntax

CISZ_DATA=(*nnnnn*,op)

Where:

nnnnn is an integer (0 - 32768).

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

AMS equivalent

```
DEFINE CLUSTER (....) DATA (CONTROLINTERVALSIZE(8192) ....)
```

Related Enhanced Allocation Management function

SET_CISZ_DATA

Example

```
CISZ_DATA=8192
```

CISZ_INDEX

(VSAM only.) The CISZ_INDEX keyword specifies the defined control interval size for the index component of a VSAM cluster.

Syntax

CISZ_INDEX=(*nnnnn*,op)

Where:

nnnnn is an integer (0 - 32768).

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

AMS equivalent

```
DEFINE CLUSTER (....) DATA(....) INDEX(CONTROLINTERVALSIZE(8192) ....)
```

Related Enhanced Allocation Management function

SET_CISZ_INDEX

Example

```
CISZ_INDEX=8192
```

CONTIGUOUS

(Non-VSAM disk data sets only.) The CONTIGUOUS keyword specifies whether the CONTIG option was specified when allocating a new data set.

Syntax

CONTIGUOUS=YES | NO

Operands

Specify one of the following operands:

YES
CONTIG was specified.

NO CONTIG was not specified.

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
DD01 DD SPACE=(CYL,(1,1),,CONTIG)
```

Related Enhanced Allocation Management function

SET_CONTIGUOUS

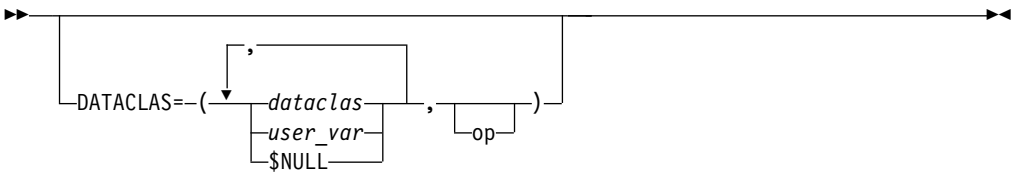
Example

```
CONTIG=Y
```

DATACLAS

The DATACLAS keyword specifies the SMS data class that is assigned to a data set.

Syntax



DATACLAS=(*dataclas* | *user_var*, *dataclas* | *user_var*,..., | $NULL,op)

Where:

dataclas is the 1 - 8 character SMS data class name. Wildcard characters are permitted.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

Note: During ACS processing, the value that is assigned by the user or by the system is used. In the other environments, the value that is assigned by Advanced Allocation Management (if specified) is used; otherwise, the value that is assigned by the user or by the system is used.

JCL equivalent

```
DD01 DD DATACLAS=SMALL
```

IDCAMS equivalent

```
DEFINE CLUSTER (NAME(...) DATACLASS(SMALL) ...)
```

Related Enhanced Allocation Management function

SET_DATACLAS

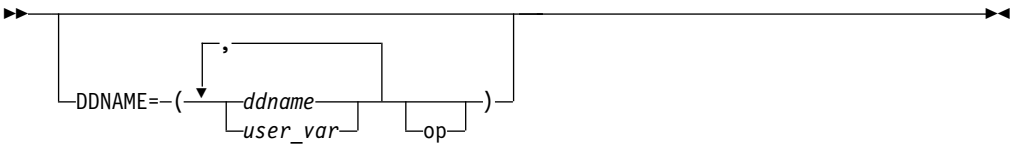
Examples

```
DATACLAS=LISTDATA
DATACLAS=&DATA_S
DATACLAS=(LISTDATA,SCRIPT%,EQ)
DATACLAS=($NULL,NE)
DATACLAS=(LISTDATA,SCRIPT%,$NULL,TST12,NE)
```

DDNAME

The DDNAME keyword specifies a ddname that is assigned to a data set.

Syntax



DDNAME=(*ddname* | *user_var*, *ddname* | *user_var*,...,op)

Where:

ddname is a 1 - 8 character ddname. Wildcard characters are permitted.

Note: VSAM clusters that are defined through IDCAMS do not have a defined DDNAME.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
DD01 DD ......
```

Examples

```
DDNAME=SYSUT1
DDNAME=(SYSUT1,WORK*,EQ)
DDNAME=(&NAMES)
```

DEFERMOUNT

(Tape devices only.) The DEFERMOUNT keyword specifies whether the DEFER option was specified for the UNIT keyword.

Syntax

DEFERMOUNT=YES | NO

Operands

Specify one of the following operands:

YES
DEFER was specified.

NO DEFER was not specified.

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE, and CATCHECK.

JCL equivalent

```
DD01 DD UNIT=(3490,,DEFER)
```

Related Enhanced Allocation Management function

SET_DEFERMOUNT

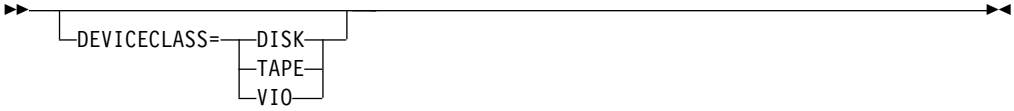
Example

```
DEFER=Y
```

DEVICECLASS

The DEVICECLASS keyword specifies the type of device that is assigned to the data set.

Syntax



DEVICECLASS=DISK | TAPE | VIO

Operands

Specify one of the following operands:

DISK
The data set is assigned to a disk device.

TAPE
The data set is assigned to a magnetic tape device.

VIO
The data set is defined to a VIO device.

Note: DEVICECLASS cannot be determined for new SMS data set allocations that do not have a UNIT and VOLSER specification in the JCL or dynamic allocation request. If the DEVICECLASS is not known, any DEVICECLASS selection criterion that is specified in a rule definition is considered a non-match and the rule definition is not selected. Therefore, do not use DEVICECLASS as a sole selection criterion in environments where new SMS data sets are allocated without a UNIT and VOLSER specification.

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

None (although the UNIT that is assigned to the device indirectly defines the device class).

Example

```
DEVICEC=D
```

DIRBLKS

(PDS and PDS/E data sets only.) The DIRBLKS keyword specifies the number of directory blocks that are assigned to a PDS or PDS/E data set.

Syntax

DIRBLKS=(*nnnnn*,op)

Where:

nnnnn is an integer (0 - 65535).

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE, CATCHECK, and STOW.

JCL equivalent

```
DD01 DD SPACE=(,(,,120))
```

Related Enhanced Allocation Management function

SET_DIRBLKS

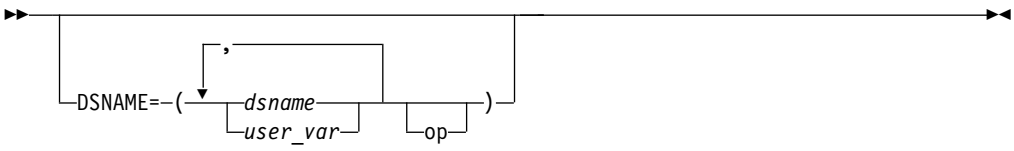
Example

```
DIRBLKS=120
```

DSNAME

The DSNAME keyword specifies a data set name.

Syntax



DSNAME=(*dsname* | *user_var*, *dsname* | *user_var*,...,op)

Note: For VSAM, the DSNAME keyword defines the cluster name.

Where:

dsname is a 1 - 44 character data set name. Wildcard characters are permitted.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
DD01 DD DSN=PAY01.FILE
```

IDCAMS equivalent

```
DEFINE CLUSTER (NAME(PAY01.FILE) ...)
```

Examples

```
DSNAME=PDGLO.TEST.DATA.SET
DSNAME=&TESTDAT
DSNAME=(ABC.WRK*,LT)
DSNAME=(TST.WORK.DATA,ABC.DATA,D18.AUD.DATA)
```

DSNAME_DATA

(VSAM data sets only.) The DSNAME_DATA keyword specifies the data set name of the data component of a VSAM cluster.

Syntax

DSNAME_DATA=(*dsname*,op)

Where:

dsname is a 1 - 44 character data set name. Wildcard characters are permitted.

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

IDCAMS equivalent

```
DEFINE CL (...) DATA(NAME(PAY01.FILE))
```

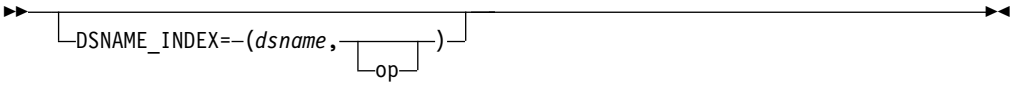
Example

```
DSNAME_DATA=PAY01.FILE
```

DSNAME_INDEX

(VSAM data sets only.) The DSNAME_INDEX keyword specifies the data set name of the index component of a VSAM cluster.

Syntax



DSNAME_INDEX=(*dsname*,op)

Where:

dsname is a 1 - 44 character data set name. Wildcard characters are permitted.

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

IDCAMS equivalent

```
DEFINE CL (...) INDEX(NAME(PAY01.FILE))
```

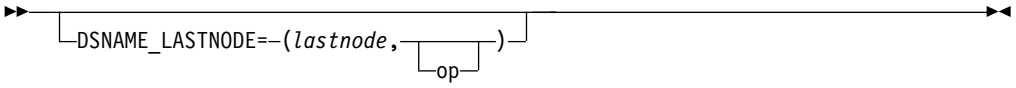
Example

```
DSNAME_INDEX=PAY01.FILE
```

DSNAME_LASTNODE

The DSNAME_LASTNODE keyword specifies the last qualifier of a data set name.

Syntax



DSNAME_LASTNODE=(*lastnode*,op)

Where:

lastnode is a 1 - 8 character data set name qualifier. Wildcard characters are permitted.

Note: For VSAM, the DSNAME_LASTNODE keyword pertains to the cluster name.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, CATCHECK, and STOW.

JCL equivalent

```
DD01 DD DSN=QUAL1.QUAL2.QUAL3.QUALLAST
```

IDCAMS equivalent

```
DEFINE CLUSTER (NAME(QUAL1.QUAL2.QUAL3.QUALLAST) ...)
```

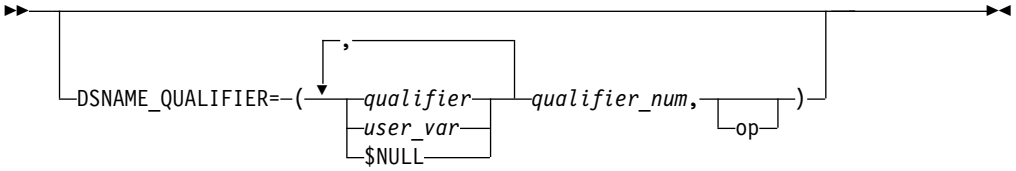
Example

```
DSNAME_LASTNODE=QUALLAST
```

DSNAME_QUALIFIER

The DSNAME_QUALIFIER keyword specifies one of the qualifiers of a data set name.

Syntax



DSNAME_QUALIFIER=(*qualifier* | *user_var*, *qualifier* | *user_var*,..., | $NULL, *qualifier_num*,op)

Note: For VSAM, the DSNAME_QUALIFIER keyword pertains to the cluster name. If you do not specify a qualifier number, the DSNAME_QUALIFIER keyword is ignored.

Where:

qualifier
is the 1 - 8 character data set name qualifier. Wildcard characters are permitted.

qualifier_num
is the position (1 - 22) of the qualifier in the data set name. When multiple qualifiers are specified, *qualifier_num* applies to the entire list. For example, if you specify `DSNAME_QUALIFIER=(QUAL1,$NULL,TEST*,2,EQ)` the *qualifier_num* of 2 indicates that qualifier position 2 applies to all qualifiers that are specified in the list.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, CATCHECK, and STOW.

JCL equivalent

```
DD01 DD DSN=QUAL1.QUAL2.QUAL3.QUAL4
```

IDCAMS equivalent

```
DEFINE CLUSTER (NAME(QUAL1.QUAL2.QUAL3.QUAL4) ...)
```

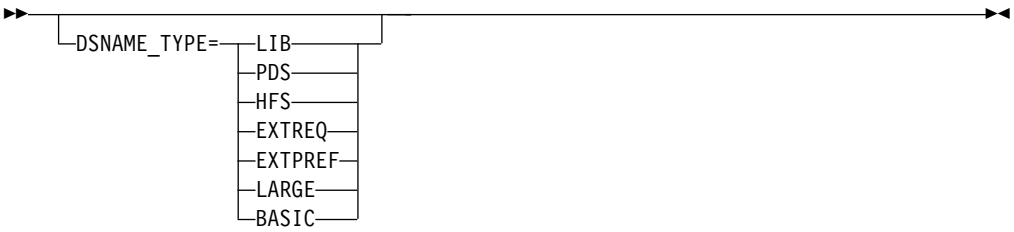
Examples

```
DSNAME_QUALIFIER=(QUAL2, 2)
DSNAME_QUALIFIER=(&QUALIF,2)
DSNAME_QUALIFIER=(DATA,INFO,5)
DSNAME_QUALIFIER=($NULL,5)
```

DSNAME_TYPE

The DSNAME_TYPE keyword specifies the type of data set.

Syntax



DSNAME_TYPE=LIB | PDS | HFS | EXTREQ | EXTPREF | LARGE | BASIC

Operands

Specify one of the following operands:

LIB
PDSE data set.

PDS
Partitioned data set (PDS).

HFS
Hierarchical file system data set (HFS data set).

EXTREQ
Extended format data set if the data set is VSAM, sequential, or if DSORG is omitted from all sources.

EXTPREF
Prefer extended format data set if the data set is VSAM, sequential, or if DSORG is omitted from all sources.

LARGE
Allows the specification of more than 65535 tracks for the primary and secondary allocation for non-VSAM data sets.

BASIC
Allows the specification of fewer than 65535 tracks for the primary and secondary allocation for non-VSAM data sets.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, CATCHECK, and STOW.

JCL equivalent

```
DD01 DD DSNTYPE=HFS
```

Data class equivalent

Data Set Name Type and If Ext parameters.

Related Enhanced Allocation Management function

SET_DSNAME_TYPE

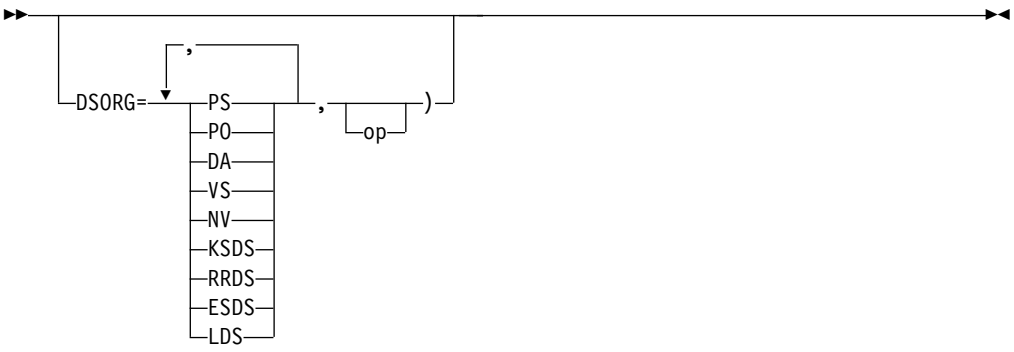
Example

```
DSNAME_TYPE=HFS
```

DSORG

The DSORG keyword specifies the data set organization for non-VSAM data sets and the record organization for VSAM data sets.

Syntax



DSORG=(PS, PO, DA, VS, NV, KSDS, RRDS, ESDS, LDS,...,op)

Operands

Specify one or more of the following operands:

PS Physical sequential.

PO Partitioned organized.

DA Direct access.

VS VSAM.

NV Non-VSAM.

KSDS
VSAM key-sequenced data set.

RRDS
A VSAM variable-length or fixed-length relative-record data set.

ESDS
VSAM entry-sequenced data set.

LDS
VSAM linear data set.

Note:

- One of these comparator operators are permitted: EQ or NE.
- Wildcard characters are not permitted.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, CATCHECK, and STOW.

Note: The CATCHECK environment does not apply to VSAM data sets.

JCL equivalent

```
DD01 DD DSORG=PS
DD01 DD RECORG=KS
```

IDCAMS equivalent

```
DEFINE CLUSTER(INDEXED ...)
DEFINE CLUSTER(NONINDEXED ...)
DEFINE CLUSTER(LINEAR ...)
DEFINE CLUSTER(NUMBERED ...)
```

Data class equivalent

Recorg

Related Enhanced Allocation Management function

SET_DSORG

Example

```
DSORG=PS
DSORG=(PO,PS,DA,EQ)
```

EATTR

The EATTR keyword specifies the assigned Extended Attribute value for a data set. z/OS uses the Extended Attributes value to assign eligibility for use of cylinder-managed space on an Extended Address Volume (EAV).

Syntax

EATTR=OPT | NO | $NULL

Operands

OPT
Extended Attributes are optional.

NO No Extended Attributes.

$NULL
Extended Attribute options were not specified.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, and CATCHECK.

AMS equivalent

```
DEFINE CLUSTER (EATTR(OPT) ....)
```

Data class equivalent

EATTR

JCL equivalent

```
DD01 DD EATTR=OPT
```

Related Enhanced Allocation Management function

SET_EATTR

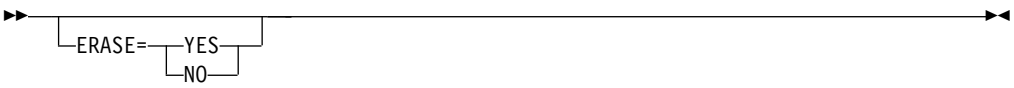
Example

```
EATTR=OPT
```

ERASE

(VSAM data sets only.) The ERASE keyword specifies whether the VSAM ERASE option was specified.

Syntax



ERASE=YES | NO

Operands

Specify one of the following operands:

YES
ERASE was specified.

NO NOERASE was specified.

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

IDCAMS equivalent

```
DEFINE CL (ERASE ....)
```

Data class equivalent

None.

Related Enhanced Allocation Management function

SET_ERASE

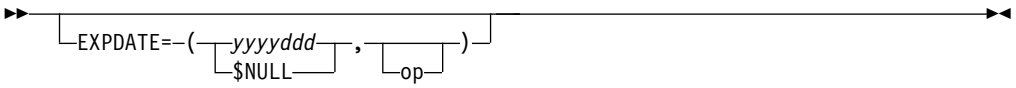
Example

```
ERASE=YES
```

EXPDATE

The EXPDATE keyword specifies an expiration date for a data set.

Syntax



EXPDATE=(*yyyyddd* | $NULL,op)

Where:

yyyyddd is a Julian date (*yyyy* is a four-digit year and *ddd* is a three-digit day).

Note: If you specify EXPDATE and RETPD, EXPDATE takes precedence.

Because the system uses the same internal field to store the EXPDATE and RETPD value, if one or the other is specified in the JCL or DEFINE statement, the EXPDATE or RETPD selection criteria keyword can be used as a selection criterion provided the proper date format is specified.

For example, if you specify a RETPD of 1 and the creation date was 2008001 as expressed in EXPDATE format, the value of the field as stored internally by the system is 2008002. Therefore, either of the following selection criteria will match:

```
RETPD=1
EXPDATE=2008002
```

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
DD01 DD EXPDT=2009/003
```

IDCAMS equivalent

```
DEFINE CLUSTER(TO(2009003) ..)
```

Data class equivalent

Retpd or Expdt.

Related Enhanced Allocation Management function

SET_EXPDATE

Example

```
EXPDATE=2009003
```

EXTADDR

(VSAM data sets only.) The EXTADDR keyword specifies the Extended Addressability attribute for a VSAM cluster.

Syntax

EXTADDR=YES|NO

Operands

Specify one of the following operands:

YES
Extended addressability was defined for the data set.

NO Extended addressability was *not* defined for the data set.

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

Data class equivalent

Extended Addressability.

Related Enhanced Allocation Management function

SET_EXTADDR.

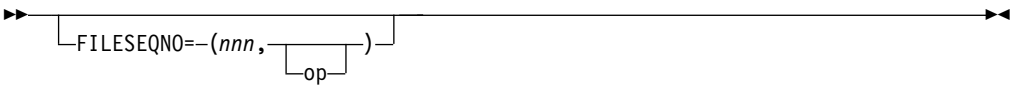
Example

```
EXTADDR=N
```

FILESEQNO

(Tape data sets only.) The FILESEQNO keyword specifies a file sequence number of a data set.

Syntax



FILESEQNO=(*nnn*,op)

Where:

nnn is the sequence number (0 - 255).

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE, and CATCHECK.

JCL equivalent

```
DD01 DD LABEL=(4,)
```

Related Enhanced Allocation Management function

SET_FILESEQNO

Example

```
FILESEQNO=4
```

FRLOG

(VSAM data sets only.) The FRLOG keyword specifies whether VSAM batch logging was specified for a VSAM data set.

Syntax

FRLOG=REDO | NONE | ALL | UNDO | $NULL

Operands

Specify one of the following operands:

REDO
Forward recovery was requested.

NONE
The VSAM batch logging option was not requested.

ALL
Backward and forward recovery was requested.

UNDO
Backward recovery was requested.

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

IDCAMS equivalent

```
DEFINE CL (FRLOG(REDO) ...)
```

Data class equivalent

FRlog

Related Enhanced Allocation Management function

SET_FRLOG

Example

```
FRLOG=REDO
```

FSPC_CA

(VSAM data sets only.) The FSPC_CA keyword specifies the freespace percentage for the control area of a VSAM data set.

Syntax

FSPC_CA=(*nnn*,op)

Where:

nnn is an integer (0 - 100).

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

IDCAMS equivalent

```
DEFINE CL (FREESPACE(0 10) ...)
```

Data class equivalent

% Freespace CA

Related Enhanced Allocation Management function

SET_FSPC_CA

Example

```
FSPC_CA=10
```

FSPC_CI

(VSAM data sets only.) The FSPC_CI keyword specifies the freespace percentage for the control interval of a VSAM data set.

Syntax

FSPC_CI=(*nnn*,op)

Where:

nnn is an integer (0 - 100).

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

IDCAMS equivalent

```
DEFINE CL (FREESPACE(10 0) ...)
```

Data class equivalent

% Freespace CI

Related Enhanced Allocation Management function

SET_FSPC_CI

Example

```
FSPC_CI=10
```

GROUPID

The GROUPID keyword specifies a security group (such as a RACF or equivalent security product) identifier.

Syntax

GROUPID=(*group_id*,op)

Where:

group_id is a 1 - 8 character group ID. Wildcard characters are permitted.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
JJHTEST01 JOB ,,GROUP=DEVELOPR
```

Example

```
GROUPID=DEVELOPR
```

GUARSPACE

The GUARSPACE keyword specifies the Guaranteed Space attribute for the data set as defined in the SMS storage class.

Syntax

GUARSPACE=YES | NO

Operands

Specify one of the following operands:

YES
Guaranteed Space was specified.

NO Guaranteed Space was not specified.

Note:

- Guaranteed Space is defined in the SMS storage class.
- The GUARSPACE keyword requires additional calls to SMS Configuration Services and its presence in the RULEDEFS member can cause a measurable increase in CPU time for all tasks in the system, especially tasks that perform large numbers of allocations. Therefore, carefully consider the specification of the GUARSPACE keyword in your rule definition member.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

Storage class equivalent

Guaranteed Space

Example

```
GUARSPACE=YES
```

ILOAD

(VSAM data sets only.) The ILOAD keyword specifies the initial load option that is specified for a VSAM data set.

Syntax

ILOAD=SPEED | RECOVERY

Operands

Specify one of the following operands:

SPEED
The SPEED option was specified.

RECOVERY
The RECOVERY option was specified.

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

IDCAMS equivalent

```
DEFINE CLUSTER (RECOVERY ...)
DEFINE CLUSTER (SPEED ...)
```

Data class equivalent

Initial Load

Related Enhanced Allocation Management function

SET_ILOAD

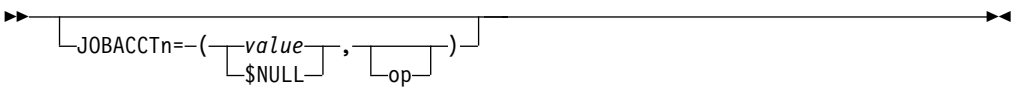
Example

```
ILOAD=S
```

JOBACCTn

(VSAM and non-VSAM.) The JOBACCTn keyword specifies one of the first fifteen Job Accounting field specifications.

Syntax



JOBACCTn=(*value*|$NULL,op)

Where:

n is an integer (1 - 15).

value is a character string value that has a maximum length of 144. If the string contains special characters or blanks, enclose the string with double quotation marks. Wildcard characters are permitted.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
JOB ,'accounting information'
```

Example

```
JOBACCTn= "Department 5 - Room 5A"
```

JOBCLASS

The JOBCLASS keyword specifies a single character JES job class.

Syntax

JOBCLASS=(*class*,op)

Where:

class is one character that specifies the job class. Wildcard characters are permitted.

Default

None.

JCL equivalent

```
JJHTEST01 JOB ,,CLASS=A
```

Environments

ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

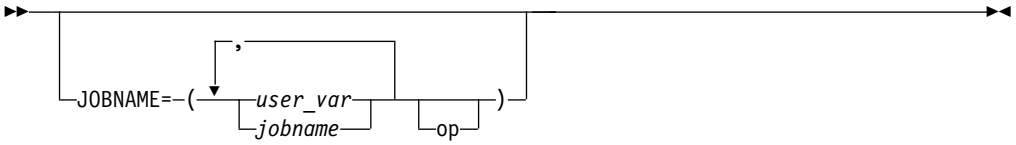
Example

```
JOBCLASS=A
```

JOBNAME

The JOBNAME keyword specifies the name of the job, started task, or TSO user ID that is associated with the task that is processing a data set.

Syntax



JOBNAME=(*jobname* | *user_var*, *jobname* | *user_var*,...,op)

Where:

jobname is the 1 - 8 character name of the job, started task, or TSO user ID. Wildcard characters are permitted.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
JJHTEST1 JOB ...
```

Examples

```
JOBNAME=JJHTEST
JOBNAME=&JNAMES
JOBNAME=(PAYROLL,ACCT,EQ)
```

JOBTYPE

The JOBTYPE keyword specifies the type of address space.

Syntax

JOBTYPE=JOB | STC | TSU | APPC

Operands

Specify one of the following operands:

JOB
Normal job address space.

STC
Started task.

TSU
Time-sharing user.

APPC
APPC address space.

Note: Wildcard characters are not permitted.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

Example

```
JOBTYPE=JOB
```

KEY_LEN

(Key sequenced VSAM data sets only.) The KEY_LEN keyword specifies the key length of a key-sequenced VSAM data set.

Syntax

KEY_LEN=(*nnn*,op)

Where:

nnn is the length of a key (1 - 255).

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

JCL equivalent

```
DD01 DD KEYLEN=6,...
```

IDCAMS equivalent

```
DEFINE CLUSTER(KEYS(8 ..) ...)
```

Data class equivalent

Keylen

Related Enhanced Allocation Management function

SET_KEY_LEN

Example

```
KEY_L=12
```

KEY_OFF

(Key sequenced VSAM data sets only.) The KEY_OFF keyword specifies the offset of the key in a key-sequenced VSAM data set.

Syntax

KEY_OFF=(*nnnnn*,op)

Where:

nnnnn is the offset of the key (0 - 32767) with a data record.

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

JCL equivalent

```
DD01 DD KEYOFF=0,...
```

IDCAMS equivalent

```
DEFINE CLUSTER(KEYS(... 12) ...)
```

Data class equivalent

Keyoff

Related Enhanced Allocation Management function

SET_KEY_OFF

Example

```
KEY_O=12
```

LOGGING

(VSAM RLS data sets only.) The LOGGING keyword specifies the type of logging performed for the sphere of a VSAM RLS data set.

Syntax

LOGGING=BACK | BOTH | NONE | $NULL

Operands

Specify one of the following operands:

BACK
Backward recovery.

BOTH
Backward and forward recovery.

NONE
Neither forward or backward recovery.

Default

None. The LOGGING keyword has no default.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

IDCAMS equivalent

```
DEFINE CL(LOG(UNDO))
```

Data class equivalent

Log

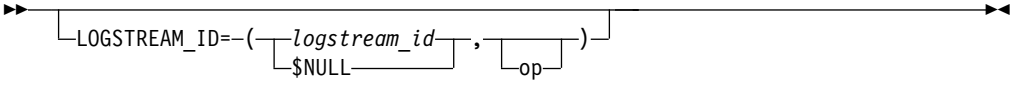
Example

```
LOGGING=BOTH
```

LOGSTREAM_ID

(SMS-managed VSAM data sets only.) The LOGSTREAM_ID keyword specifies the name of a logstream data set for an SMS-managed VSAM data set.

Syntax



LOGSTREAM_ID=(*logstream_id* | $NULL,op)

Where:

logstream_id is a string of 1 - 26 characters that names the logstream. Wildcard characters are permitted.

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

JCL equivalent

```
DD01 DD LGSTREAM=PAY.STREAM
```

IDCAMS equivalent

```
DEFINE CLUSTER(LOGSTREAMID(PAY.STREAM) ...)
```

Data class equivalent

Logstream ID

Related Enhanced Allocation Management function

SET_LOGSTREAM_ID

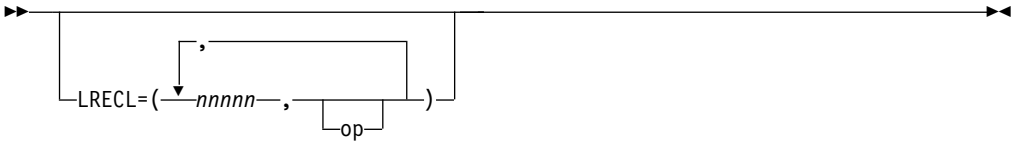
Example

```
LOGSTREAM_ID=PAY.STREAM
```

LRECL

The LRECL keyword specifies the record length of a data set.

Syntax



LRECL=(*nnnnn*,op,*nnnnn*,op)

Where:

nnnnn is an integer (1 - 32767) that defines the length of a non-VSAM record or the average and maximum record sizes of a VSAM data set.

Note: For non-VSAM data sets the second length and comparison operand are ignored.

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

JCL equivalent

```
DD01 DD LRECL=80,....
```

IDCAMS equivalent

```
DEFINE CL (RECORDSIZE(80 120) ...)
```

Data class equivalent

Lrecl

Related Enhanced Allocation Management function

SET_LRECL

Example

```
LRECL=(80,,120)
```

MAXSPACE

The MAXSPACE keyword specifies the maximum potential space on the first volume of a data set.

Syntax

MAXSPACE=(*nnnnnnnn*,op)

Where:

nnnnnnnn is an integer (0 - 16777215) that specifies the maximum potential space (in allocation units) that a data set can occupy on the first volume.

Note:

- For non-VSAM data sets, MAXSPACE is calculated by adding the space that is required for the primary allocation to the space required for 15 extents worth of the secondary allocation. The Advanced Allocation Management MAXSPACE value is expressed in the specified units of allocation (CYL, TRK, and so on), not as the equivalent number of bytes.
- For VSAM data sets, MAXSPACE is calculated by adding the space that is required for the primary allocation to the space required for 254 extents worth of secondary allocation.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
DD01 DD SPACE=(CYL,(1,1)),...
```

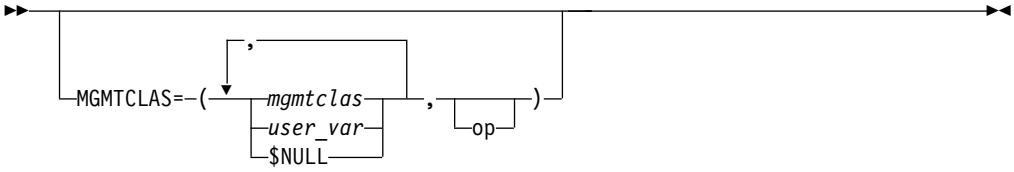
Example

```
MAXSP=16577225
```

MGMTCLAS

(SMS-managed data sets only.) The MGMTCLAS keyword specifies the SMS management class that is assigned to an SMS data set.

Syntax



MGMTCLAS=(*mgmtclas* | *user_var* | $NULL, *mgmtclas* | *user_var* | $NULL,...,op)

Where:

mgmtclas is the 1 - 8 character name that is assigned to the SMS management class. Wildcard characters are permitted.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

Note: During ACS processing, the value that is used is the value that is assigned by the user or the system before Advanced Allocation Management processing. In the other environments, the value that is used is the value that is assigned by Advanced Allocation Management (if specified); otherwise, it is the value that is assigned by the user or by the system.

JCL equivalent

```
DD01 DD MGMTCLAS=STANDARD
```

IDCAMS equivalent

```
DEFINE CLUSTER(MANAGEMENTCLASS(LONGTERM) ...)
```

Related Enhanced Allocation Management function

SET_MGMTCLAS

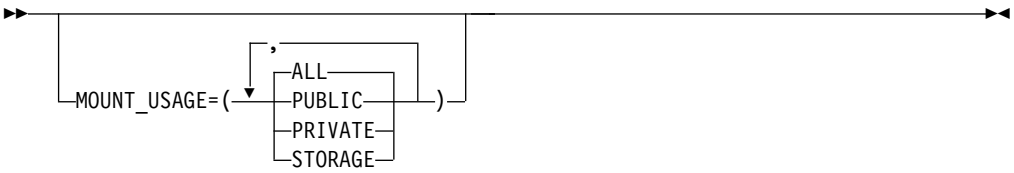
Examples

```
MGMTCLAS=TSO
MGMTCLAS=(&CLASSA,&CLASSC)
MGMTCLAS=(S1P01,MC06,EQ)
MGMTCLAS=($NULL, MC06,S1P01,AMG*)
```

MOUNT_USAGE

(Non-SMS, non-VSAM data sets only.) MOUNT_USAGE specifies the mount usage type of the current volume at the time an Advanced Allocation Management function is performed.

Syntax



MOUNT_USAGE=(PUBLIC | PRIVATE | STORAGE | ALL)

Operands

PUBLIC
The volume is mounted PUBLIC.

PRIVATE
The volume is mounted PRIVATE.

STORAGE
The volume is mounted STORAGE.

ALL
The volume can be mounted with any status.

Notes:

1. Wildcard characters are not permitted.
2. You can specify more than one keyword value. For example: `MOUNT_USAGE=(PRIVATE,PUBLIC,STORAGE,ALL)`.
3. For the PRIM_REDUCE function and all SET functions, for the initial allocation on the first volume of supported data sets, the mount usage type is only known where the VOLSER is identified in the JCL or dynamic allocation request.

   If the mount usage type is not known, any MOUNT_USAGE selection criterion that is specified in a rule definition is considered a non-match and the rule definition is not selected. Consequently, the PRIM_REDUCE function and all SET functions for the initial allocation on the first volume of a data set should not be used with the MOUNT_USAGE keyword.

Default

ALL

Environments

ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

Note: For any function that is pertinent to the ACS or ALLOCATION environments, the mount usage type is only known where the VOLSER is identified in the JCL or dynamic allocation request.

Example

```
MOUNT_USAGE=PRIVATE
```

NDISP

The NDISP keyword specifies the normal termination disposition for a data set.

Syntax

NDISP=PASS | KEEP | CATLG | UNCATLG | DELETE

Operands

Specify one of the following operands:

PASS
The specified or defaulted normal disposition is DISP=(,PASS).

KEEP
The specified or defaulted normal disposition is DISP=(,KEEP).

CATLG
The specified or defaulted normal disposition is DISP=(,CATLG).

UNCATLG
The specified or defaulted normal disposition is DISP=(,UNCATLG).

DELETE
The specified or defaulted normal disposition is DISP=(,DELETE).

Notes:

1. The NDISP keyword does not apply to VSAM IDCAMS DEFINE statements.
2. The system takes the following defaults with the normal termination disposition parameter:
   - If you omit the normal termination disposition, the default is DELETE for a NEW data set and KEEP for an existing data set.
   - If you omit the DISP parameter, the default disposition is DELETE for the normal termination disposition.

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
DD01 DD DISP=(,DELETE,)
```

Related Enhanced Allocation Management function

SET_NDISP

Example

```
NDISP=CATLG
```

NUMVOLS

The NUMVOLS keyword specifies the number of volumes that are defined to a VSAM cluster.

Syntax

NUMVOLS=(*nn*,op)

Where:

nn is an integer (1 - 59) that specifies the number of volumes.

Note: The volume count is determined by the number of unique, specific, volsers for the data and index components (duplicate volsers are counted once). When non-specific volser entries denoted by an asterisk are specified, the larger of the number of non-specific volsers that are specified for the data and index components are added to the number of unique specific volsers, if any.

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE.

IDCAMS equivalent

```
DEFINE CLUSTER(VOLUMES(VOL001 VOL002) ...)
DEFINE CLUSTER(VOLUMES(* * * *) ...)
```

Data class equivalent

Volume Count

Related Enhanced Allocation Management function

SET_NUMVOLS

Example

```
NUMVOLS=5
```

NUMVOLS_DATA

The NUMVOLS_DATA keyword specifies the number of volumes to be defined for the data component of a VSAM cluster.

Syntax

NUMVOLS_DATA=(*nn*,op)

Where:

nn is an integer (1 - 59) that specifies the number of volumes.

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE.

IDCAMS equivalent

```
DEFINE CLUSTER(...) DATA(VOLUMES(VOL001 VOL002) ...)
DEFINE CLUSTER(...) DATA(VOLUMES(* * * *) ...)
```

Related Enhanced Allocation Management function

SET_NUMVOLS_DATA

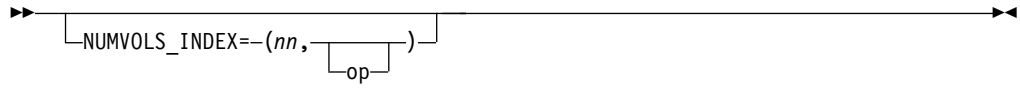
Example

```
NUMVOLS_DATA=5
```

NUMVOLS_INDEX

The NUMVOLS_INDEX keyword specifies the number of volumes to be defined for the index component of a VSAM cluster.

Syntax



NUMVOLS_INDEX=(*nn*,op)

Where:

nn is an integer (1 - 59) that specifies the number of volumes.

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE.

IDCAMS equivalent

```
DEFINE CLUSTER(...) INDEX(VOLUMES(VOL001 VOL002) ...)
DEFINE CLUSTER(...) INDEX(VOLUMES(* * * *) ...)
```

Related Enhanced Allocation Management function

SET_NUMVOLS_INDEX

Example

```
SET_NUMVOLS_INDEX=5
```

OWNERID

The OWNERID keyword specifies the owner identification for the data set as stored in the catalog.

Syntax

OWNERID=(*owner_id*,op)

Where:

owner_id is the 1 - 8 character owner identification. Wildcard characters are permitted.

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

IDCAMS equivalent

```
DEFINE CL (OWNER(PLEXAA) ...)
```

Example

```
OWN=PLEXAA
```

PQTY

The PQTY keyword specifies the primary quantity value that is assigned to the data set.

Syntax

PQTY=(*nnnnnnnn*,op)

Where:

nnnnnnnn is an integer (0 - 16777215) that specifies the primary quantity for a data set.

- For new non-VSAM data sets, it is the actual amount requested.
- For non-VSAM new data sets, it is the amount of space that is allocated to the first extent.
- For VSAM data sets, it is the requested primary quantity; the value is represented in space allocation units. For example, `SPACE=(TRK,(5,2))` has a PQTY value of 5 as would `SPACE=(CYL,(5,2))`.

Note: When a DEFINE is performed for a VSAM data set and space is specified using the RECORDS, KILOBYTES, or MEGABYTES parameters, IDCAMS converts the specifications to tracks or cylinders and then saves the converted specification in the catalog. Therefore, if you specify a rule definition that checks SPACEUNITS for performing a function that occurs during VSAM end of volume (such as SEC_ALLOC), SPACEUNITS=RECORDS, KB, or MB does not generate a match. This can affect PQTY (primary quantity) matching. For example, if a DEFINE specifies MB(1), the specification is converted to two cylinders and saved, and then that value is saved in the catalog. Therefore, attempting to check PQTY=1 for a function that is performed during VSAM end-of-volume also causes a non-match to occur.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
DD01 DD SPACE=(...,(1))
```

IDCAMS equivalent

```
DEFINE CLUSTER(CYLINDERS(1) ....)
DEFINE CLUSTER(TRACKS(1) ....)
DEFINE CLUSTER(RECORDS(1000) ....)
DEFINE CLUSTER(MEGABYTES(2) ....)
DEFINE CLUSTER(KILOBYTES(2048) ...)
```

Related Enhanced Allocation Management function

SET_PQTY

Example

```
PQTY=2048
```

PQTY_DATA

(VSAM data sets only.) The PQTY_DATA keyword specifies the primary quantity value that is assigned to the data component of a VSAM data set.

Syntax

PQTY_DATA=(*nnnnnnnn*,op)

Where:

nnnnnnnn is an integer (0 - 16777215) that specifies the requested primary allocation quantity for the data component in the requested allocation units.

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

IDCAMS equivalent

```
DEFINE CLUSTER(...) DATA(CYLINDERS(1) ....)
DEFINE CLUSTER(...) DATA(TRACKS(1) ....)
DEFINE CLUSTER(...) DATA(RECORDS(1000) ....)
DEFINE CLUSTER(...) DATA(MEGABYTES(2) ....)
DEFINE CLUSTER(...) DATA(KILOBYTES(2048) ...)
```

Related Enhanced Allocation Management function

SET_PQTY_DATA

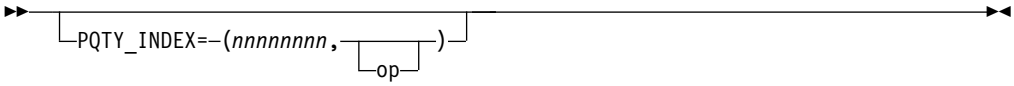
Example

```
PQTY_DATA=2048
```

PQTY_INDEX

(VSAM data sets only.) The PQTY_INDEX keyword specifies the primary quantity value that is assigned to the index component of a VSAM data set.

Syntax



PQTY_INDEX=(*nnnnnnnn*,op)

Where:

nnnnnnnn is an integer (0 - 16777215) that specifies the requested primary allocation quantity for the index component in the requested allocation units.

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

JCL equivalent

```
DEFINE CLUSTER(...) INDEX(CYLINDERS(1) ....)
DEFINE CLUSTER(...) INDEX(TRACKS(1) ....)
DEFINE CLUSTER(...) INDEX(RECORDS(1000) ....)
DEFINE CLUSTER(...) INDEX(MEGABYTES(2) ....)
DEFINE CLUSTER(...) INDEX(KILOBYTES(2048) ...)
```

Related Enhanced Allocation Management function

SET_PQTY_INDEX

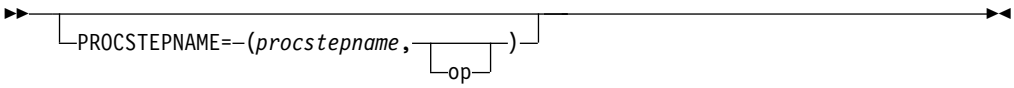
Example

```
PQTY_INDEX=1000
```

PROCSTEPNAME

The PROCSTEPNAME keyword specifies a procedure step name.

Syntax



PROCSTEPNAME=(*procstepname*,op)

Where:

procstepname is the 1 - 8 character procedure step name. Wildcard characters are permitted.

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
PSTEP01 EXEC PROC01
```

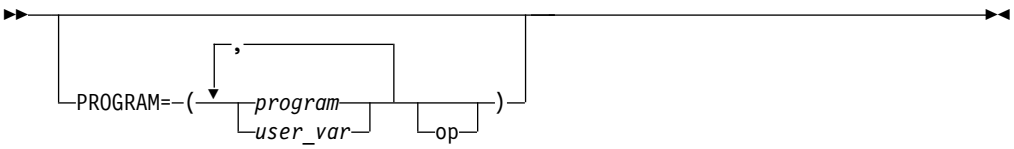
Example

```
PROCSTEPNAME=JJHPROC
```

PROGRAM

The PROGRAM keyword specifies the job-step program name that is active when a particular Advanced Allocation Management function is performed.

Syntax



PROGRAM=(*program* | *user_var*, *program* | *user_var*,...,op)

Where:

program is the 1 - 8 character program name. Wildcard characters are permitted.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
STEP01 EXEC PGM=HUMRSR01
```

Examples

```
PROGRAM=RDEFDRVR
PROGRAM=&PNAMES
PROGRAM=(RDEF*,IDC*,LT)
```

PSPACE_KB

The PSPACE_KB keyword specifies the requested or actual primary space that is allocated to a data set in kilobytes.

Syntax

PSPACE_KB=(*nnnnnnnn*,op)

Where:

nnnnnnnn is an integer (0 - 16777215) that specifies the number of kilobytes for the primary allocation quantity of a data set.

- For a new non-VSAM data set, this value is the requested primary allocation quantity.
- For an existing non-VSAM data set, this value is the actual space that is allocated to the first extent of a data set.
- For VSAM data sets, this value is the requested primary allocation quantity.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

Note: For the ACS environment the value is obtained from the SMS ACS variable &SIZE.

JCL equivalent

```
DD01 DD SPACE=(TRK,(1,1))
```

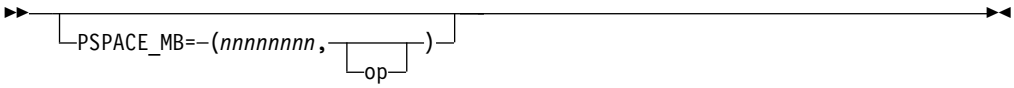
Example

```
PSPACE_KB=16577200
```

PSPACE_MB

The PSPACE_MB keyword specifies the requested, or actual, primary space that is allocated to a data set in megabytes.

Syntax



PSPACE_MB=(*nnnnnnnnn*,op)

Where:

nnnnnnnnn is an integer (0 - 16777215) that specifies the number of megabytes for the primary allocation quantity of a data set.

- For a new non-VSAM data set this value is the requested primary allocation quantity.
- For an existing non-VSAM data set this value is the actual space that is allocated to the first extent of a data set.
- For VSAM data sets this value is the requested primary allocation quantity.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

Note: For the ACS environment the value is calculated from the ACS variable &SIZE.

JCL equivalent

```
DD01 DD SPACE=(1,(5,1)),AVGREC=M
```

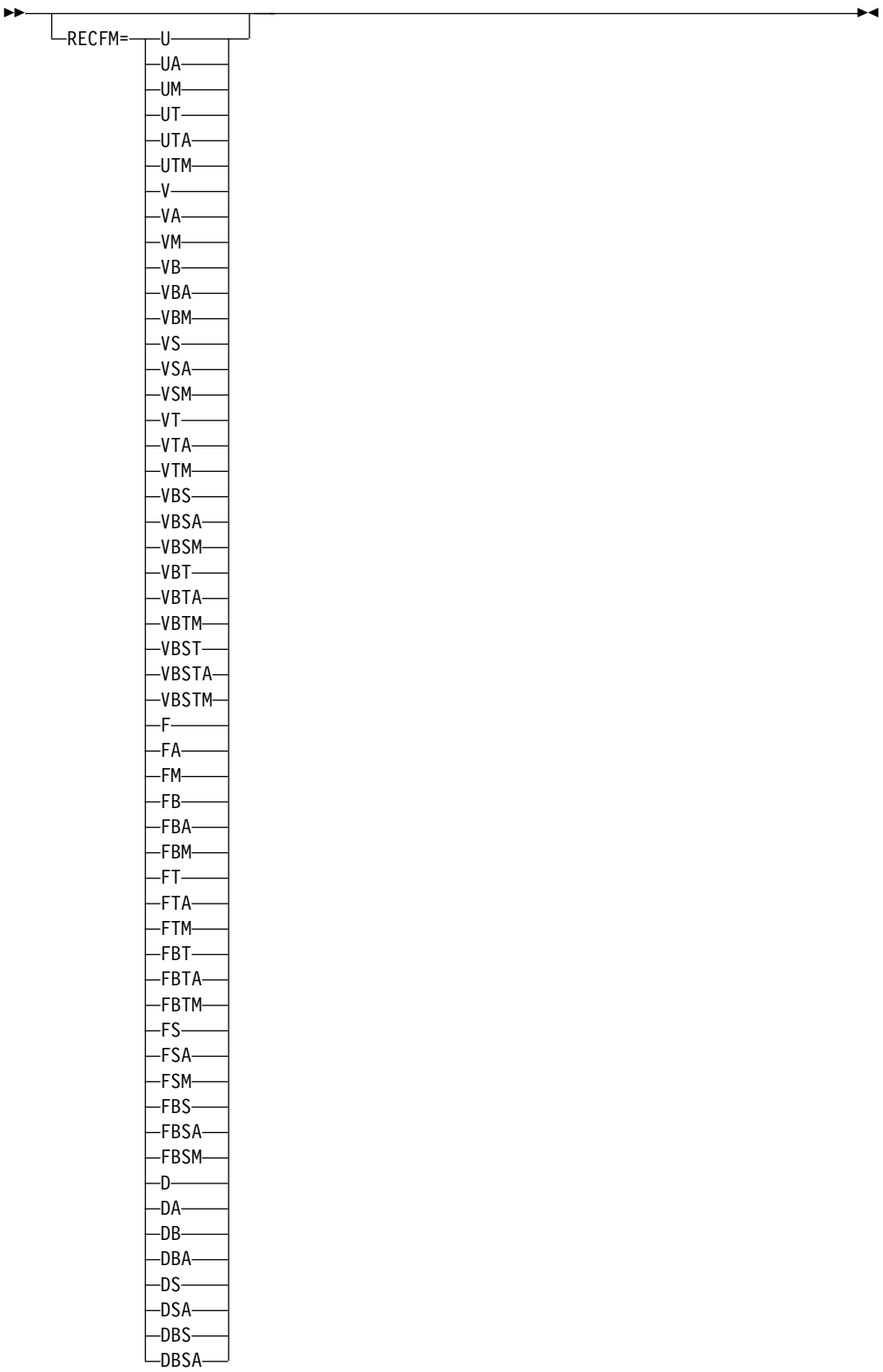
Example

```
PSPACE_MB=5
```

RECFM

(Non-VSAM data sets only.) The RECFM keyword specifies the record format for a non-VSAM data set.

Syntax



RECFM=U | UA | UM | UT | UTA | UTM | V | VA | VM | VB | VBA | VBM | VS | VSA | VSM | VT | VTA | VTM | VBS | VBSA | VBSM | VBT | VBTA | VBTM | VBST | VBSTA | VBSTM | F | FA | FM | FB | FBA | FBM | FT | FTA | FTM | FBT | FBTA | FBTM | FS | FSA | FSM | FBS | FBSA | FBSM | D | DA | DB | DBA | DS | DSA | DBS | DBSA

Operands

D ASCII variable-length.

DA ASCII variable-length (ANSI format).

DB ASCII variable-length (blocked).

DBA
ASCII variable-length (blocked, ANSI format).

DBS
ASCII variable-length (blocked, spanned).

DBSA
ASCII variable-length (blocked, spanned, ANSI format).

DS ASCII variable-length (spanned).

DSA
ASCII variable-length (spanned, ANSI format).

F Fixed length.

FA Fixed length (ANSI format).

FB Fixed length (blocked).

FBA
Fixed length (blocked, ANSI format).

FBM
Fixed length (blocked, machine code format).

FBS
Fixed length (blocked, standard blocks).

FBSA
Fixed length (blocked, standard blocks, ANSI format).

FBSM
Fixed length (blocked, standard blocks, machine code format).

FBT
Fixed length (blocked, track overflow).

FBTA
Fixed length (blocked, track overflow, ANSI format).

FBTM
Fixed length (blocked, track overflow, machine code format).

FM Machine code format.

FS Fixed length (standard blocks).

FSA
Fixed length (standard blocks, ANSI format).

FSM
Fixed length (standard blocks, machine code format).

FT Fixed length (track overflow).

FTA
Fixed length (track overflow, ANSI format).

FTM
Fixed length (track overflow, machine code format).

U Undefined-length.

UA Undefined-length (ANSI format).

UM Undefined-length (machine code format).

UT Undefined-length (track overflow).

UTA
Undefined-length (track overflow, ANSI format).

UTM
Undefined-length (track overflow, machine code format).

V Variable length.

VA
Variable length (ANSI format).

VB Variable length (blocked).

VBA
Variable length (blocked, ANSI format).

VBM
Variable length (blocked, machine code format).

VBS
Variable length (blocked, spanned).

VBSA
Variable length (blocked, spanned, ANSI format).

VBSM
Variable length (blocked, spanned, machine code format).

VBST
Variable length (blocked, spanned, track overflow).

VBSTA
Variable length (blocked, spanned, track overflow, ANSI format).

VBSTM
Variable length (blocked, spanned, track overflow, machine code format).

VBT
Variable length (blocked, track overflow).

VBTA
Variable length (blocked, track overflow, ANSI format).

VBTM
Variable length (blocked, track overflow, machine code format).

VM
Variable length (machine code format).

VS
Variable length (spanned).

VSA
Variable length (spanned, ANSI format).

VSM
Variable length (spanned, machine code format).

VT
Variable length (track overflow).

VTA
Variable length (track overflow, ANSI format).

VTM
Variable length (track overflow, machine code format).

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
DD01 DD RECFM=FB
```

Data class equivalent

Recfm

Related Enhanced Allocation Management function

SET_RECFM

Example

```
RECFM=V
```

RELEASESPACE

(Non-VSAM data sets.) The RELEASESPACE keyword specifies whether space release was specified.

Syntax

RELEASESPACE=YES | NO

Operands

Specify one of the following operands:

YES
Space release was specified.

NO Space release was not specified.

Note: The RELEASESPACE keyword requires additional calls to SMS Configuration Services and its presence in the RULEDEFS member can cause a measurable increase in CPU time for all tasks in the system, especially tasks that perform large numbers of allocations. Therefore, carefully consider the specification of the RELEASESPACE keyword in your rule definitions member.

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
DD01 DD SPACE=(TRK,(1,1),,RLSE))
```

Management class equivalent

Partial Release

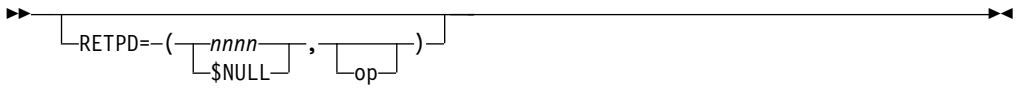
Example

```
RELEASESPACE=Y
```

RETPD

The RETPD keyword specifies the retention period for a data set.

Syntax



RETPD=(*nnnn* | $NULL,op)

Where:

nnnn is an integer (0 - 9999) that specifies the number of days to retain the data set.

Note: If you specify EXPDATE and RETPD, the EXPDATE keyword takes precedence.

Because the same internal field is used by the system to store the EXPDATE and RETPD value, if one or the other is specified in the JCL or DEFINE, the EXPDATE or RETPD keyword can be used as a selection criterion provided the proper date format is specified. For example, if you specify a RETPD of 1 and the creation date was 2008001 as expressed in EXPDATE format, the value of the field as stored internally by the system is 2008002. Therefore, either of the following selection criteria match:

```
RETPD=1
EXPDATE=2008002
```

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
DD01 DD RETPD=100
```

IDCAMS equivalent

```
DEFINE CLUSTER(FOR(200) ...)
```

Data class equivalent

Retpd or Expdt.

Related Enhanced Allocation Management function

SET_RETPD

Example

```
RETPD=365
```

REUSE

(VSAM data sets only.) The REUSE keyword specifies whether the VSAM REUSE option was specified.

Syntax

REUSE=YES | NO

Operands

Specify one of the following operands:

YES
REUSE was specified.

NO NOREUSE was specified.

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

IDCAMS equivalent

```
DEFINE CL (REUSE ....)
```

Data class equivalent

Reuse

Related Enhanced Allocation Management function

SET_REUSE

Example

```
REUSE=YES
```

SDISP

The SDISP keyword specifies the status disposition for a data set.

Syntax

SDISP=NEW | SHR | OLD | MOD

Operands

NEW
The specified or defaulted status disposition is DISP=(NEW).

SHR
The specified or defaulted status disposition is DISP=(SHR).

OLD
The specified or defaulted status disposition is DISP=(OLD).

MOD
The specified or defaulted status disposition is DISP=(MOD).

Notes:

1. The SDISP keyword does not apply to a VSAM IDCAMS DEFINE statement.
2. The system takes the following defaults with the status disposition parameter: If you omit the DISP parameter, the default disposition is NEW for the status disposition.

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
DD01 DD DISP=(NEW,,)
```

Example

```
SDISP=NEW
```

SHROPT_XRGN

(VSAM data sets only.) The SHROPT_XGRN keyword specifies the SHAREOPTIONS for cross-region sharing.

Syntax

SHROPT_XRGN=(*n*,op)

Where:

n is an integer (1 - 4) that specifies the VSAM cross-region shareoption.

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

IDCAMS equivalent

```
DEFINE CL (SHAREOPTIONS(1) ...)
```

Data class equivalent

Shareoptions Xregion

Related Enhanced Allocation Management function

SET_SHROPT_XRGN

Example

```
SHROPT_XRGN=1
```

SHROPT_XSYS

(VSAM data sets only.) The SHROPT_XSYS keyword specifies the SHAREOPTIONS for cross-system sharing.

Syntax

SHROPT_XSYS=(*n*,op)

Where:

n is an integer (3 or 4) that specifies the VSAM cross-system shareoption.

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

IDCAMS equivalent

```
DEFINE CL (SHAREOPTIONS(. 3) ...)
```

Data class equivalent

Shareoptions Xsystem

Related Enhanced Allocation Management function

SET_SHROPT_XSYS

Example

```
SHROPT_XSYS=3
```

SMS_MANAGED

The SMS_MANAGED keyword specifies whether the data set is under SMS control.

Syntax

SMS_MANAGED=YES|NO

Operands

Specify one of the following operands:

YES
The data set is SMS-managed.

NO The data set is not SMS-managed.

Note: If an SMS storage class is assigned to the data set before modification by Advanced Allocation Management, the data set is considered to be SMS-managed. If the SMS storage class is added or removed by Advanced Allocation Management in the ACS environment, subsequent environments will reflect an SMS management status of NO if the storage class was removed, or YES if a storage class was added.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

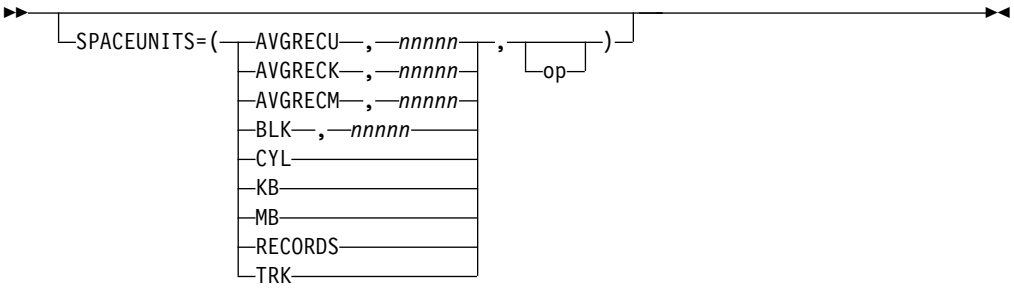
Example

```
SMS_MANAGED=YES
```

SPACEUNITS

The SPACEUNITS keyword specifies the allocation unit type for a data set.

Syntax



SPACEUNITS=(BLK | CYL | TRK | KB | MB | RECORDS | AVGRECU | AVGRECK | AVGRECM,*nnnnn*,op)

Operands

BLK
Space is allocated in blocks.

CYL
Space is allocated in cylinders.

TRK
Space is allocated in tracks.

KB Space is allocated in kilobytes.

MB Space is allocated in megabytes.

RECORDS
Space is allocated using the IDCAMS DEFINE CLUSTER RECORDS keyword.

AVGRECU
Space is allocated in records with AVGREC=U specified.

AVGRECK
Space is allocated in records with AVGREC=K specified.

AVGRECM
Space is allocated in records with AVGREC=M specified.

nnnnn
is an integer (0 - 65535) that specifies the block size or record size when allocations are made with the allocation space units of BLK or AVGRECU, AVGRECK, or AVGRECM respectively. This value does not apply to VSAM data sets.

Note: When a DEFINE is performed for a VSAM data set and space is specified using the RECORDS, KILOBYTES, or MEGABYTES parameters, IDCAMS

converts the specifications to tracks or cylinders and then saves the converted specification in the catalog. Therefore, if you specify a rule definition that checks SPACEUNITS for performing a function that occurs during VSAM end of volume (such as SEC_ALLOC), SPACEUNITS=AVGRECU, KB, or MB does not generate a match. This can affect PQTY (primary quantity) matching. For example, if a DEFINE specifies MB(1), the specification is converted to two cylinders and saved and then the value is saved in the catalog. Therefore, attempting to check PQTY=1 for a function that is performed during VSAM end-of-volume also causes a non-match to occur.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
DD01 DD SPACE=(8000,(2000,200))
```

IDCAMS equivalent

```
DEFINE CLUSTER(CYLINDERS(...) ....)
DEFINE CLUSTER(TRACKS(...) ....)
DEFINE CLUSTER(RECORDS(...) ....)
DEFINE CLUSTER(MEGABYTES(...) ....)
DEFINE CLUSTER(KILOBYTES(...) ...)
```

Related Enhanced Allocation Management function

SET_SPACEUNITS

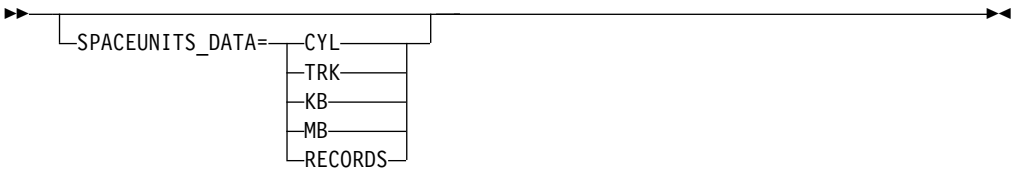
Example

```
SPACEU=CYL
```

SPACEUNITS_DATA

(VSAM data sets only.) The SPACEUNITS_DATA keyword specifies the allocation units for the data component of a VSAM data set.

Syntax



SPACEUNITS_DATA=(CYL | TRK | RECORDS | KB | MB)

Operands

CYL
Space is allocated in cylinders.

TRK
Space is allocated in tracks.

KB Space is allocated in kilobytes.

MB Space is allocated in megabytes.

RECORDS
Space is allocated in records.

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

IDCAMS equivalent

```
DEFINE CLUSTER(...) DATA(CYLINDERS(...) ....)
DEFINE CLUSTER(...) DATA(TRACKS(...) ....)
DEFINE CLUSTER(...) DATA(RECORDS(...) ....)
DEFINE CLUSTER(...) DATA(MEGABYTES(...) ....)
DEFINE CLUSTER(...) DATA(KILOBYTES(...) ...)
```

Related Enhanced Allocation Management function

SET_SPACEUNITS_DATA

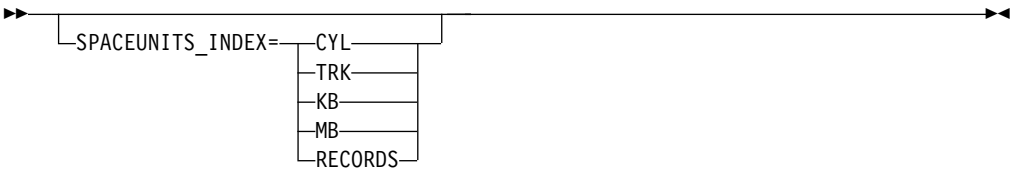
Example

```
SPACEUNITS_D=REC
```

SPACEUNITS_INDEX

(VSAM data sets only.) The SPACEUNITS_INDEX keyword specifies the allocation units for the index component of a VSAM data set.

Syntax



SPACEUNITS_INDEX=(CYL | TRK | RECORDS | KB | MB)

Operands

CYL
Space is allocated in cylinders.

TRK
Space is allocated in tracks.

KB Space is allocated in kilobytes.

MB Space is allocated in megabytes.

RECORDS
Space is allocated in records.

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

IDCAMS equivalent

```
DEFINE CLUSTER(...) INDEX(CYLINDERS(...) ....)
DEFINE CLUSTER(...) INDEX(TRACKS(...) ....)
DEFINE CLUSTER(...) INDEX(RECORDS(...) ....)
DEFINE CLUSTER(...) INDEX(MEGABYTES(...) ....)
DEFINE CLUSTER(...) INDEX(KILOBYTES(...) ...)
```

Related Enhanced Allocation Management function

SET_SPACEUNITS_INDEX

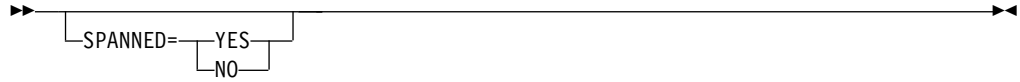
Example

```
SPACEUNITS_INDEX=REC
```

SPANNED

(VSAM data sets only.) The SPANNED keyword specifies whether the SPANNED option is defined for a VSAM data set.

Syntax



SPANNED=YES | NO

Operands

Specify one of the following operands:

YES
The SPANNED option was defined.

NO The NOSPANNED option was defined.

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

IDCAMS equivalent

```
DEFINE CLUSTER (SPANNED ...)
DEFINE CLUSTER (NOSPANNED ...)
```

Data class equivalent

Spanned / Nonspanned

Related Enhanced Allocation Management function

SET_SPANNED

Example

```
SPAN=Y
```

SQTY

The SQTY keyword specifies the secondary quantity value that is assigned to the data set.

Syntax

SQTY=(*nnnnnnnn*,op)

Where:

nnnnnnnn is an integer (0 - 16777215) that specifies the requested secondary quantity.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
DD01 DD SPACE=(...,(,2))
```

IDCAMS equivalent

```
DEFINE CLUSTER(CYLINDERS(... 1) ....)
DEFINE CLUSTER(TRACKS(... 1) ....)
DEFINE CLUSTER(RECORDS(... 1000) ....)
DEFINE CLUSTER(MEGABYTES(... 2) ....)
DEFINE CLUSTER(KILOBYTES(... 2048) ...)
```

Related Enhanced Allocation Management function

SET_SQTY

Example

```
SQTY=2048
```

SQTY_DATA

(VSAM data sets only.) The SQTY_DATA keyword specifies the secondary quantity value that is assigned to the data component of a VSAM data set.

Syntax

SQTY_DATA=(*nnnnnnnn*,op)

Where:

nnnnnnnn is an integer (0 - 16777215) that specifies the requested secondary allocation quantity for the data component in the requested allocation units.

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

IDCAMS equivalent

```
DEFINE CLUSTER(...) DATA(CYLINDERS(... 1) ....)
DEFINE CLUSTER(...) DATA(TRACKS(... 1) ....)
DEFINE CLUSTER(...) DATA(RECORDS(... 1000) ....)
DEFINE CLUSTER(...) DATA(MEGABYTES(... 2) ....)
DEFINE CLUSTER(...) DATA(KILOBYTES(... 2048) ...)
```

Related Enhanced Allocation Management function

SET_SQTY_DATA

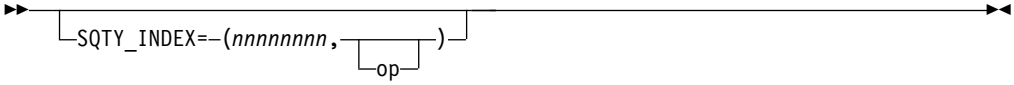
Example

```
SQTY_DATA=1000
```

SQTY_INDEX

(VSAM data sets only.) The SQTY_INDEX keyword specifies the secondary quantity value that is assigned to the index component of a VSAM data set.

Syntax



SQTY_INDEX=(*nnnnnnnn*,op)

Where:

nnnnnnn is an integer (0 - 16777215) that specifies the requested secondary allocation quantity for the index component in the requested allocation units.

Default

None.

Environments

ALLOCATE, EXTEND, and UNALLOCATE.

JCL equivalent

```
DEFINE CL ... INDEX(TRACKS(1) ...)
```

IDCAMS equivalent

```
DEFINE CLUSTER(...) INDEX(CYLINDERS(... 1) ....)
DEFINE CLUSTER(...) INDEX(TRACKS(... 1) ....)
DEFINE CLUSTER(...) INDEX(RECORDS(... 1000) ....)
DEFINE CLUSTER(...) INDEX(MEGABYTES(... 2) ....)
DEFINE CLUSTER(...) INDEX(KILOBYTES(... 2048) ...)
```

Related Enhanced Allocation Management function

SET_SQTY_INDEX

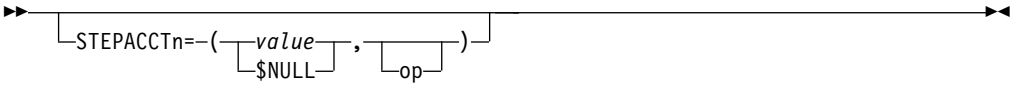
Example

```
SQTY_INDEX=1000
```

STEPACCTn

(VSAM and non-VSAM data sets.) The STEPACCTn keyword specifies one of the first fifteen Step Accounting field specifications.

Syntax



STEPACCTn=(*value* | $NULL,op)

Where:

n is an integer (1 - 15).

value is a character string with a maximum length of 144. If the string contains special characters or blanks, surround it with double quotation marks. Wildcard characters are permitted.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
STEP01 EXEC PGM=PAY01,ACCT=('accounting information')
```

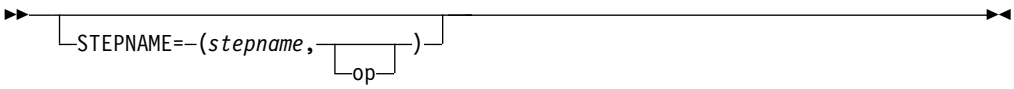
Example

```
STEPACCTn="Department 950"
```

STEPNAME

The STEPNAME keyword specifies the step name that is assigned to a job step.

Syntax



STEPNAME=(*stepname*,op)

Where:

stepname is the 1 - 8 character step name. Wildcard characters are permitted.

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
STEP01 EXEC PGM=PAYR01
```

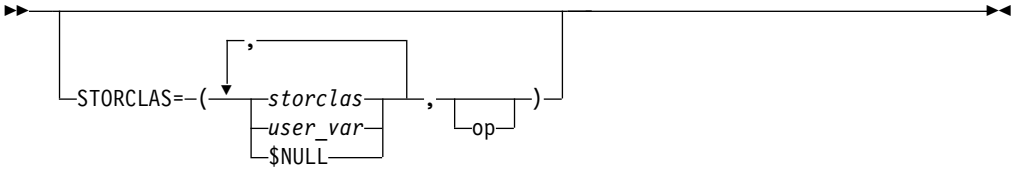
Example

```
STEPNAME=JJHSTEP
```

STORCLAS

The STORCLAS keyword specifies an SMS storage class that is assigned to an SMS data set.

Syntax



STORCLAS=(*storclas* | *user_var* | $NULL, *storclas* | *user_var* | $NULL,...,op)

Where:

storclas is the 1 - 8 character string that specifies the SMS storage class. Wildcard characters are permitted.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK

Note: During ACS processing, the value that is used is the value that is assigned by the user or the system before Advanced Allocation Management processing. In the other environments, the value that is used is the value that is assigned by Advanced Allocation Management (if specified); otherwise, it is the value that is assigned by the user or by the system.

JCL equivalent

```
DD01 DD STORCLAS=PAYRCLAS
```

IDCAMS equivalent

```
DEFINE CLUSTER(STORAGECLASS(PAYRCLAS) ...)
```

Related Enhanced Allocation Management function

SET_STORCLAS

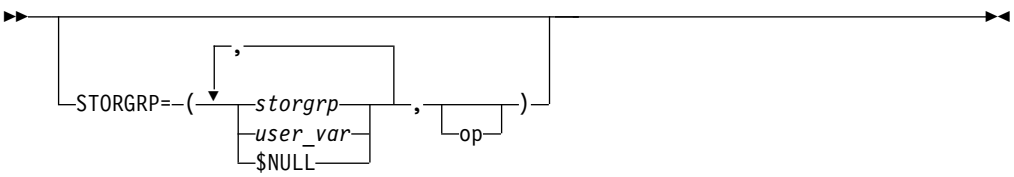
Examples

```
STORCLAS=TSO
STORCLAS=$NULL
STORCLAS=(TSO,S1P0,EQ)
STORCLAS=(TSO,S1P*,$NULL)
STORCLAS=(&CLASSES)
```

STORGRP

The STORGRP keyword specifies an SMS storage group that is assigned to an SMS data set.

Syntax



STORGRP=(*storgrp* | *user_var* | $NULL, *storgrp* | *user_var* | $NULL,...,op)

Where:

storgrp is a 1 - 8 character string that specifies the SMS storage group. Wildcard characters are permitted.

Note: With SMS, it is possible to define more than one eligible storage group for a new data set allocation. In this situation, if the Advanced Allocation Management STORGRP filter parameter is used, the specified storage group name is checked against all eligible storage groups for the allocation. If the specified name matches any of the eligible storage groups, then Advanced Allocation Management considers that portion of the rule definition to be a match. After a specific storage group is assigned to the data set, only that assigned storage group is checked against the storage group name that is specified by the STORGRP filter parameter.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

Note: During ACS processing, the value that is used is the value that is assigned by the user or the system before Advanced Allocation Management processing. In the other environments, the value that is used is the value that is assigned by Advanced Allocation Management (if specified) otherwise, it is the value that is assigned by the user or by the system.

Related Enhanced Allocation Management function

SET_STORGRP

Examples

```
STORGRP=STGLARGE
STORGRP=$NULL
STORGRP=(AUTO,BKUP,MIGR*,EQ)
STORGRP=(AUTO,BKUP,MIGR*,$NULL)
STORGRP=&GROUPS
```

SUBSYS

The SUBSYS keyword specifies a subsystem name that is assigned to an Advanced Allocation Management address space.

Syntax

SUBSYS=(*subsys_id*,op)

Where:

subsys_id is a 1 - 4 character value that specifies the Advanced Allocation Management subsystem ID (the value that is assigned in the OPTIONS member for the SSID parameter). Wildcard characters are permitted.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

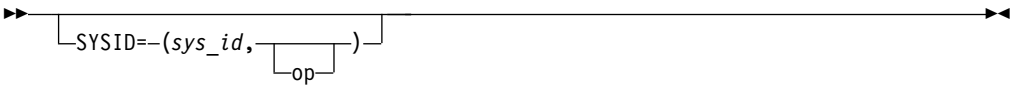
Example

```
SUBSYS=GL01
```

SYSID

The SYSID keyword specifies an OS/390 or z/OS system ID.

Syntax



SYSID=(*sys_id*,op)

Where:

sys_id is 1 - 4 character string that specifies the OS/390 or z/OS system ID. Wildcard characters are permitted.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
/*JOBPARM SYSAFF=ABCD
```

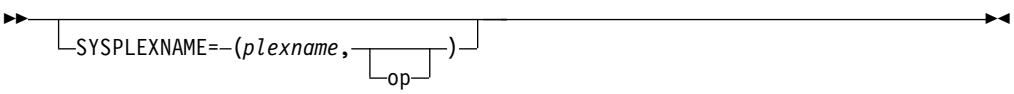
Example

```
SYSID=RS16
```

SYSPLEXNAME

The SYSPLEXNAME keyword specifies the z/OS name that is assigned to the sysplex.

Syntax



SYSPLEXNAME=(*plexname*,op)

Where:

plexname is a 1 - 8 character string that specifies the OS/390 or z/OS sysplex name. Wildcard characters are permitted.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

Example

```
SYSPLEXNAME=PLEX001
```

TAPE_COMP

(Tape data sets only.) The TAPE_COMP keyword specifies the defined data compaction setting for a tape data set.

Syntax

TAPE_COMP=YES | NO

Operands

Specify one of the following operands:

YES
TRTCH=COMP was specified.

NO TRTCH=NOCOMP was specified.

Default

None.

Environments

ALLOCATE, UNALLOCATE, and CATCHECK.

JCL equivalent

```
DD01 DD DCB=('TRTCH=COMP')
```

Data class equivalent

Compaction

Related Enhanced Allocation Management function

SET_TAPE_COMP

Example

```
TAPE_C=Y
```

TAPE_LABEL

(Non-VSAM tape data sets only.) The TAPE_LABEL keyword specifies the type of tape label that is used for a data set.

Syntax

TAPE_LABEL=SL | SUL | AL | AUL | NSL | NL | BLP | LTM

Operands

AL ISO/ANSI Version 1 or ISO/ANSI/FIPS Version 3 labels.

AUL
User labels and ISO/ANSI Version 1 or ISO/ANSI/FIPS Version 3 labels.

BLP
Bypass label processing.

LTM
Leading tapemarks.

NL Unlabeled tapes.

NSL
Non-standard labels.

SL Standard labels.

SUL
Standard and user labels.

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE, and CATCHECK.

JCL equivalent

```
DD01 DD LABEL=(,SL),...
```

Related Enhanced Allocation Management function

SET_TAPE_LABEL

Example

```
TAPE_L=SL
```

TEMPDS

The TEMPDS keyword specifies whether the data set is temporary.

Syntax

TEMPDS=YES | NO

Operands

Specify one of the following operands to indicate whether data set permanence should be evaluated:

YES
The data set must be a temporary data set for the rule definition to match.

NO The data set must not be a temporary data set for the rule definition to match.

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

Example

```
TEMPDS=YES
```

UNIT_AFFINITY

The UNIT_AFFINITY keyword specifies whether UNIT affinity is defined for a data set.

Syntax

UNIT_AFFINTY=YES | NO

Operands

Specify one of the following operands:

YES
UNIT affinity was specified.

NO UNIT affinity was not specified.

Note: This keyword does not apply to VSAM DEFINEs or SMS data sets.

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
DD01 DD UNIT=AFF=DD02
```

Related Enhanced Allocation Management function

SET_UNIT_AFFINITY

Example

```
UNIT_A=YES
```

UNIT_COUNT

The UNIT_COUNT keyword specifies the number of units that are assigned to the data set.

Syntax

UNIT_COUNT=(*nn*,op)

Where:

nn specifies the number of units (1 - 59) assigned to the data set.

Note: Do not use UNIT_COUNT for VSAM DEFINEs. UNIT_COUNT is valid for JCL or dynamically allocated data sets (including SMS and non-SMS VSAM and non-VSAM data sets). For IDCAMS DEFINEs, use the NUMVOLS keyword instead.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
DD01 DD UNIT=(SYSDA,3)
```

Related Enhanced Allocation Management function

SET_UNIT_COUNT

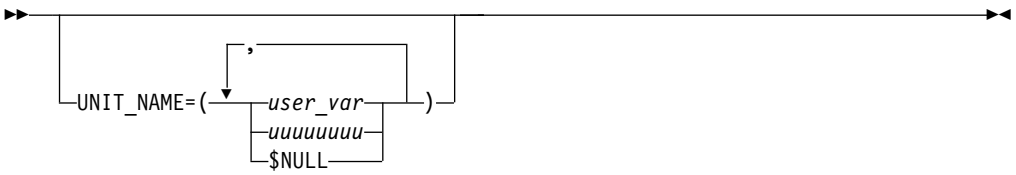
Example

```
UNIT_C=24
```

UNIT_NAME

The UNIT_NAME keyword specifies the generic or esoteric unit names.

Syntax



UNIT_NAME=(*uuuuuuuu* | *user_var* | $NULL, *uuuuuuuu* | *user_var* | $NULL,....)

Note: UNIT_NAME is only applicable to non-SMS, non-VSAM data sets. It does not apply to VSAM DEFINEs or SMS data sets. When a data set is allocated as DISP=NEW, the unit name that is used for matching is the one that is identified with the UNIT= JCL keyword or equivalent dynamic allocation text unit, or the one defaulted to by the system. For non-DISP=NEW allocations, the unit name is the generic unit name as identified in the catalog.

Where:

uuuuuuuu is the 1 - 8 character unit name. Wildcard characters are permitted.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
DD01 DD UNIT=SYSALLDA
```

Related Enhanced Allocation Management function

SET_UNIT_NAME

Examples

```
UNIT_NAME=SYSALLDA
UNIT_NAME=&UNAMES
```

UNIT_TYPE

The UNIT_TYPE keyword specifies the type of unit on which a data set is allocated.

Syntax

UNIT_TYPE=(OTHER | 3390 | 3380 | 9345 | 3420 | 3480 | 3490 | 3590 | ANY)

Notes:

1. For non-SMS data sets, the volume serial number must be specified in the JCL or IDCAMS DEFINE to determine the unit type for functions in the ACS and ALLOCATION environments. A non-match occurs if UNIT_TYPE is specified under such circumstances.
2. For SMS VSAM and non-VSAM data sets, because the volume serial number is not yet chosen by the system in the ACS and ALLOCATION environments, the unit type is not known, therefore if UNIT_TYPE is specified in the selection criteria, the rule definition statement is considered not to match.
3. For new SMS data set allocations that do not have a UNIT and VOLSER specification in the JCL or dynamic allocation request, UNIT_TYPE cannot be determined. If UNIT_TYPE is not known, any UNIT_TYPE selection criterion that is specified in a rule definition is considered a non-match and the rule definition is not selected. Therefore, do not use UNIT_TYPE as a sole selection criterion in environments where new SMS data sets are allocated without a UNIT and VOLSER specification.

Operands

OTHER
A unit type other than 3390, 3380, 9345, 3420, 3480, 3490, or 3590.

3390
3390 device.

3380
3380 device.

9345
9345 device.

3420
3420 device.

3480
3480 device.

3490
3490 device.

3590
3590 device.

ANY
Any unit type.

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

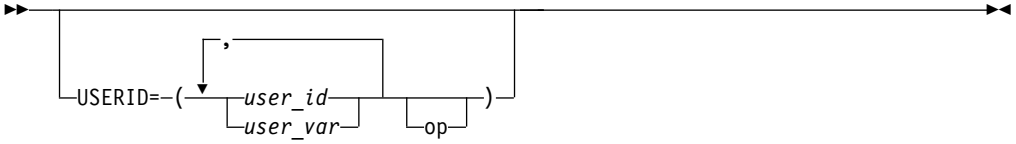
Example

```
UNIT_TYPE=3380
```

USERID

The USERID keyword specifies a user ID (such as a RACF or equivalent security product user ID).

Syntax



USERID=(*user_id* | *user_var*, *user_id* | *user_var*,...,op)

Where:

user_id is 1 - 7 character user ID. Wildcard characters are permitted.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
JJHTEST01 JOB ,,USER=DVLPJJH
```

Examples

```
USERID=PDSMITH
USERID=&IDLIST
USERID=(PDSMITH,TSJONES,QASMI*)
```

VOLCT

The VOLCT keyword specifies the number of volumes that are defined to the data set.

Syntax

VOLCT=(*nnn*,op)

Where:

nnn is an integer (1 - 255) that specifies the volume count.

Notes:

- For preexisting data sets, the value is considered to be the number of unique prime and candidate volumes for both the index and data components of a cluster.
- For DASD data sets, the maximum number of volumes is limited to 59. For tape data sets, the maximum number of volumes is limited to 255.
- Do not use VOLCT for VSAM DEFINEs. VOLCT is valid for JCL or dynamically allocated data sets (including SMS and non-SMS VSAM and non-VSAM data sets). For IDCAMS DEFINEs use the NUMVOLS keyword instead.

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
DD01 DD VOL=(,,,4,SER=VOL001)
```

Data class equivalent

Volume Count

Related Enhanced Allocation Management function

SET_VOLCT

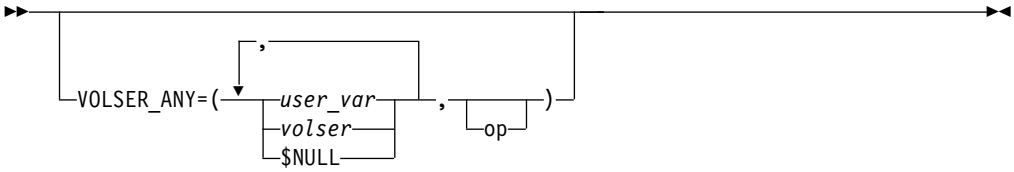
Example

```
VOLCT=124
```

VOLSER_ANY

The VOLSER_ANY keyword specifies any volume serial number.

Syntax



VOLSER_ANY=(*volser* | *user_var* | $NULL, *volser* | *user_var* | $NULL,...,op)

Where:

volser is the 1 - 6 character volume serial number. Wildcard characters are permitted.

Notes:

1. For non-SMS non-VSAM data sets, the volume serial number must be specified in the JCL (or dynamic allocation) for the VOLSER_ANY selection criterion to match for the PRIM_REDUCE function on the first volume.
2. For SMS VSAM and non-VSAM data sets, because the volume serial number is not yet chosen by the system at PRIM_REDUCE time, if VOLSER_ANY is specified in the selection criteria, the rule definition statement is considered not to match for the PRIM_REDUCE function on the first volume.

Default

None.

Environments

ACS, ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
DD01 DD VOL=SER=(BIG001,BIG002,...)
```

IDCAMS equivalent

```
DEFINE CLUSTER(VOLUMES(BIG001 BIG002 ...) ...)
```

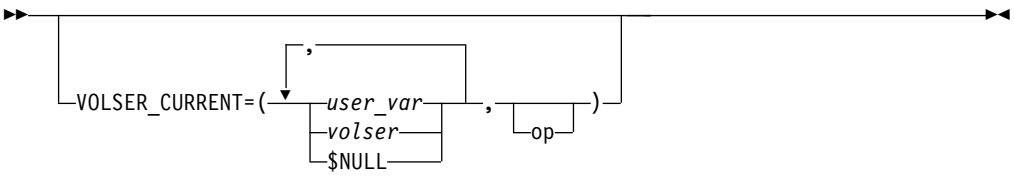
Examples

```
VOLSER_ANY=SMS001
VOLSER_ANY=&VSN
VOLSER_ANY=(SMS001,TST005,EQ)
```

VOLSER_CURRENT

The VOLSER_CURRENT keyword specifies a volume serial number of the current volume that is being processed when a particular function is being performed.

Syntax



VOLSER_CURRENT=(*volser* | *user_var* | $NULL, *volser* | *user_var* | $NULL,...,op)

Where:

volser is the 1 - 6 character volume serial number. Wildcard characters are permitted.

Notes:

1. For non-SMS non-VSAM data sets, the volume serial number must be specified in the JCL (or dynamic allocation) for the VOLSER_CURRENT selection criterion to match for the PRIM_REDUCE function on the first volume.
2. For non-SMS VSAM data sets, the first volume serial number in the DEFINE for the data set (in the index or data component) is used for matching purposes for the PRIM_REDUCE function on the first volume.
3. For SMS VSAM and non-VSAM data sets, because the volume serial number is not yet chosen by the system at PRIM_REDUCE time, if VOLSER_CURRENT is specified in the selection criteria with a value other than $NULL, the rule definition statement is considered not to match for the PRIM_REDUCE function on the first volume.

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
DD01 DD VOL=SER=(BIG001,BIG002,...)
```

IDCAMS equivalent

```
DEFINE CLUSTER(VOLUMES(BIG001 BIG002 ...) ...)
```

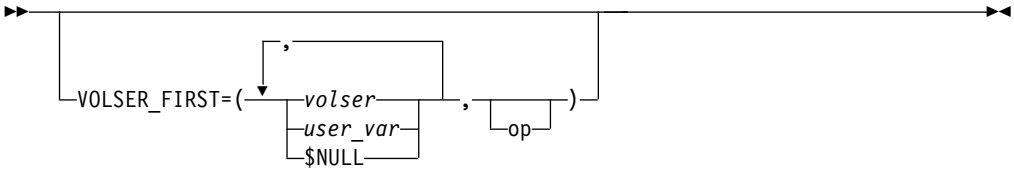
Examples

```
VOLSER_CURRENT=SMS001
VOLSER_CURRENT=&VSER
VOLSER_CURRENT=(SMS001,DEV002,WRK006,EQ)
```

VOLSER_FIRST

The VOLSER_FIRST keyword specifies a volume serial number of the first volume of the data set.

Syntax



VOLSER_FIRST=(*volser* | *user_var* | $NULL, *volser* | *user_var* | $NULL,...,op)

Where:

volser is the 1 - 6 character volume serial number. Wildcard characters are permitted.

Notes:

1. For non-SMS non-VSAM data sets, the volume serial number must be specified in the JCL (or dynamic allocation) for the VOLSER_FIRST selection criterion to match for the PRIM_REDUCE function on the first volume.
2. For non-SMS VSAM data sets, the first volume serial number in the DEFINE for the data set (in the index or data component) is used for matching purposes for the PRIM_REDUCE function on the first volume.
3. For SMS VSAM and non-VSAM data sets, because the volume serial number is not yet chosen by the system at PRIM_REDUCE time, if VOLSER_FIRST is specified in the selection criteria, the rule definition statement is considered not to match for the PRIM_REDUCE function on the first volume.

Default

None.

Environments

ALLOCATE, EXTEND, UNALLOCATE, STOW, and CATCHECK.

JCL equivalent

```
DD01 DD VOL=SER=BIG001
```

IDCAMS equivalent

```
DEFINE CLUSTER(VOLUMES(BIG001 ...) ...)
```

Examples

```
VOLSER_FIRST=SMS001
VOLSER_FIRST=&VNUMS
VOLSER_FIRST=(NS1600,NS1700,NS1800,NE)
```

VSAM_SEPARATE

(VSAM data sets only.) VSAM_SEPARATE specifies whether the index and data components of a VSAM data set are on separate volumes.

Syntax

VSAM_SEPARATE=YES|NO

Operands

Specify one of the following operands:

YES
Index and data components are on separate volumes.

NO Index and data components are *not* on separate volumes.

Default

None.

Environments

EXTEND and UNALLOCATE.

IDCAMS equivalent

```
DEFINE CL (...) DATA(VOL(VOL001) ...) INDEX(VOL(VOL002) ...)
```

Related Enhanced Allocation Management function

SET_VSAM_SEPARATE

Example

```
VSAM_SEP=Y
```

Comments

You can include comments with your syntax in two ways.

- If an asterisk is specified in column 1, the entire record is considered a comment.
- If an asterisk in any other column is preceded and followed by a blank, the area of the record from the asterisk to the end of the record is considered a comment. For example:

```
DEFAULTS PRIM_REDUCE=ENABLE
         PRIM_REDUCE_LIM=50
         PRIM_REDUCE_INC=10
*        Here is the first type of comment
INCLUDE  JOBNAME=ABCJOB2
INCLUDE  JOBNAME=ABCJOB3
         PRIM_REDUCE_INC=05
INCLUDE  JOBNAME=ABCJOB * Here is the second type of comment
```

Lists

A list of items such as a volume list, must be enclosed in parentheses.

If the list exceeds the limit of a single record, continue the list on subsequent records by ending the list on a record at a comma and then continuing the list on the next record. For example:

```
VOLSER=(DASD01,DASD02,DASD03,
        DASD04,DASD05,DASD06)
```

A set of keywords that are included within parentheses and separated by commas can be continued onto the next line. For example:

```
PRIM_REDUCE=(ENABLE,
                    ENABLE)
```

For information about specifying comparison operands with lists of items, see "Comparison operands and lists" on page 262.

Multiple operands

For keywords that have multiple operands, enclose the operands in parentheses.

Separate multiple operands for a keyword with a comma. Do not include spaces on either side of the equal sign that associates a keyword with its operands.

If you do not need to specify all of the operands for a keyword, use a comma to denote the missing operands. For example:

```
DEFAULTS PRIM_REDUCE=ENABLE
         PRIM_REDUCE_LIM=(50,50)
         PRIM_REDUCE_INC=(10,20)
INCLUDE JOBNAME=JOBABC12
PRIM_REDUCE=(,ENABLE)
```

Note: If you specify only the first operand and no other subsequent operands, you do not need to enclose the operand in parentheses or include trailing commas to denote the missing operands.

Spaces between operands that are separated by commas are ignored. Therefore, the following RULEDEFS specifications are equivalent:

```
PRIM_REDUCE=(ENABLE,     ENABLE)
PRIM_REDUCE=(ENABLE,ENABLE)
```

Wildcard characters

The following wildcard characters are supported where indicated in the syntax descriptions.

Table 10. Supported wildcard characters

Wildcard character	Description
Asterisk (*)	Indicates zero or more characters.
Percent sign (%)	Indicates a single matching character.

Job name filtering

Use wildcard characters to specify select job names.

Consider the following example:

```
DEFAULTS PRIM_REDUCE=ENABLE
         PRIM_REDUCE_LIM=50
         PRIM_REDUCE_INC=10
 *
```

```
INCLUDE  JOBNAME=ABC%%%%2
INCLUDE  JOBNAME=JHH*
INCLUDE  JOBNAME=JOB1*TST
         PRIM_REDUCE_INC=05
```

The `INCLUDE JOBNAME=ABC%%%%2` statement indicates that all job names that begin with ABC, followed by four characters, and ending with 2 (for example, ABCTEST2 or ABC12342) are to be included in processing.

The `INCLUDE JOBNAME=JHH*` statement indicates that all job names that begin with JHH such as JHHTEST, JHHABC, JHH1, and so on, are to be included in processing.

The `INCLUDE JOBNAME=JOB1*TST` statement indicates that all job names that begin with JOB1 and end with TST such as JOB1ATST or JOB15TST are to be included in processing.

Data set name filtering

To implement data set name filtering for the selection criteria DSNAME, DSNAME_DATA, DSNAME_INDEX, and LOGSTREAMID, specify the DSN_PATTERN_MATCHING keyword in the OPTIONS member.

For information about the OPTIONS member global definitions and general operation options, see "Specifying subsystem options" on page 29.

DSN_PATTERN_MATCHING wildcard character patterns

Table 11 shows valid wildcard character patterns and examples for the DSN_PATTERN_MATCHING options.

Table 11. DSN_PATTERN_MATCHING wildcard character patterns and examples

Wildcard character specification	Description	Examples

*	An asterisk that is combined with valid DSN characters, or a percent sign (%), represents zero to eight characters of any value.	`ABC.DEF*`
		
		This specification matches all data set names that have ABC as the first qualifier and the characters DEF at the beginning of the second qualifier. For example: ABC.DEF, ABC.DEF3, or ABC.DEFINE1.
		
		`ABC.*DEF`
		
		This specification matches all data set names that have ABC as the first qualifier and the characters DEF at the end of the second qualifier. For example: ABC.ADEF or ABC.A123DEF.
		
		`ABC.*D%`
		
		This specification matches all data set names that have ABC as the first qualifier, the character D in the second qualifier, followed by one non-blank DSN character. For example: ABC.ADE or ABC.JOBD1.

Table 11. DSN_PATTERN_MATCHING wildcard character patterns and examples (continued)

Wildcard character specification	Description	Examples

. or ..	Represents one or more valid characters when BASIC is specified.
	
	Represents one DSN qualifier when EXTENDED1 or EXTENDED2 is specified.
		`*.DEF.JOB`
		
		When BASIC is specified, this specification matches on data set names that have *any valid leading qualifiers* where the last two qualifiers of the data set name are DEF.JOB. For example: ABC.DEF.JOB, WORK1.DEF. JOB, or TEST1.ABC.DEF. JOB. This specification does not match on a data set name of DEF.JOB.
		
		When EXTENDED1 or EXTENDED2 is specified, this specification matches on data set names that have *any valid first qualifier* where the last two qualifiers are DEF.JOB (such as TST3.DEF.JOB or ABC.DEF.JOB).
		Note: This specification does not match on data set names such as DEF.JOB, ABC.TST3.DEF.JOB, or WORK1.TEST5.AUD.DEF. JOB.
		
		`ABC.DEF.*.JOB`
		
		When BASIC is specified, this specification matches on data set names that have ABC as the first qualifier, DEF as the second qualifier, *one or more qualifiers*, and JOB as the last qualifier. For example: ABC.DEF.GHI.JOB or ABC.DEF.TST3.T52.JOB.
		
		When EXTENDED1 or EXTENDED2 is specified, this specification matches on data set names that have ABC as the first qualifier, DEF as the second qualifier, *any valid third qualifier*, and JOB as the last qualifier. For example: ABC.DEF.TST3.JOB or ABC.DEF.GHI.JOB.
		Note: This specification does not match on data set names such as ABC.DEF.JOB or ABC.DEF.TST3.T52.JOB.

.* last qualifier	Represents one or more trailing DSN qualifiers when BASIC is specified. Represents one DSN qualifier when EXTENDED1 is specified. Represents zero or more DSN qualifiers when EXTENDED2 is specified.	`ABC.DEF.*` When BASIC is specified, this specification matches data set names that have ABC as the first qualifier, DEF as the second qualifier, *and one or more trailing qualifiers*. For example: ABC.DEF.TST3, ABC.DEF.GHI.JOB, or ABC.DEF.TST3.JOB.T52.ADDR5. **Note:** This specification does not match on the data set name ABC.DEF. When EXTENDED1 is specified, this specification matches all data set names that have ABC as the first qualifier, DEF as the second qualifier, *and any valid third qualifier*. For example: ABC.DEF.TST3 or ABC.DEF.JOB. **Note:** This specification does *not* match on the data set name ABC.DEF or data set names with more than three qualifiers. When EXTENDED2 is specified, this specification matches data set names that have ABC as the first qualifier and DEF as the second qualifier, including data set names with *zero or more trailing qualifiers*. For example: ABC.DEF, ABC.DEF.TST3, ABC.DEF.GHI.JOB, or ABC.DEF.TST3.JOB.T52.ADDR5.

Table 11. DSN_PATTERN_MATCHING wildcard character patterns and examples (continued)

Wildcard character specification	Description	Examples
. or .. or .**	Represents zero or more DSN qualifiers. **Note:** The BASIC option does not support the use of double asterisks.	**.DEF.JOB When EXTENDED1 or EXTENDED2 is specified, this specification matches on data set names that end with the qualifiers of DEF and JOB. For example: DEF.JOB, ABC.DEF.JOB, ABC.TST3.DEF. JOB, or T52.TST3.WORK.DEF. JOB. ABC.DEF.**.JOB When EXTENDED1 or EXTENDED2 is specified, this specification matches on data set names that have ABC as the first qualifier, DEF as the second qualifier, zero or more qualifiers, and JOB as the last qualifier. For example: ABC.DEF.JOB or ABC.DEF.TST3.JOB. ABC.DEF.** When EXTENDED1 or EXTENDED2 is specified, this specification matches data set names that have ABC as the first qualifier and DEF as the second qualifier. For example: ABC.DEF, ABC.DEF.TST3, or ABC.DEF.TST3.JOB.T52.ADDR5.
%	Represents one non-blank DSN character.	ABC.D%%%D This specification matches all data sets that have ABC as the first qualifier and a second qualifier that begins with D, followed by three characters, ending with D. For example: ABC.D123D or ABC.DABCD.

Rule definition example (DSN_PATTERN_MATCHING=BASIC)

The following example applies to default Advanced Allocation Management processing as specified by the DSN_PATTERN_MATCHING=BASIC subsystem option.

```
INCLUDE DSNAME=WRKABC*
INCLUDE DSNAME=WORK.ABC*.IJK
INCLUDE DSNAME=PABC.CDEF.*
INCLUDE DSNAME=NODE1.NO*E4
```

The `INCLUDE DSNAME=WRKABC*` statement indicates that all data sets that have names that begin with WRKABC (such as WRKABC, WRKABCTS, WRKABC1.TEST, WRKABC2.TST.DSK1) are to be included in processing:

The `INCLUDE DSNAME=WORK.ABC*.IJK` statement indicates that all data sets that have names that begin with WORK.ABC and end with .IJK (such as WORK.ABC.IJK, WORK.ABC123.IJK or WORK.ABCTST1.DSK1.IJK) are to be included in processing.

The `INCLUDE DSNAME=PABC.CDEF.*` statement indicates that all data sets that have names that begin with PABC.CDEF. (such as PABC.CDEF.TST or PABC.CDEF.TEST.WRK1) are to be included in processing. PABC.CDEF is not included in processing.

The `INCLUDE DSNAME=NODE1.NO*E4` statement indicates that all data sets that have names that begin with NODE1.NO and end with E4 (such as

NODE1.NOABC.NODE4, or NODE1.NODE2.NODE3.NODE4) are to be included in processing.

Wildcard characters and string comparisons

You can use wildcard characters with string comparisons.

Consider the following example:

```
INCLUDE DSNAME=(WORK.ABC*.IJK,EQ)
```

`INCLUDE DSNAME=WORK.ABC*.IJK` indicates that all data sets that have names that begin with WORK.ABC and end with .IJK (such as `WORK.ABC123.IJK` or `WORK.ABCTST1.DSK1.IJK`) are to be included in processing.

However, if a comparison other than EQ or NE is made against a string that contains generic characters (* or %), the comparison is made only against the character string up to, and including, the character that immediately precedes the generic character. For example:

```
INCLUDE DSNAME=(WORK.*.IJK,GT)
```

In the example, a data set name string that has a value greater than "WORK." is included in processing.

For more information, see "String comparisons" on page 262 and "Comparison operands."

Character string values

All character string values can contain generic characters unless otherwise noted.

Comparison operands

Many selection criteria keywords allow you to specify the following optional comparison operands.

EQ (Default.) Equal.

NE Not equal.

GE Greater than or equal.

LE Less than or equal.

GT Greater than.

LT Less than.

For example:

```
JOBNAME=(PAYROLL3,GT)
```

In the example, a match occurs if the JOBNAME is greater than the string "PAYROLL3".

The following example shows how to use the greater than (GT) comparison operand to specify a retention period that is greater than a specific number of days:

```
INCLUDE
     RETPD=(5,GT)
```

In the example, when the specified retention period is greater than five days, the definition is considered a match.

Note: If you do not specify a comparison operand for a keyword that allows comparison operands, an equal comparison (EQ) is assumed.

String comparisons

Advanced Allocation Management performs string comparisons using the EBCDIC character set. Strings of unequal length are assumed to be padded with blanks.

If a comparison other than EQ or NE is made against a string that contains generic characters such as * or %, the comparison is made only against the character string up to, and including, the character that immediately precedes the generic character. For more information, see "Wildcard characters and string comparisons" on page 261.

Numeric comparisons

Comparisons can be performed for numeric values.

The following example is considered a match if the BLKSIZE is greater than 4096.

```
BLKSIZE=(4096,GT)
```

With numeric values, a numeric comparison is performed. Therefore, leading zeros are ignored. For example, 4096, 04096, and 00004096 are considered to be equal.

Comparison operands and lists

Some selection criteria keywords allow the specification of a list of items and a comparison operand. The following topics describe the comparison operand logic when specified with a list, such as a list of volumes.

EQ

For the EQ comparison operand, the selection criteria is met if *any* of the operands in the list are equal (OR logic). This does not apply to user-defined variables. See "Comparison operands and user-defined variables" for information about user-defined variables and comparison operators.

Consider the following example:

```
USERID=(TSJRH,PDSMITH,PDJONES,EQ)
```

In the example, if a user ID matches *any* of the three specified user IDs in the list, then the result is true.

GE, LE, GT, NE, and LT

For all comparison operands other than EQ, the selection criteria is met only if *all* of the operands in the list satisfy the comparison operand (AND logic). For example, in the following specification the result is true only if the volsers are greater than *all* three volumes that are specified in the list:

```
VOLSER_FIRST=(NS1600,NS1700,NS1800,GT)
```

Comparison operands and user-defined variables

Do not place comparison operands on a variable definition VALUE keyword. Doing so causes the comparison operands to be treated as another keyword substitution value. To be treated as comparison operands, specify them only on rule definition keywords that support comparison operands.

Comparison operands are allowed in lists that support them, including lists that contain variables. However, comparison operands that are specified within a variable are treated as another substitution value in the list. Consider the following variable definition specification:

```
VARIABLE_NAME=&X
VALUE=(TSJRHA,TSJRHB,NE)
```

In the example, NE is treated as a substitution value, and not as a comparison operand.

To be treated as a comparison operand, you must specify NE within a rule definition keyword that supports comparison operands. The following example shows how to correct the previous example.

In the variable definition, specify:

```
VARIABLE_NAME=&X
VALUE=(TSJRHA,TSJRHB)
```

In the rule definition specify:

```
INCLUDE JOBNAME=(TSJRH,&X,NE)
```

Numeric suffixes

To designate a scale, append a suffix to selection criteria and function numeric values.

Choose from the following suffixes:

Table 12. Numeric suffixes for selection criteria and function numeric values

Suffix	Description
K	Kilobytes (value * 1024)
M	Megabytes (value * 1048576)
G	Gigabytes (value * 1073741824)
T	Terabytes (value * 1099511627776)

For example, 4K represents the numeric value 4096. The suffix counts toward the maximum allowed length of the field. Therefore, when you specify 4K as the value for a keyword, it is considered to have a length of 2.

Null or unspecified values ($NULL)

For some selection criteria and Enhanced Allocation Management function keywords, the reserved string ($NULL) designates that a null or unspecified value exists for the particular item.

For example, in the following DD statement, a selection criterion of ADISP=$NULL is considered to be a match because the abnormal disposition was not specified using the JCL DD DISP keyword:

```
//DD1 DD DSNAME=PAY.FILE,DISP=(,CATLG)
```

Note: For selection criteria keywords, use only the EQ and NE comparison operands with $NULL. Other comparison operands are ignored.

For selection criteria processing, $NULL indicates that an allocation attribute was not specified for a JCL, DYNALLOC, or DEFINE statement. Specifying $NULL allows you to set a value if no value was specified. For example:

```
INCLUDE
     DEVICECLASS=TAPE
     RETPD=$NULL
     SET_RETPD=1
```

In the example, when an allocation is directed to tape, if no retention period is specified, then set it to one day.

ACS environment considerations

Automatic Class Selection (ACS) environment considerations are applicable to the DATACLAS, MGMTCLAS, STORCLAS, and STORGRP selection criteria and the SET_DATACLAS, SET_MGMTCLAS, SET_STORCLAS, and SET_STORGRP functions.

SMS can make multiple calls to the ACS environment to determine the four SMS constructs of the data, storage and management classes and the storage group. For example, SMS might make an initial call to determine the data, storage and management classes and then make a second call to establish the storage group. When you use an SMS construct as a selection criterion, it is only considered a match if the particular selection criterion was requested in the call to the ACS environment.

For example, if ACS makes a call to determine the data class and a rule definition contains selection criteria for both data and storage classes, even if the data class matches the rule definition selection criterion, because a storage class selection criterion was specified, but the storage class was not requested in the ACS call, the rule definition is considered a non-match.

Note: SMS makes a single ACS call for all JCL-defined data sets before the start of the first step of the job. So, messages that are associated with changes to the Data Class, Storage Class, Management Class and Storage Group using the SET_DATACLAS, SET_STORCLAS, SET_MGMTCLAS, and SET_STORGRP Enhanced Allocation Management functions respectively, are displayed at the beginning of the job ahead of the execution of the first step.

Similarly, after a match is made on the selection criteria in a rule definition, the only SMS constructs that can be set with Advanced Allocation Management are the constructs that were originally requested in the call to the ACS environment. For example, if only the data class is requested in the call to the ACS environment, only the SET_DATACLAS Enhanced Allocation Management function is accepted. Specified SET_STORCLAS, SET_MGMTCLAS, or SET_STORGRP functions are ignored.

Space parameter considerations

The same rules that apply during normal z/OS and SMS processing also apply when you use Advanced Allocation Management to change an allocation variable.

For example, IDCAMS does not permit space parameters to be specified concurrently at both the cluster and data or index levels. Therefore, to set space parameters at the cluster level, add SET statements (Enhanced Allocation Management function keywords) to nullify them at the data and index levels, and vice versa.

For example, if you specify the following statements:

```
SET_SPACEUNITS_DATA=CYL
SET_PQTY_DATA=2
SET_SQTY_DATA=2

SET_SPACEUNITS_INDEX=CYL
SET_PQTY_INDEX=1
SET_SQTY_INDEX=1
```

also include the following SET statements:

```
SET_SPACEUNITS=$NULL
SET_PQTY=$NULL
SET_SQTY=$NULL
```

Note: To nullify space parameters for a particular level, you must set all of the variables (SET_SPACEUNITS, SET_PQTY, and SET_SQTY) to null.

Consistency rules for space parameters (batch or dynamic allocation)

Follow these consistency rules for space parameters for batch or dynamic allocation (VSAM and non-VSAM data sets):

- If SPACEUNITS, PQTY, or SQTY are implicitly or explicitly set to be overridden by the dataclas ($DCO) then:
  - all of those parameters must be set to be overridden by the dataclas

    or
  - AVGREC and average-record-length (a subparameter of the SPACE keyword) must have originally been specified for the allocation.

  For example, if `SPACE=(TRK,1000)` is coded in the JCL and you specify `SET_SPACEUNITS=$DCOC` in the rule definitions, a consistency check failure message is issued and the SET operation is not performed. This is because a dataclas can only specify space parameters in AVGREC units and if `AVGREC=M` is specified in the dataclas, an invalid allocation of (average_record_length * 1000 * 1000000) would result.

  However, if `SPACE=(TRK,1000)` is coded in the JCL and you specify `SET_SPACEUNITS=$DCOC` in the rule definitions, and you also specify `SET_PQTY=$DCOC` and `SET_SQTY=$DCOC`, then space quantities that are consistent with AVGREC from the dataclas are used, resulting in a meaningful allocation.
- If SPACEUNITS, PQTY, or SQTY are implicitly or explicitly set to be overridden by the dataclas (dataclas-override (DCO)), then AVGVALUE, PQTY, and SQTY must be defined in the dataclas, or none must be defined in the dataclas.
- If SPACEUNITS, PQTY, or SQTY are set to $NULL, then all of those parameters must be set to $NULL.
- If SPACEUNITS is explicitly changed to a non-$DCO ($DCOU or $DCOC), non-$NULL value, then a new corresponding PQTY value must also be specified (unless the original PQTY value was zero).
- If SPACEUNITS is explicitly changed to a non-$DCO ($DCOU or $DCOC), non-$NULL value, then a new corresponding SQTY value must also be specified (unless the original SQTY value was zero).
- If SPACEUNITS is implicitly or explicitly set to AVGRECU, AVGRECK, or AVGRECM, the record-size must also be specified.

Consistency rules for space parameters (VSAM DEFINEs)

Follow these consistency rules for space parameters for VSAM defines:

- If SPACEUNITS, PQTY, or SQTY are implicitly or explicitly set to be overridden by the dataclas ($DCO), then all of those parameters must be set to be overridden by the dataclas.
- If SPACEUNITS, PQTY, or SQTY are implicitly or explicitly set to be overridden by the dataclas (dataclas-override (DCO)), then AVGVALUE, PQTY, and SQTY must be defined in the dataclas, or none must be defined in the dataclas.
- If SPACEUNITS, PQTY, or SQTY are set to $NULL, then all of those parameters must be set to $NULL.
- If SPACEUNITS is explicitly changed to a non-$DCO ($DCOU or $DCOC), non-$NULL value, then a new corresponding PQTY value must also be specified (unless the original PQTY value was zero).
- If SPACEUNITS is explicitly changed to a non-$DCO ($DCOU or $DCOC), non-$NULL value, then a new corresponding SQTY value must also be specified (unless the original SQTY value was zero).
- If SPACEUNITS is implicitly or explicitly set to AVGRECU, AVGRECK, or AVGRECM, the average record size must also be specified.

Chapter 6. Volume group definitions

The volume group definitions member (VGRPDEFS) in the Advanced Allocation Management control data set contains the list of volume serial numbers that are associated with a particular volume group name. You use the volume group name and the VOL_ADD_VGRP keyword to define a group of volumes that Advanced Allocation Management uses to satisfy a volume addition as directed by the VOL_ADD keyword.

Note: This functionality applies only to non-SMS volume extends.

When you issue the **ACTIVATE** command, the contents of the specified member becomes the active set of definitions that Advanced Allocation Management uses during processing. The following messages, which are displayed in the JES SYSMSG output, show the successful activation of a volume group definitions member:

```
GLO4199I      1  * VGRP DEFINITIONS
GLO4199I      2  GROUP_NAME=JJHGRP01
GLO4199I      3  VOLUMES=(VOL000,VOL001,VOL002,VOL003,VOL004,VOL005)
GLO4109I VGRP DEFINITIONS SUCCESSFULLY ACTIVATED -AOPT
```

If an error occurs during the activation, messages, JES SYSMSG displays messages to help determine the cause of the error.

Creating volume group definitions

To create a volume group definitions member, complete these steps.

1. Make a copy of the sample volume group definitions member (VGRPDEFS), which is in the control data set that was generated during the installation and configuration process.
2. Modify the new (copied) volume group definitions member. To modify the member, use the options on the Tivoli Advanced Allocation Management Control Member Management panel (preferred) or use the ISPF EDIT function.
   a. Confirm that the first line of the member is the comment:
      ```
      * VGRP DEFINITIONS
      ```
   b. Specify the volume group names and the volume serial numbers to associate with the volume group name. For more information, see "Volume group definition syntax" on page 268.
3. Use a valid member name to save the member. All control data set member names must adhere to standard PDS member name conventions.

Activating volume group definitions

To activate the volume group definitions, issue the **ACTIVATE** operator command.

Issue the **ACTIVATE** operator command whenever the Advanced Allocation Management started task is running. Use the following format:
MODIFY *stcname*,ACTIVATE VGRPDEFS *member*
Where *stcname* is the name of the started task, and *member* is the name of the volume group definitions member.
For example: `ACTIVATE VGRPDEFS VOLGRP2`

Note: If you do not specify a member name on the **ACTIVATE** command, the member name VGRPDEFS is used.

Displaying volume group definitions

The **DISPLAY** command displays information about the active set of volume group definitions, including the member name from which they were activated and the date and time the member was activated.

To display the active volume group definitions, issue the following command from the operator console:

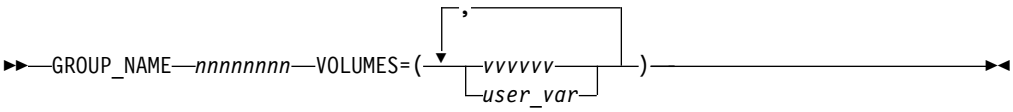
MODIFY *stcname*,DISPLAY VGRPDEFS
Where *stcname* is the name of the started task.

Note: You can display a volume group list member only if you previously activated it.

Volume group definition syntax

Use volume group definition syntax to define volume groups. The volume group definition syntax consists of two keywords: GROUP_NAME and VOLUMES.

Note: Volume group definitions do not apply to SMS data sets.



In the keyword descriptions, the underlined portion of the keyword indicates the minimally acceptable abbreviation.

For volume group definition syntax, keywords can be specified in columns 1 - 72 inclusive. The maximum record length for the Advanced Allocation Management control data set is 80 bytes. Except for lists of items, a keyword and its operands must be contained within a single record.

GROUP_NAME

The GROUP_NAME keyword defines a volume group.

Syntax

GROUP_NAME=*nnnnnnnn*

Where *nnnnnnnn* is a maximum of eight alphanumeric characters to define a volume group.

Example

```
GROUP_NAME=JJHGRP01
```

VOLUMES

The VOLUMES keyword defines the list of volumes in the volume group.

Syntax

VOLUMES=(*vvvvvv* | *user_var*,*vvvvvv* | *user_var*,.....)

Where *vvvvvv* is a six-character volume serial number, and *user_var* is a user-defined variable. Multiple volumes and user variables can be listed. Wildcard characters are permitted.

Tip: Specifying specific volumes rather than generic volumes might result in improved processing speed.

Examples

In the following example, VOL001,VOL002,VOL003 are defined as the list of volumes in the volume group:

```
VOLUMES=(VOL001,VOL002,VOL003)
```

In the following example, &A_VOLS,&B_VOLS are defined as the list of volumes in the volume group:

```
VOLUMES=(&A_VOLS,&B_VOLS)
```

Single volume group list example

In the following example, a single volume group (GROUP01) list contains volumes VOL001 - VOL004, and volumes that begin with the letter "D".

```
* VGRP DEFINITIONS
 GROUP_NAME=GROUP01
 VOLUMES=(VOL001,VOL002,VOL003,VOL004,
          D*)
```

Note: When you list multiple values for an operand such as a volume list, you must enclose the list in parentheses. If the list exceeds the limit of a single record, continue the list on subsequent records by ending the list on a record with a comma and then continuing the list on the next record.

The following example shows a copy of the VGRPDEFS member that was modified to define the volume group JJHGRP01:

```
* VGRP DEFINITIONS

GROUP_NAME=JJHGRP01                          * NAME OF THE VOLUME GROUP

      VOLUMES=(VOL000,VOL001,VOL002,VOL003,  * <--- CHANGE TO APPROPRIATE
      VOL004,VOL005)                         *      VOLUME SERIAL NUMBERS
```

The GROUP_NAME keyword defines the name of the volume group. The VOLUMES keyword defines the list of the volumes in the group.

For an example of using a user-defined variable to specify a volume group list, see “User-defined variable examples” on page 273.

Multiple volume group lists example

The following example shows the specification of three volume groups (GROUP01, GROUP02, and GROUP03).

```
 * VGRP DEFINITIONS

GROUP_NAME=GROUP01

VOLUMES=(VOL00A,VOL00B,VOL00C,VOL00D,
VOL00E,VOL00F,VOL00G,VOL00H,
VOL00I,VOL00J,VOL00K,VOL00L)

GROUP_NAME=GROUP02

VOLUMES=(VOL00M,VOL00N,VOL00O,VOL00P,
VOL00Q,VOL00R,VOL00S,VOL00T,
VOL00U,VOL00V,VOL00W,VOL00X)

GROUP_NAME=GROUP03

VOLUMES=(VOL01A,VOL02B,VOL03C,VOL04D,
VOL05E,VOL06F,VOL07G,VOL08H,
VOL09I,VOL10J,VOL11K,VOL12L)
```

Chapter 7. User-defined variables

To further customize Advanced Allocation Management processing, you can create user-defined variables that can be specified on select keywords.

The user-defined variable definitions member (VARDEFS) contains variable definitions that are used as substitution values that can be specified on select rule definition and volume group definition keywords as noted by *user_var* in the syntax diagrams. When specified, they are incorporated into the selection criteria for the specified rule or volume group definitions when the rule definitions or volume group definitions are activated.

After you define variables in a VARDEFS member and activate rule or volume group definitions, the following actions occur:

- Variables are incorporated into the appropriate rule definitions and volume group definitions.
- Advanced Allocation Management searches the current VARDEFS member for a matching variable name, and inserts the value of the variable into the rule. If the variable is not found, or if a VARDEFS member syntax error occurs, activation fails.

Creating user-defined variable definitions

To create a user-defined variable definitions member, complete the following steps.

1. Create a copy of the sample VARDEFS member, which is in the control data set that the installation and configuration process generates.
2. Modify the copy. Use the options on the Tivoli Advanced Allocation Management Control Member Management panel (preferred), or use the ISPF EDIT function.
   a. Verify that the first line of the member is the comment:

      ```
      * VAR DEFINITIONS
      ```

   b. Create user-defined variables. A user-defined variable consists of a variable name that begins with an ampersand. Define the variable using the VARIABLE_NAME= keyword, followed by a VALUE= *keyword* statement (where *keyword* can be 1 - 44 characters). When the RULEDEFS or VGRPDEFS are activated, the variable must satisfy the syntax requirements of the *keyword*. For more information, see "User defined variable syntax" on page 272.

      Note:

      - User-defined variables cannot be embedded in other variables.
      - Do not place comparison operands on a VARDEFS member VALUE keyword. Doing so causes the operands to be treated as another keyword substitution value. To be treated as comparison operands, specify them only within rule definition keywords that support comparison operands. For more information, see "Comparison operands and user-defined variables" on page 262.
3. Save the member using a valid one to eight-character member name. All control data set member names must adhere to standard PDS member name conventions.

After you define the variable in the variable definitions member, the variable can be specified on select keywords as noted in the syntax descriptions.

4. To specify the variable definitions to use when the rule definitions or volume group definitions are activated, issue the Advanced Allocation Management operator command **SETVARDEFS** from a z/OS console. For more information, see "Specifying the VARDEFS member."

Specifying the VARDEFS member

The Advanced Allocation Management operator command **SETVARDEFS** specifies the variable definitions to use when the rule definitions or volume group definitions are activated.

To specify the variable definitions to use when the rule definitions or volume group definitions are activated, complete the following step:

Issue the following operator command from a z/OS console:
`F glostc,SETVARDEFS membername`
Where *membername* is the name of the variables definitions member to use.

Displaying user-defined variable definitions

Display information about the current variable definitions, including the member name and the date and time that the member was defined.

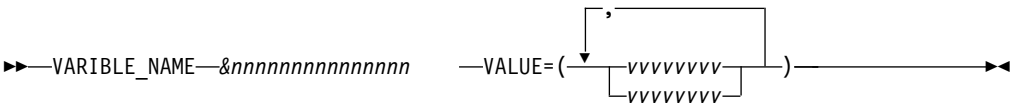
To display the currently active variable definitions, issue the following command from the operator console:

```
MODIFY stcname,DISPLAY VARDEFS
```

Where *stcname* is the name of the started task.

User defined variable syntax

User-defined variable definition syntax consists of the VARIBLE_NAME and VALUE keywords.



In the syntax descriptions, the underlined portion of the keyword indicates the minimally acceptable abbreviation.

VARIABLE_NAME

The VARIABLE_NAME keyword defines the name of a user-defined variable.

Syntax

VARIABLE_NAME=&*nnnnnnnnnnnnnnn*

Where *nnnnnnnnnnnnnnn* is a variable name that begins with an ampersand and has maximum of 16 characters (including the ampersand).

Example

```
VARIABLE_NAME=&GLOVAR1
```

VALUE

The VALUE keyword defines the valid values for the variable that you define with the VARIABLE_NAME keyword.

Syntax

VALUE=(*vvvvvvvv,vvvvvvvv*)

Where *vvvvvvvv* is 1 - 44 characters.

User-defined variable examples

The following examples show how to define and implement user-defined variables.

TSO user ID

The user-defined variable &JOBS_TSO is defined in the VARDEFS member:

```
 VARIABLE_NAME=&JOBS_TSO
     VALUE=(AUSER1,USER2,USER3)
```

Once defined, the variable can be specified in a rule definitions member on the JOBNAME keyword. For example:

```
INCLUDE JOBNAME=(PRODJOB,&JOBS_TSO)
```

Data set names

The user-defined variables &DSNEXC and &DSNGRP1 are defined in the VARDEFS member:

```
VARIABLE_NAME=&DSNEXC
      VALUE=(SYS1.*,
         ABC.PARMLIB,
         ANY.OTHER.SPECIAL.DATASETS)
VARIABLE_NAME=&DSNGRP1
      VALUE=(BCD.RCKC3.GLO.DSN1,
             CDE.RCKC2.GLP.DSN*,
             DEF.RCKC2.GLO.DSN4*)
```

Once defined, the variables can be specified in a rule definitions member on an INCLUDE or EXCLUDE statement. For example:

```
EXCLUDE DSNAME=(&DSNEXC,DSN1.*)
INCLUDE DSNAME=&DSNGRP1
```

Volume groups

The user-defined variable &VOLSET1 is defined in the VARDEFS member:

```
 VARIABLE_NAME=&VOLSET1
     VALUE=(VOL001,VOL002,VOL003,VOL004)
```

Once defined, the variable can be specified in a volume group definitions member on the VOLUMES keyword. For example:

```
* VGRP DEFINITIONS
       GROUP_NAME=GROUP01
       VOLUMES=&VOLSET1
```

Chapter 8. Managing control data set members and subsystems

From the Advanced Allocation Management ISPF interface, you can manage control data set members, view and update subsystem options, display subsystem statistics, and review Advanced Allocation Management event history.

Starting the ISPF interface

Use a TSO command to start the Advanced Allocation Management ISPF interface.

To start the Advanced Allocation Management ISPF interface, complete these steps:

1. Run the Advanced Allocation Management user interface CLIST using a TSO command. For example: **TSO GLO**.

   The Tivoli Advanced Allocation Management Primary Option Menu is displayed.
2. Press Enter to remove the copyright statement.

To access Advanced Allocation Management online help, press PF1.

Managing control data set members

To manage the contents of an Advanced Allocation Management control data set, complete these steps.

1. From the Tivoli Advanced Allocation Management Primary Option Menu, type 1 (**Manage Control Members**). The Tivoli Advanced Allocation Management Control Member Management panel opens. For example:

```
              Tivoli Advanced Allocation Management Control Member Management

 Option ===>

 blank Display list            R Rename control member
     V View control member     D Delete control member
     E Edit control member     S Syntax-check control member

 Enter a control data set name below or specify an active subsystem
 ID to retrieve the control data set name for that subsystem. Leave
 both fields blank to obtain a list of active subsystems.

 Control data set:
     Data set name . . . .  'ABCD.GLO.CONTROL'_____________________________
     Active Subsystem ID .  ____    (Select control data set by subsystem)

Control member:
    Member name . . . . .  RULESET1 (If V, E, R, D, S selected)
    New member name . . .           (If E, R selected)

Output Options:                     (If S selected)
  Listing Data Set (**)
   (fully qualified, without quotes)
  Output Mode . . . . . V           (View or Browse)
```

Figure 12. Tivoli Advanced Allocation Management Control Member Management panel

2. Specify the control data set that contains the members that you want to view, edit, rename, delete, or syntax-check. For more information, see "Specifying a control data set."
3. To select a control data set member, choose one of the following methods:
   - In the **Member name** field, type the name of the control data set member that you want to view, edit, rename, delete, or syntax-check.
   - To display a list of members from which you can perform view, edit, rename, delete, or syntax-check operations, leave the **Member name** field blank and press Enter.
4. Select one of the following options:

Option	Description
V (View)	"Viewing a control data set member" on page 277
E (Edit)	"Editing a control data set member" on page 278 and "Creating control data set members" on page 277
R (Rename)	"Renaming control data set members" on page 278
D (Delete)	"Deleting control data set members" on page 278
S (Syntax-check)	"Syntax-checking control data set members" on page 277

Specifying a control data set

To use the **V** (View), **E** (Edit), **R** (Rename), **D** (Delete), or **S** (Syntax-check) options, you must specify the control data set in the **Control data set** area of the Tivoli Advanced Allocation Management Control Member Management panel.

Choose one of the following methods to specify the control data set:

- If you know the name of the control data set, type the name of the control data set in the **Data set name** field. You must enclose the data set name in single quotation marks.
- To retrieve the corresponding subsystem control data set name of an active Advanced Allocation Management subsystem, type the four character subsystem ID of an active Advanced Allocation Management subsystem in the **Active Subsystem ID** field and press Enter.
- If multiple Advanced Allocation Management subsystems are active, you can display a list of active subsystem IDs from which you can select a subsystem and retrieve the corresponding data set name. Leave the **Control data set** fields blank and press Enter. The Tivoli Advanced Allocation Management Subsystem Selection panel opens.

  Note: If only one Advanced Allocation Management subsystem is active and you do not specify a control data set or member in the **Data set name** and **Member name** fields and you press Enter, the name of the control data set that is associated with the active Advanced Allocation Management subsystem is displayed in the **Data set name** field.

  To specify a subsystem from the Tivoli Advanced Allocation Management Subsystem Selection panel, select one of the following options:

  – To specify a specific subsystem, type an Advanced Allocation Management subsystem ID in the **Subsystem ID** field and press Enter. Advanced

Allocation Management retrieves the name of the control data set that is associated with the specified subsystem.

– To select a subsystem from a list of active Advanced Allocation Management subsystems, leave the **Subsystem ID** field blank and press Enter. The Tivoli Advanced Allocation Management Active Subsystems panel lists the active subsystems. Type S in the field to the left of the active subsystem that you want select and press Enter. Advanced Allocation Management retrieves the name of the control data set that is associated with the selected subsystem.

Creating control data set members

Use the **Edit** option in the ISPF interface to create a new control data set members.

1. From the Tivoli Advanced Allocation Management Control Member Management panel, type the name of the new control data set member in the **New member name** field.
2. Type E in the **Option** line, and press Enter. The new member is created and ISPF EDIT is invoked so that you can edit the new member.

Syntax-checking control data set members

After you specify a control data set and member on the Tivoli Advanced Allocation Management Primary Option Menu, or after you display a list of control data set members, you can check the syntax of RULEDEFS, VGRPDEFS, and VARDEFS control data set members.

Note: You cannot syntax-check OPTIONS members.

1. To syntax-check the selected member, type S on the **Options** line. To syntax-check a member from a list of members, type S next to the member that you want to check. The ISPF BROWSE function is invoked to view the results returned by syntax check routine.
2. Optional: If you want to retain the resulting output or prefer to use a data set name of your own choosing, specify the name of a listing data set in the **Listing Data Set** field. The specified data set must be fully qualified, without quotation marks.

   If the data set you specify does not exist, it is created. If the data set you specify exists, the contents are overwritten.

   Note: By default, the results of the syntax check are saved in a temporary file named *user_id*.*stcname*.SYNTAX.
3. Specify an output mode: V (view) or B (browse).

Advanced Allocation Management checks the selected control data set member for syntax errors and then invokes the ISPF BROWSE or VIEW function so that you can review the results of the syntax check. If no errors are detected, the message NO SYNTAX ERRORS FOUND is displayed at the end of the report.

Viewing a control data set member

To view a control data set member, complete the following step.

To view the selected member, type V on the **Options** line. To view a member from a list of members, type V next to the member that you want to view. The ISPF VIEW function is invoked so that you can view the selected control data set member.

Editing a control data set member

To edit a control data set member, complete the following step.

To edit the selected member, type E on the **Options** line. To edit a member from a list of members, type E next to the member that you want to edit. The ISPF EDIT function is invoked so that you can modify the selected member.

Renaming control data set members

To rename a control data set member, complete these steps.

1. To rename the selected member, type R on the **Options** line. To rename a member from a list of members, type R next to the member that you want to rename. The Member Rename panel opens. The current member name is displayed in the **Old Name** field.
2. To rename the member, type the new 1 - 8 character member name in the **New member name** field and press Enter. The selected member is renamed.

Deleting control data set members

To delete a control data set member, complete these steps.

1. To delete the selected member, type D on the **Options** line. To delete a member from a list of members, type D next to the member that you want to delete.
2. Confirm the deletion and press Enter. After you confirm the deletion, the member is deleted.

Viewing and updating subsystem options

To view and modify (if authorized) selected subsystem settings, complete these steps.

An Advanced Allocation Management subsystem must be active to access the Tivoli Advanced Allocation Management Subsystem Settings panel.

1. From the Tivoli Advanced Allocation Management Primary Option Menu, type 2 (**View/Update Options**) The Tivoli Advanced Allocation Management Subsystem Settings panel opens. For example:

```
          Tivoli Advanced Allocation Management Subsystem Settings
Fixed Subsystem Settings:
  Subsystem ID. . . . . . . AOP1
  Started Task Name . . . . ALLOCP1
  Control Data Set Name . . ABCD.GLO.CONTROL


Modifiable Subsystem Settings:
  Subsystem Status. . . . . .E        (Enable or Disable)
  Active Rule Definition. . .RULESET1_(name of member to be activated)
                                      ("*" to re-activate same member)
  Active VGRP Definition. . .VGRPSET1 (name of member to be activated)
                                      ("*" to re-activate same member)
  Current VAR Definition. . .VARSET1_ (name of member to SETVARDEFS)

Status Information:
 Rule Definition last activated . . . 09/15/2014 13:06:50
 VGRP Definition last activated . . . 09/15/2014 13:07:12
 VAR  Definition last set . . . . . . 09/15/2014 13:05:20

Modifiable Product-level Settings:
Intercepts Status . . . . E        (Enable or Disable) (** see note)

**- intercepts apply to ALL active Advanced Allocation Management subsystems
```

Figure 13. Tivoli Advanced Allocation Management Subsystem Settings

2. View or update the following settings as needed.

   The following **Fixed Subsystem Settings** are view only:

Option	Description
Subsystem ID	The active subsystem ID.
Started Task Name	The name of the started task that is associated with the subsystem ID.
Control Data Set Name	The name of the control data set for the active subsystem.

   The following **Modifiable Subsystem Settings** can be modified by authorized users:

Option	Description
Subsystem status	The current subsystem status: enabled (**E**) or disabled (**D**). To change the status, specify one of the following options: **E** Enable the subsystem so that processing occurs. **D** Disable the subsystem so that no processing occurs.
Active rule definition	The name of the active rule definitions member of the control data set that is displayed in the **Control Data Set Name** field. To activate a rule definitions member, type the name of the rule definition member that you want to activate in this field and press Enter. To reactivate the same member, type an asterisk in this field. **Note:** Rule definitions must be activated for their definitions to take effect.

Option	Description
Active VGRP definition	The name of the active volume group definitions member of the control data set that is displayed in the **Control Data Set Name** field. To activate a volume group definitions member, type the name of the volume group definition member that you want to activate in this field and press Enter. To reactivate the same member, type an asterisk in this field. **Note:** Volume group definitions must be activated for their definitions to take effect.
Current VAR Definition	The name of the current user-defined variable definitions member of the control data set that is displayed in the **Current VAR Definition** field.

The following **Status Information** fields are view only:

Option	Description
Rule definition last activated	The date in the format MM/DD/YYYY (month, day, and year) and time in the format HH:MM:SS (hours, minutes, and seconds) that the rule definition member was last activated.
VGRP definition last activated	The date in the format MM/DD/YYYY (month, day, and year) and time in the format HH:MM:SS (hours, minutes, and seconds) that the volume group definition member was last activated.
VAR Definitions last set	The date in the format MM/DD/YYYY (month, day, and year) and time in the format HH:MM:SS (hours, minutes, and seconds) that the variable definitions member name was last set.

Modifiable Product-level Settings fields can only be modified by authorized users:

Option	Description
Intercepts Status	The current intercept status: enabled (**E**) or disabled (**D**). To change the status, specify one of the following options: **E** Enable product-level intercepts to allow Advanced Allocation Management to perform processing. **D** Disable product-level intercepts. Disabling the intercepts allows the Advanced Allocation Management started task to remain active without performing processing. Disabling the product-level intercepts can be useful in situations when you do not want to shut down Advanced Allocation Management completely (for example, for debugging purposes). **Note:** The **Intercepts Status** setting enables or disables product-level intercepts for *all* active Advanced Allocation Management subsystems. Enabling or disabling the product-level intercepts does not start or stop the Advanced Allocation Management started task.

3. When you are finished viewing or updating the settings, press PF3 to exit.

Chapter 9. Space recovery strategies and statistics

When there is insufficient space to allocate a data set, use the PRIM_REDUCE, SEC_REDUCE, SEC_BEST, and SEC_INC functions to modify extent distribution.

Space recovery strategies

You can combine these functions to design an extent distribution scheme that best suits your needs. For example, using PRIM_REDUCE and SEC_REDUCE with the VOL_ADD function ensures that allocations are successful, regardless of the current state of disk volume fragmentation.

PRIM_REDUCE

The PRIM_REDUCE function allows the initial allocation on the first and subsequent volumes of a data set to succeed when there is insufficient space to allocate the original amount. If you use PRIM_REDUCE without specifying any secondary allocation functions and some or all, of the requests for secondary allocations fail, a data set spreads a single extent across multiple volumes as shown in Figure 14.

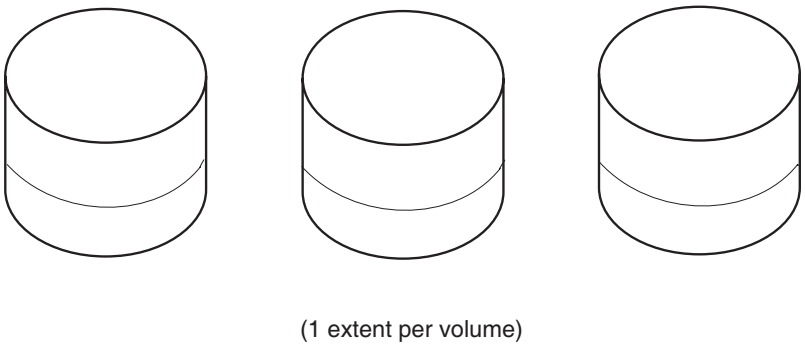


Figure 14. Space recovery strategy (PRIM_REDUCE)

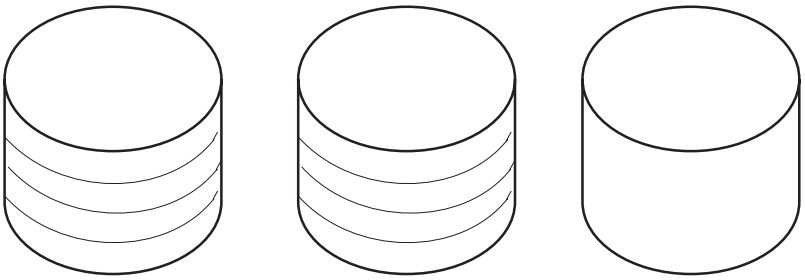
This scenario assumes that you originally specified sufficient volumes and that you used the VOL_ADD function to add volumes.

Note: If you specify explicit volsers for a data set and one of the non-first volumes contains no free space when an extend to a new volume occurs, end-of-volume processing produces an E37 abend. To skip the volume that has no free space and try the next explicitly specified volume or use the VOL_ADD function to add a volume, you must specify PRIM_REDUCE=(,ENABLE).

SEC_REDUCE

When the failure of an attempt to allocate the secondary quantity for a data set fails, SEC_REDUCE reduces the secondary quantity so that the allocation succeeds. Therefore, when you specify SEC_REDUCE and a volume has little space or is fragmented, the data set uses as many extents as permitted for the data set type, up to the maximum amount. If the data set extends to other volumes, the same processing occurs again. This causes the distribution of extents for the data set to

consist of many potentially smaller extents across fewer volumes as shown in Figure 15, as compared to using the PRIM_REDUCE function by itself.

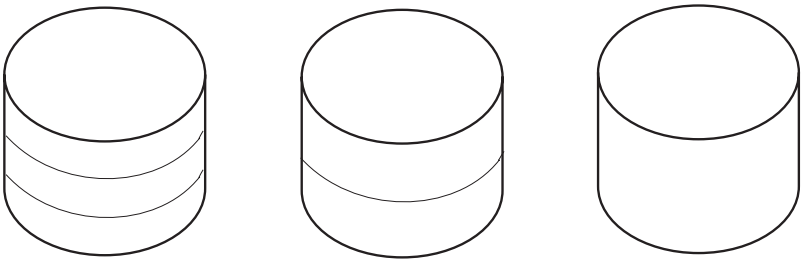


(Many extents per volume)

Figure 15. Space recovery strategy (SEC_REDUCE)

SEC_BEST

The SEC_BEST function reduces a secondary quantity *prior* to the allocation attempt, resulting in fewer extents across fewer volumes and reduced data set fragmentation.



(Larger and fewer extents per volume)

Figure 16. Space recovery strategy (SEC_BEST)

To allocate a secondary quantity of a specific size, if a contiguous area of space is not available, the system attempts to use up to five extents. The result can be many small extents that exceed the maximum extent limit for the volume, and cause the data set to extend to another volume. Using the SEC_BEST function ensures that each secondary allocation attempt uses a single extent of the maximum size space available, up to the original quantity requested. This approach reduces fragmentation and data set extends. Compared to the PRIM_REDUCE function, the SEC_BEST function produces many extents spread across fewer volumes. Because Advanced Allocation Management reduces the size of the extent before an allocation attempt, the extents are typically larger in size and fewer in number than can occur when using the SEC_REDUCE function.

SEC_INC

The SEC_INC function accommodates unforeseen changes in workload by increasing the secondary allocation quantity at a specific point. However, this does have implications on the number of extents and extent distribution. If a data set grows much larger than normal, it might create many extents on the original

volume and possibly on additional volumes. The SEC_INC function increases the secondary allocation, resulting in fewer extents and possibly fewer volumes.

Displaying recovery statistics

To display Advanced Allocation Management recovery statistics, complete these steps.

An Advanced Allocation Management subsystem must be active to access the Tivoli Advanced Allocation Management Recovery Statistics panel.

1. From the Tivoli Advanced Allocation Management Primary Option Menu, type 3 (**Display Statistics**).
2. Press Enter to update the statistics. The Tivoli Advanced Allocation Management Recovery Statistics panel displays the statistics for the selected Advanced Allocation Management subsystem.
3. When you finish viewing the statistics, press PF3.

Recovery statistics

The following table provides descriptions of the recovery statistics.

Table 13. Recovery statistics

Field	Description
Statistics recording started at	The date in the format MM:DD:YY (month, day, and year) and time in the format HH:MM:SS (hours, minutes, and seconds) Advanced Allocation Management began recording statistics for the selected subsystem.
Initial volume primary space allocation failures recovered	The number of times that the PRIM_REDUCE function reduced primary space.
Subsequent volume primary space allocation failures recovered	The number of times Advanced Allocation Management reduced the primary allocation on a volume that was added.
Undefined secondary allocation space abends avoided (D37)	The number of times the SEC_ALLOC function added a secondary allocation during D37 processing when no secondary space value was specified or when it was specified as zero.
Unavailable secondary allocation space abends avoided (B37/E37)	The number of times the SEC_BEST function decreased the secondary allocation to fit in the largest single extent on the volume during end-of-volume processing.
Unavailable secondary allocation space abends recovered (B37/E37)	The number of times the SEC_REDUCE function recovered from secondary allocation space failures by reducing the allocation amount to a value that is consistent with the free space on the volume.
Insufficient volumes defined abends recovered (B37/E37)	The number of times that the VOL_ADD function dynamically added volumes during end-of-volume processing.

Table 13. Recovery statistics (continued)

Field	Description
Possible insufficient space abends avoided (B37/E37)	The number of times the SEC_INC function increased the size of the secondary allocation as data set extends occurred after a particular extent and at a particular volume number.
Space release option added to primary or secondary allocation	The number of times Advanced Allocation Management added the SPACE_RELEASE option to an allocation.
Insufficient PDS directory space for member save errors recovered	The total number of PDS directory space errors that were recovered as a result of the Advanced Allocation Management PDS_DIR_INC function.
NOT CATLGD 2 errors processed	The total number of NOT CATLGD 2 related errors that Advanced Allocation Management processed after the started task was initialized.
Data sets initialized	The number of data sets that Advanced Allocation Management initialized.
Total number of abends and errors either avoided or recovered	The total number of abends and errors that Advanced Allocation Management avoided or recovered.
Number of tracks recovered from adding space release	The total number of tracks that were freed by the SPACE_RELEASE function.
Number of Allocation parms set by Enhanced Allocation Management	The number of data set parameters that were added, changed, or deleted by Enhanced Allocation Management.
Non-SMS Volume(s) selected by Advanced Volume Selection	The number of non-SMS data sets for which volumes were selected at allocation time by Advanced Volume Selection.
Number of data set allocations failed by TERM_ALLOC	The number of data set allocations that were forced to fail as a result of the TERM_ALLOC function.
Total data sets processed by RULEDEF_TEST mode	The number of data sets that went through primary or secondary allocation and matched a rule definition with RULEDEF_TEST set to ENABLE.

Chapter 10. Event recording and reporting

The Advanced Allocation Management event recording and reporting facility helps you evaluate product benefits, specific events that Advanced Allocation Management encountered, and the subsequent actions that were taken for a particular job or time frame, and so on.

Advanced Allocation Management uses the z/OS System Management Facility (SMF) to record and retain historical information that can be later archived, retrieved, and reported upon. The report utility extracts the SMF records that Advanced Allocation Management created and reports them in various formats. You can view the data using the **View Event History** option on the Tivoli Advanced Allocation Management Primary Option Menu or by using the batch reporting utility. A DSECT of the record layout is provided so that you can write your own programs to report on the SMF data, processing it in the manner you choose.

Note: Event data recording is optional. Advanced Allocation Management does not write event data to SMF unless you configure it to do so. For more information, see "Specifying subsystem options" on page 29.

Enabling event recording

After you enable event recording, Advanced Allocation Management automatically begins recording event data to SMF.

To enable event recording, complete the following steps:

1. Add a new, previously unused SMF record type in the range 128 - 255 to the MVS SMFPRM*xx* PARMLIB member, which contains the list of record types to record.

   For more information about SMF and the MVS SMFPRM*xx* PARMLIB member, see the book *z/OS MVS System Management Facilities (SMF)*, which is available at ibm.com.
2. Issue the MVS **SET** command or perform an IPL to activate the updated SMFPRM*xx* member.
3. Specify the record type in the SMF_RECORD_ID parameter of the OPTIONS member. For more information, see "Specifying subsystem options" on page 29.
4. If Advanced Allocation Management is running, stop and restart it to register the OPTION member change.

   Use the **View Event History** option on the Tivoli Advanced Allocation Management Primary Option Menu or the batch reporting facility to obtain reports on the data.

Disabling event recording

Disable event recording by removing the record type from the SMFPRM*xx* PARMLIB member and the SMF_RECORD_ID parameter.

To disable event recording, complete the following steps:

1. Remove the record type from the MVS SMFPRM*xx* PARMLIB member.

2. Remove the record type from the SMF_RECORD_ID parameter of the OPTIONS member. For more information, see "Specifying subsystem options" on page 29.
3. If Advanced Allocation Management is running, stop and restart it to register the OPTION member change.

Report filter criteria and filter sets

Filter criteria provide a way to select only the records that match certain specifications to be reported.

Filter criteria are specified as elements of *filter sets*. A filter set represents one or more filter criteria to be ANDed together to determine whether a record matches. In following example, only the records where the job name is ABC *and* the ddname is XYZ, are to be reported:

```
FILTER_SET=(JOBNAME=ABC,DDNAME=XYZ)
```

Notes:

1. FILTER_SET= must precede the criteria.
2. Filter set criteria must be separated by commas, even if they are on separate lines. You do not need to specify a comma after the last filter set criteria that you specify.

Specifying multiple filter sets

You can specify more than one filter set. Each filter set that you specify is ORed with the other filter sets.

For example, the following specification indicates only those records where the job name is ABC *or* DEF are to be reported:

```
FILTER_SET=(JOBNAME=ABC)
FILTER_SET=(JOBNAME=DEF)
```

Note: If you are using the **View Event History** option on the Tivoli Advanced Allocation Management Primary Option Menu, you can specify a maximum of 10 filter sets and a maximum of 10 filter criteria per filter set. If you are using the batch reporting utility, a maximum of 32 filter sets and a maximum of 32 filter criteria per filter set can be specified.

Filter criteria formats

Each of the following FILTER_SET= specifications are valid formats for specifying filter criteria.

```
FILTER_SET=JOBNAME=ABC

FILTER_SET=(JOBNAME=ABC)

FILTER_SET=(JOBNAME=ABC,DDNAME=XYZ)

FILTER_SET=(JOBNAME=ABC,
           DDNAME=XYZ)

FILTER_SET=(
           JOBNAME=ABC,
           DDNAME=XYZ
          )
```

Report options

To produce the output you want, use the report options DETAIL (**Detail Report**), SUMMARY (**Summary Report**), and FORMAT (**Format**) in combination with each other.

The following rules apply when you specify report options:

- DETAIL=N and SUMMARY=N are not valid together.
- FORMAT=L and SUMMARY=Y are not valid together.
  (If SUMMARY=Y, FORMAT must be R)
- FORMAT=L and DETAIL=N are not valid together.
  (If FORMAT=L, DETAIL must be Y)

Report option combinations and results

The following table shows report option combinations and corresponding results.

Table 14. Report options and results

If you specify:	The result is:
FORMAT=LIST DETAIL=N SUMMARY=N	This is not a valid combination. An error condition is returned.
FORMAT=LIST DETAIL=N SUMMARY=Y	This is not a valid combination. An error condition is returned.
FORMAT=LIST DETAIL=(A or B) SUMMARY=N	Unformatted records are generated in a list. No summary report is generated.
FORMAT=LIST DETAIL=(A or B) SUMMARY=Y	This is not a valid combination. An error condition is returned.
FORMAT=REPORT DETAIL=N SUMMARY=N	This is not a valid combination. An error condition is returned.
FORMAT=REPORT DETAIL=N SUMMARY=Y	A summary report is generated.
FORMAT=REPORT DETAIL=(A or B) SUMMARY=N	Detail records are generated in a formatted report. No summary report is generated.
FORMAT=REPORT DETAIL=(A or B) SUMMARY=Y	Detail records are generated in a formatted report. A summary report is generated at the end.

Generating event history reports from the ISPF interface

Use the Advanced Allocation Management event recording and reporting facility to create reports online or through a batch reporting utility. With the options that are available on the Tivoli Advanced Allocation Management View Event History panel, you can customize and generate Advanced Allocation Management event history reports.

Note: Advanced Allocation Management does not record event data unless you specifically configure it to do so. For more information, see “Enabling event recording” on page 287.

For information about the batch reporting utility, see “Batch reporting utility” on page 292.

To create reports online, complete these steps:

1. From the Tivoli Advanced Allocation Management Primary Option Menu, type 4 (**View Event History**). The Tivoli Advanced Allocation Management View Event History panel opens. For example:

```
          Tivoli Advanced Allocation Management View Event History

Input Options:
  SMF Source File (**).
  SMF Record ID . . . .
  Filter Set Criteria .

Enter "/" to select option
   Specify additional SMF source files
   Specify additional filter set criteria

Report Options:
  Sort-By Field . . . .
  Detail Report . . . .      (Basic, All, or No)
  Summary Report. . . .      (Yes or No)
  Format. . . . . .          (Report or List)
  Work DS Prefix. . . .      (Defaults to TSO userid)

Output Options:
  Listing Data Set (**)
  Output Mode . . . . .      (View or Browse)

 **- data set names must be fully qualified, without quotes
```

Figure 17. Tivoli Advanced Allocation Management View Event History

2. In the **Input Options** area, specify the following options:

Option	Description
SMF Source File	(Required.) Type the name of the non-VSAM SMF data set to search for Advanced Allocation Management event records. **Note:** If the data set is on tape, use the batch utility to create the report.
SMF Record ID	(Required.) Type the 3-character numeric value that is specified by SMF_RECORD_ID in the OPTIONS member. For more information, see "Specifying subsystem options" on page 29.
Filter Set Criteria	Specify the filter set criteria. For more information, see "Report filter criteria and filter sets" on page 288.
Specify additional SMF source files	(Optional.) Type a forward slash, and specify up to nine additional non-VSAM SMF source files. **Note:** If the data set is on tape, use the batch utility to create the report. Press Enter to process the additional files.
Specify additional filter set criteria	(Optional.) Type a forward slash, and specify up to nine additional filter criteria. **Note:** To specify more than ten filter criteria, use the batch reporting utility. Press Enter to process the additional criteria.

3. The View Event History report options (**Detail Report**, **Summary Report**, and **Format**) are designed to work in combination with each other to produce the appropriate output. Specify the following **Report Options**:

Option	Description
Sort-By Field	(Optional.) Specify the field to sort the report. **Note:** If you do not provide a sort field, the records are reported in the order in which they are read.
Detail Report	Detail records provide a detailed description of each entity that encounters an event that is recorded. Specify one of the following options: **Y** (Default.) Include detail records in the report. **N** Do not include detail records in the report. **A** (ALL) Generate an expanded report with additional values. **B** (BASIC) Include detail records in the report. (Specifying this option is the same as specifying Y.) **Note:** If **Format**=R (Report) is also specified, the detail records are formatted. If **Format**=L (List) is also specified, the detail records are composed of raw data, in a format that is suitable for input into another report writing utility.
Summary Report	Specify whether to generate a summary of the records that are processed: **Y** Generate a summary of the records that are processed. **N** Do not generate a summary of the records that are processed. **Note:** Because the summary report is always formatted, **Summary Report**=Y is valid only when **Format**=R (Report) is also specified.
Format	Specify the type of output to generate: **R (Report)** Generate a formatted report. **L (List)** Generate raw data records in a format that is suitable for input into other report writing utilities.

Option	Description
Work DS prefix	(Optional.) By default, temporary work data sets are created using a high-level qualifier of your user ID. If you want to specify a different high-level qualifier, enter it here. Ensure that the data set name node that you specify meets the following requirements: 1. The first character must be an alphabetic or national ($, #, @) character. 2. Characters 2 - 8 must be an alphanumeric or national character, a hyphen, or an open-bracket ("{"). If the resulting data set names do not exist, they are created. If the data sets you specify exist, their contents are overwritten.

4. Specify the following **Output Options**:

Option	Description
Listing Data Set	Specify the name of the listing data set: • To retain the output, specify a data set name. If the data set does not exist, it is created. If the data set exists, the contents are overwritten. • If you do not specify a data set name, a temporary data set is allocated and then deleted after you finish browsing or viewing the output.
Output Mode	Specify the ISPF facility to display the resulting report: **V (View)** Type `V` to specify ISPF VIEW. **View** allows for greater manipulation (such as record sorting and line manipulations) of the displayed data, but is limited by the size of the report that it can display. **B (Browse)** Type `B` to specify ISPF BROWSE. **Browse** is unlimited in the size of the report it can display. However, it is much more limited in the way the data can be manipulated.

5. When you finish specifying options, press PF3.

Batch reporting utility

The batch reporting utility generates event history reports in batch.

In some cases, the batch reporting utility provides greater flexibility and control over the data and the output destination than the **View Event History** option on the Tivoli Advanced Allocation Management Primary Option Menu.

Configuring batch reporting

To use batch reporting, you must configure the JCL in the GLOJLOGR member of the Advanced Allocation Management sample library.

To configure the batch reporting utility JCL, complete the following steps:

1. Replace `GLO.V3R3` with the high-level qualifier of the installed Advanced Allocation Management load libraries.
2. Specify the names of the SMF data sets to search for Advanced Allocation Management event records. In the sample JCL, `SMFIN=SMF.DATA.SET1` is the SMF data set specification.
3. Specify the SMF record type that was specified in the SMF_RECORD_ID parameter of the OPTIONS member. For more information, see "Specifying subsystem options" on page 29.
4. Modify the following required DD statements in the batch reporting JCL:
   a. The HISTOUT DD statement specifies where to write the report output. To ensure that the proper attributes are set (depending upon the functions that were requested), allow the utility to place file attributes on the data set at the time it is written. However, you can specify the following items:
      - If Format=R is specified, HISTOUT must have a record format of fixed (F) or fixed block (FB) and a record length (LRECL) of 80.
      - If Format=L is specified, HISTOUT must have a record format of variable (V) or variable blocked (VB) and a record length (LRECL) of 1024.

   b. The CNTLOUT DD statement specifies where to write execution statistics and diagnostic information. CNTLOUT must have a record format of fixed (F) or fixed block (FB) and a record length (LRECL) of 80.
   c. The CNTLIN DD statement specifies the location of the input control statements. It must have a record format of fixed (F) or fixed block (FB) and a record length (LRECL) of 80.
5. Specify input control statements. The following control statements are valid input to the batch reporting utility.

   Note: All control statements must start in card column one. This restriction does not apply to continuation statements.

   SMFIN

   (Required.) Specify the name of one or more non-VSAM SMF data sets to search for Advanced Allocation Management event records. Separate multiple data set names with a comma. A maximum of 32 data set names can be specified. Each of the following SMFIN= specifications are valid formats for specifying data set names using SMFIN:

```
SMFIN=SMF.data.set1

SMFIN=(SMF.data.set1)

SMFIN=(SMF.data.set1,SMF.data.set2)

SMFIN=(SMF.data.set1,
       SMF.data.set2)

SMFIN=(
       SMF.data.set1,
       SMF.data.set2
      )
```

Note: Multiple dsnames that are specified on an SMFIN statement must be separated by commas, even if they are specified on separate lines. You do not need to specify a comma after the last dsname.

SMF_RECORD_ID=xxx
(Required.) Specify the SMF record type that was designated for use by Advanced Allocation Management. *xxx* is a one to three digit record identifier (128 - 255).

Note: *xxx* is the numeric value that is specified in the Advanced Allocation Management control data set OPTIONS member that is generated during the installation and configuration process.

FILTER_SET
Specify a set of filter criteria to select only the records that match certain specifications for reporting. Each filter set is composed of one or more filter sets. A maximum of 32 filter sets can be specified and a maximum of 32 filter criteria per filter set can be specified.

SORT_BY
The field by which the selected records are to be sorted.

Note: SORT_BY is optional. If you do not specify a value for SORT_BY, the records are reported in the order in which they are read.

DETAIL_REPORT
Detail records provide a detailed description of each entity that encounters an event that is recorded. The following options are valid:

A (or ALL)
Include extended detail records in the report.

B (or BASIC)
Include basic detail records in the report.

Y (or YES)
Include detail records in the report.

Note: Specifying Y is the same as specifying B and is supported for downward compatibility.

N (or NO)
Do not include detail records in the report.

Note:

- If Format=R (Report) is also specified, the detail records are formatted.
- If Format=L (List) is also specified, the detail records are composed of raw data, in a format suitable for input into another report-writing utility.

SUMMARY_REPORT
Specify whether to generate a summary of the records that are processed. The following options are valid:

Y (or YES)
Generate a summary of the records that were processed.

N (or NO)
Do not generate a summary.

Note: Because the summary report is always formatted, Summary Report=Y is valid only when Format=R (Report) is also specified.

FORMAT
Specify the type of output to generate. The following options are valid:

R (or REPORT)
Generate a formatted report.

L (or LIST)
Generate raw data records in a format that is suitable for input into other report writing utilities.

6. When you are finished modifying the batch reporting utility JCL, submit the JCL to generate the report.

Sample execution report

The following example execution report was generated by the batch reporting utility JCL.

Note: The reports that you generate can vary depending upon the options and filters that you select.

Figure 18 shows a sample execution report.

```
          * * * TIVOLI Advanced Allocation Management EVENT HISTORY REPORT * * *

VALIDATING CONTROL STATEMENTS-

SMFIN=PDABCD.SMF.OUT
SMF_RECORD_ID=128

END OF CONTROL STATEMENT INPUT

THE FOLLOWING REPORTING OPTIONS ARE IN EFFECT-
FORMAT=REPORT
DETAIL_REPORT=BASIC
SUMMARY_REPORT=YES


STATISTICS SUMMARY:

NUMBER OF SMF RECORDS READ.....................0000013
NUMBER OF PROCESSING ACTIVITY RECORDS READ.....0000013
NUMBER OF RECORDS THAT PASSED FILTERING.........0000013

PROCESSING IS COMPLETE; RETURN CODE IS 0000000
```

Figure 18. Sample execution report

The following statistics are summarized in the sample execution report:

Number of SMF records read
The number of SMF records that are read by Advanced Allocation Management.

Number of processing activity records read
The number of processing activity records that are read by Advanced Allocation Management.

Number of records that passed filtering
The number of records that passed filtering.

Sample formatted basic detail and extended detail record reports

The following examples of a formatted basic detail report and a formatted extended detail record report were generated by batch reporting utility JCL.

Note: The reports that you generate can vary depending upon the options and filters that you select.

Example: Formatted basic detail report

Figure 19 shows an example of a formatted basic detail record report.

```
* * * TIVOLI Advanced Allocation Management PROCESSING ACTIVITY DETAIL REPORT * * *

DATE: 2014/02/07    JOBNAME: GLOJIVP1    STEPNAME: PRIMRED
TIME: 14:22:26      JOBTYPE: JOB         PROCSTEP:
SYSTEM ID: ABC1     PROGRAM: IEBDG       DDNAME: SYSUT2
SUBSYS: APRD      RDEF_LINE: 00002986    DSORG: PS
DSNAME: PDABCD.GLO.TEST.DATA
ACTIVITY: INITIAL VOLUME PRIMARY SPACE ALLOCATION FAILURE RECOVERED

DATE: 2014/02/07   JOBNAME: GLOJIVP1    STEPNAME: SECALLOC
TIME: 14:22:26     JOBTYPE: JOB         PROCSTEP:
SYSTEM ID: ABC1    PROGRAM: IEBDG       DDNAME: SYSUT2
SUBSYS: APRD     RDEF_LINE: 00002990    DSORG: PS
DSNAME: PDABCD.GLO.TEST.DATA
ACTIVITY: UNDEFINED SECONDARY ALLOCATION SPACE ABEND AVOIDED (D37)
```

Figure 19. Sample formatted basic detail record report

The basic detail record report provides fundamental information about Advanced Allocation Management processing activity.

Example: Formatted extended detail record detail report

Figure 20 shows an example of a formatted extended detail record report.

```
    * * * TIVOLI Advanced Allocation Management PROCESSING ACTIVITY DETAIL REPORT * * *

DATE:      2014/02/07    JOBNAME: GLOJIVP1    STEPNAME: SECALLOC
TIME:      14:22:26      JOBTYPE: JOB         PROCSTEP:
SYSTEM ID: ABC1          PROGRAM: IEBDG       DDNAME:   SYSUT2
SUBSYS:    APRD        RDEF_LINE: 00000097    DSORG:    PS
DSNAME:    PDABCD.GLO.TEST.DATA
ACTIVITY:  UNDEFINED SECONDARY ALLOCATION SPACE ABEND AVOIDED (D37)
TEMPDS:    NO            SMS-MGD: YES         DATACLAS: TSODC
STORCLAS:  TSOSC        MGMTCLAS: STANDARD    STORGRP:  SGSG2
SDISP:     NEW             NDISP: CATLG       ADISP:    DELETE
```

Figure 20. Sample formatted extended detail record report

The extended detail record report includes SMS details, such as storage class and storage group information in addition to the information that is provided in the basic detail record report.

ACTIVITY field details for the NOTCAT2 function

A formatted basic or extended detail record report can include the following **ACTIVITY** field details for the NOTCAT2 function.

NOT CATLGD 2 ERR PROCESSED WITH JCLFAIL ON VOLSER *vvvvvv* AT *ttttt*
The number of NOTCAT2 errors that were processed with a JCLFAIL action.

vvvvvv
If the error was detected at allocation time, *vvvvvv* is the first volser of the newly created data set. If the error was detected at step termination time, *vvvvvv* is the first volser of the currently cataloged old data set.

ttttt
If the error was detected at allocation time, *ttttt* is set to ALLOC. If the error was detected at step termination time, *ttttt* is set to TERM.

NOT CATLGD 2 ERR PROCESSED WITH SCRATCH ON VOLSER *vvvvvv* AT *ttttt*
The number of NOTCAT2 errors that were processed with a SCRATCH (UNCATALOG and DELETE) action.

vvvvvv
vvvvvv is the first volser of the current existing data set that was deleted.

ttttt
If the error was detected at allocation time, *ttttt* is set to ALLOC. If the error was detected at step termination time, *ttttt* is set to TERM.

NOT CATLGD 2 ERR PROCESSED WITH RENAME ON VOLSER *vvvvvv* AT *ttttt*
The number of NOTCAT2 errors that were processed with a RENAME action.

vvvvvv
vvvvvv is the first volser of the current existing data set that was renamed.

ttttt
If the error was detected at allocation time, *ttttt* is set to ALLOC. If the error was detected at step termination time, *ttttt* is set to TERM.

If the Advanced Allocation Management field indicates that the NOT CATLGD 2 error was processed with the RENAME action, the following line is included:

```
NEW NAME: dsname
```

Where *dsname* is the new name that is assigned to the existing data set that was renamed.

NOT CATLGD 2 ERR PROCESSED WITH UNCATLG ON VOLSER *vvvvvv* AT *ttttt*
The number of NOTCAT2 errors that were processed with an UNCATALOG action.

vvvvvv
vvvvvv is the first volser of the current existing data set that was uncataloged.

ttttt
If the error was detected at allocation time, *ttttt* is set to ALLOC. If the error was detected at step termination time, *ttttt* is set to TERM.

ACTIVITY field details for the Enhanced Allocation Management, Advanced Volume Selection, and TERM_ALLOC functions

A formatted basic or extended detail record report can include the following **ACTIVITY** field details for the Enhanced Allocation Management, Advanced Volume Selection, and TERM_ALLOC functions:

DATA SET ALLOCATION FAILED BY THE TERM_ALLOC FUNCTION
Data set allocations that were forced to fail as a result of the TERM_ALLOC function.

ALLOCATION PARAMETER SET BY ENHANCED ALLOCATION MANAGEMENT
Data set parameters that were added, changed, or deleted by Enhanced Allocation Management.

NON-SMS VOLUME(S) SELECTED BY ADVANCED VOLUME SELECTION
The number of non-SMS data sets for which volumes were selected at allocation time by Advanced Volume Selection.

Sample formatted summary record report

The following example of a formatted summary record report was generated by the batch reporting utility JCL.

Note: The reports that you generate can vary depending upon the options and filters that you select.

Figure 21 shows a sample formatted summary record report.

```
* TIVOLI Advanced Allocation Management PROCESSING ACTIVITY SUMMARY REPORT *

INITIAL VOLUME PRIMARY SPACE ALLOCATION FAILURES RECOVERED....... 000001
SUBSEQUENT VOLUME PRIMARY SPACE ALLOCATION FAILURES RECOVERED.... 000000
UNDEFINED SECONDARY ALLOCATION SPACE ABENDS AVOIDED (D37)........ 000000
UNAVAILABLE SECONDARY ALLOCATION SPACE ABENDS AVOIDED (B37/E37).. 000000
UNAVAILABLE SECONDARY ALLOCATION SPACE ABENDS RECOVERED (B37/E37) 000000
INSUFFICIENT VOLUMES DEFINED ABENDS RECOVERED (B37/E37)......... 000023
POSSIBLE INSUFFICIENT SPACE ABENDS AVOIDED (B37/E37)............. 000023
SPACE RELEASE OPTION ADDED TO PRIMARY OR SECONDARY ALLOCATION.... 000000
INSUFFICIENT PDS DIRECTORY SPACE ERRORS RECOVERED................ 000000
NOT CATLGD 2 ERRORS PROCESSED.................................... 000002
DATA SETS INITIALIZED............................................ 000000
DATA SETS PROCESSED BY RULEDEF_TEST MODE......................... 000000
ALLOCATION PARMS SET BY ENHANCED ALLOCATION MANAGEMENT........... 000000
NON-SMS VOLUME(S) SELECTED BY ADVANCED VOLUME SELECTION.......... 000000
NUMBER OF DATA SET ALLOCATIONS FAILED BY TERM_ALLOC.............. 000000
```

Figure 21. Sample formatted summary record report

The following activity summary information is shown in the sample summary record report:

Initial volume primary space allocation failures recovered
The number of times primary space was reduced by the PRIM_REDUCE function.

Subsequent volume primary space allocation failures recovered
The number of times Advanced Allocation Management reduced the primary allocation on a volume that was added.

Undefined secondary allocation space abends avoided (D37)
The number of times the SEC_ALLOC function added a secondary

allocation during D37 processing when no secondary space value was specified (or, it was specified as zero).

Unavailable secondary allocation space abends avoided (B37/E37)
The number of times the SEC_BEST function decreased the secondary allocation to fit in the largest single extent on the volume during end-of-volume processing.

Unavailable secondary allocation space abends recovered (B37/E37)
The number of times the SEC_REDUCE function recovered from secondary allocation space failures by reducing the allocation amount to a value that was consistent with the free space on the volume.

Insufficient volumes defined abends recovered (B37/E37)
The number of times that the VOL_ADD function dynamically added volumes during end of volume processing.

Possible insufficient space abends avoided (B37/E37)
The number of times the SEC_INC function increased the size of the secondary allocation as data set extends occurred after a particular extent, and at a particular volume number.

Space release option added to primary or secondary allocation
The number of times Advanced Allocation Management added the SPACE_RELEASE option to an allocation.

Insufficient PDS directory space for member save errors recovered
The total number of PDS directory space errors that were recovered as a result of the Advanced Allocation Management PDS_DIR_INC function.

NOT CATLGD 2 errors processed
The total number of NOT CATLGD 2 related errors that were processed by Advanced Allocation Management after the started task was initialized.

Data sets initialized
The number of data sets that were initialized by Advanced Allocation Management.

Data sets processed by RULEDEF_TEST mode
The total number of data allocations or extends that matched a rule definition INCLUDE or EXCLUDE statement where RULEDEF_TEST was set to ENABLE.

Allocation parms set by Enhanced Allocation Management
The number of data set parameters that were added, changed, or deleted by Enhanced Allocation Management.

Non-SMS Volume(s) selected by Advanced Volume Selection
The number of non-SMS data sets for which volumes were selected at allocation time by Advanced Volume Selection.

Number of data set allocations failed by TERM_ALLOC
The number of data set allocations that were forced to fail as a result of the TERM_ALLOC function.

Creating a custom report utility

In addition to the **View Event History** option on the Tivoli Advanced Allocation Management Primary Option Menu and the batch reporting utility, Advanced Allocation Management provides record maps so that you can write a custom reporting utility in the language of your choosing.

The following record maps are provided:

- A DSECT to be used by assembly language programs to directly map the SMF records.
- A record description of the raw records that are generated when `DETAIL=(A or B)` and `FORMAT=L` are specified.

Although the latter record description is also in DSECT format, it can be tailored to any programming or report language syntax. Both record maps are generated during installation process and are located in the GLOSSMFR member of the Advanced Allocation Management sample library.

Note: The SMF record is composed of triplets that define multiple record sections. For more information about the use of triplets in SMF record processing, see *z/OS MVS System Management Facilities (SMF)*.

GLOSSMFR (DSECT)

The following figure shows an excerpt from the GLOSSMFR member, which shows a portion of the SMFREC DSECT.

```
*********************************************************************
SMFREC   DSECT
*
*        SMF RECORD HEADER SEGMENT
*
          DS   0D
SMFRECRPS DS   0CL72              FIXED LENGTH HEADER
SMFRECRHD DS   0CL24              RECORD HEADER SECTION
SMFRECLEN DC   FL2'0'             RECORD LENGTH
SMFRECSEG DC   BL2'0'             SEGMENT DESCRIPTOR
SMFRECFLG DC   BL1'0'             HEADER FLAG BYTE -
*                                 SYSTEM INDICATOR
*                                 BIT 0 - SUBSYSTEM IDENTIFICATION
*                                         FOLLOWS SYSTEM
*                                         IDENTIFICATION
*                                 BIT 1 - SUBTYPES USED
*                                 BIT 2   RESERVED
*                                 BIT 3-6 VERSION INDICATORS
*                                 BIT 7 - RESERVED
*                                 SEE 'STANDARD SMF RECORD
*                                 HEADER' FOR DETAILS.
SMFRECRTY DC   XL1'0'             RECORD TYPE
SMFRECTME DC   FL4'0'             TIME RECORD WAS MOVED
*                                 TO SMF BUFFER, IN HUNDREDTHS
*                                 OF A SECOND.
SMFRECDTE DC   PL4'0000'          DATE THAT THE RECORD
*                                 WAS MOVED TO THE SMF BUFFER,
*                                 IN THE FORM OCYYDDDF(WHERE F IS THE
*                                 SIGN). FOR A TSO SESSION, THE DATE
*                                 IS THE LOGOFF DATE.
SMFRECSID DC   CL4' '             SYSTEM IDENTIFICATION
SMFRECWID DC   CL4'0'             SUBSYSTEM IDENTIFIER
SMFRECSTP DC   FL2'0'             RECORD SUBTYPE
SMFRECSAR EQU  1                  1 = SUBSYSTEM ACTIVITY
SMFRECPAR EQU  3                  3 = PROCESSING ACTIVITY
*
*        SELF DEFINING SECTION
*
SMFRECSDS DS   0CL48              SELF DEFINING SECTION
*
SMFRECSOF DC   F'0'               OFFSET TO SUBSYSTEM ID SECTION
*                                 FROM START OF RECORD, INCLUDING
*                                 THE RDW
SMFRECSLN DC   FL2'0'             LENGTH OF SUBSYSTEM ID SECTION
SMFRECSON DC   FL2'0'             NUMBER OF SUBSYSTEM ID SECTIONS
*
SMFRECAOF DC   F'0'               OFFSET TO SUBSYSTEM ACTIVITY
*                                 SECTION FROM START OF RECORD,
*                                 INCLUDING THE RDW
SMFRECALN DC   FL2'0'             LENGTH OF SUBSYSTEM ACTIVITY
*                                 SECTION
SMFRECAON DC   FL2'0'             NUMBER OF SUBSYSTEM ACTIVITY
*                                 SECTIONS
*
SMFRECJOF DC   F'0'               OFFSET TO JOB IDENTIFICATION
*                                 SECTION FROM START OF RECORD,
*                                 INCLUDING THE RDW
SMFRECJLN DC   FL2'0'             LENGTH OF JOB IDENTIFICATION
*                                 SECTION
SMFRECJON DC   FL2'0'             NUMBER OF JOB IDENTIFICATION
*                                 SECTIONS
*
```

Figure 22. GLOSSMFR (SMFREC DSECT excerpt)

GLOSSMFR (detail record map)

The following figure shows the supplied FORMAT=L detail record map.

```
*************************************************************************
*        AOD1 RECORD DSECT                                              *
*************************************************************************
*
AODATA1  DSECT
*
AOD1RDW  DC    AL2(0,0)            RDW
AOD1RVER DC    CL2'03'             RECORD VERSION
AOD1DATE DC    CL10' '             DATE (YYYY/MM/DD FORMAT)
AOD1TIME DC    CL8' '              TIME (HH:MM:SS) FORMAT
         DC    CL4' '              RESERVED
AOD1SID  DC    CL4' '              MVS SYSTEM ID
         DC    CL8' '              RESERVED
AOD1SSID DC    CL4' '              ADVANCED ALLOCATION MGMT SUBSYS ID
AOD1ATYP DC    CL4' '              ADDRESS SPACE TYPE(JOB/STC/TSU/APPC)
AOD1JOBN DC    CL8' '              JOBNAME
         DC    CL8' '              RESERVED
AOD1STPN DC    CL8' '              STEPNAME
AOD1PSTP DC    CL8' '              PROCSTEPNAME
AOD1PGMN DC    CL8' '              PROGRAM
AOD1DSN  DC    CL44' '             DSNAME
AOD1DDN  DC    CL8' '              DDNAME
AOD1DSRG DC    CL2' '              DSORG (PS/PO/DA/VS)
AOD1PACT DC    CL65' '             PROCESSING ACTIVITY
         DC    CL15'00'            RESERVED
AOD1TRKR DC    CL7'0000000'        IF SPACE RELEASE; NUMBER OF TRACKS
*                                  RELEASED
         DC    CL1' '              RESERVED
*
AOD1NCTV DC    CL6' '              IF NOT CATLGD 2, FIRST
*                                  VOLSER OF DATA SET IN ERROR
*
AOD1NCTO DC    CL6' '              IF NOT CATLGD 2, FIRST
*                                  VOLSER OF OLD DATA SET
*
AOD1NCCD DC    CL4' '              ABEND CODE FOR NOTCAT2
*
AOD1NCNM DC    CL44' '             FOR NOTCAT2=RENAME, THE NEW
*                                  DATA SET NAME
*
AOD1SDSP DC    CL3' '              STATUS DISPOSITION -
*                                  (NEW/SHR/OLD/MOD)
AOD1NDSP DC    CL7' '              NORMAL DISPOSITION - (PASS/
*                                  KEEP/DELETE/CATLG/UNCATLG)
AOD1ADSP DC    CL7' '              ABNORMAL DISPOSITION -
*                                  (KEEP/DELETE/CATLG/UNCATLG)
AOD1SMSM DC    CL1' '              SMS-MANAGED DATA SET (Y/N)
AOD1TEMP DC    CL1' '              TEMPORARY DATA SET (Y/N)
AOD1SMDC DC    CL8' '              SMS DATA CLASS
AOD1SMSC DC    CL8' '              SMS STORAGE CLASS
AOD1SMMC DC    CL8' '              SMS MANAGEMENT CLASS
AOD1SMSG DC    CL8' '              SMS 1ST OR ONLY STORGRP NAME
AOD1LINE DC    CL8' '              MATCHING RULEDEF LINE NUMBER
*
AODATA1L EQU   *-AODATA1           LENGTH OF RECORD
*
* END OF AODATA MAPPING ....
*
```

Figure 23. FORMAT=L detail record map

Event recording and reporting filter criteria descriptions

The following filter criteria are available for use with Advanced Allocation Management event recording and reporting.

Value prefixes are allowed for the criteria as indicated. A *value prefix* is one or more characters followed by an asterisk that allows matching to occur on a portion of the value. For example, specifying `FILTER_SET=JOBNAME=AB*` or `FILTER_SET=JOBNAME=(AB*)` allows all jobs that have names that begin with AB, to pass filtering.

ADISP=value
Abnormal termination disposition. The following values are valid:

- KEEP
- DELETE
- CATLG
- UNCATLG

DATACLAS=value
SMS data class (1 - 8 characters). Value prefix is allowed.

DDNAME=value
ddname (1 - 8 characters). Value prefix is allowed.

DSNAME=value
Data set name (1 - 44 characters). Value prefix is allowed.

DSORG=value
Data set organization (2 characters). The following values are valid:

- PS (physical sequential)
- PO (partitioned organization)
- DA (direct access)
- VS (VSAM)
- KS (VSAM key-sequenced data set)
- RR (a VSAM variable-length or fixed-length relative-record data set)
- ES (VSAM entry-sequenced data set)
- LD (VSAM linear data set)
- NV (non-VSAM data set)

Note: If you specify `DSORG=VS`, all records that have a DSORG value of KS, ES, RR, or LD are also selected in addition to records that have the generic value of VS. Likewise, if you specify a DSORG of NV, all records that have a DSORG value of PS, PO, or DA are also selected in addition to records that have the generic value of NV.

ENDTIME=value
An end time value that is specified in yyyy/mm/dd-hh:mm:ss format where: *yyyy* = year, *mm* = month, *dd* = day, *hh* = hour, *mm* = minutes, and *ss* = seconds. To specify a time frame, use ENDTIME with STARTTIME.

Note: You must specify a hyphen (-) between the date and time values. For example: `2004/09/02-10:25:00`. Embedded spaces between the date and time values are not permitted.

JOBNAME=value
Job name (1 - 8 characters). Value prefix is allowed.

JOBTYPE=value
Job type. The following values are valid:

- JOB (a normal job address space)
- STC (started task)
- TSU (time-sharing user)
- APPC (APPC address space)

MGMTCLAS=value
SMS management class (1 - 8 characters). Value prefix is allowed.

NDISP=value
Normal termination disposition. The following values are valid:

- PASS
- KEEP
- DELETE
- CATLG
- UNCATLG

PROGRAM=value
Program name (1 - 8 characters). Value prefix is allowed.

PROCSTEPNAME=value
procstepname (1 - 8 characters). Value prefix is allowed.

SDISP=value
Status disposition. The following values are valid:

- NEW
- SHR
- OLD
- MOD

SMS-MANAGED=value
SMS management. The following values are valid:

- (Y)ES
- (N)O

STARTTIME=value
A start time value that is specified in yyyy/mm/dd hh:mm:ss format where: *yyyy* = year, *mm* = month, *dd* = day, *hh* = hour, *mm* = minutes, and *ss* = seconds. To specify a time frame, use STARTTIME with ENDTIME.

Note: You must specify a hyphen (-) between the date and time values. For example: `2014/09/02-10:25:00`. Embedded spaces between the date and time values are not permitted.

SUBSYS=value
Advanced Allocation Management subsystem ID (1 - 4 characters). Value prefix is allowed.

STEPNAME=value
Step name (1 - 8 characters). Value prefix is allowed.

STORCLAS=value
SMS storage class (1 - 8 characters). Value prefix is allowed.

STORGRP=value
SMS first or only storage group (1 - 8 characters). Value prefix is allowed.

SYSID=value
MVS system ID (1 - 4 characters). Value prefix is allowed.

TEMPDS=value
Temporary data set. The following values are valid:

- (Y)ES
- (N)O

Event recording and reporting sort field descriptions

The following sort fields are available for use with event recording and reporting.

ACTIVITY
Sort the report by Advanced Allocation Management processing action.

ADISP
Sort the report by abnormal termination disposition.

DATACLAS
Sort the report by SMS data class.

DATETIME
Sort the report by time frame.

DDNAME
Sort the report by ddname.

DSNAME
Sort the report by data set name.

DSORG
Sort the report by data set organization.

JOBNAME
Sort the report by job name.

JOBTYPE
Sort the report by job type.

MGMTCLAS
Sort the report by SMS management class.

NDISP
Sort the report by normal termination disposition.

PROGRAM
Sort the report by program name.

PROCSTEPNAME
Sort the report by procstepname.

SDISP
Sort the report by status disposition.

SMS-MANAGED
Sort the report by SMS-management specification.

STEPNAME
Sort the report by step name.

STORCLAS
Sort the report by SMS storage class.

STORGRP
Sort the report by SMS storage group.

SUBSYS
Sort the report by Advanced Allocation Management subsystem ID.

SYSID
Sort the report by MVS system ID.

TEMPDS
Sort the report by temporary data set specification.

ACTIVITY descriptions

The following table shows the functions and messages related to event recording and reporting activity.

Note: The reports that you generate can vary depending upon the options and filters that you select.

Table 15. ACTIVITY descriptions and related functions and messages

ACTIVITY description	Function	Messages
ALLOCATION PARAMETER SET BY ENHANCED ALLOCATION MANAGEMENT	Enhanced Allocation Management function (SET_* keywords)	GLO2194I, GLO3194I
DATA SET ALLOCATION FAILED BY THE TERM_ALLOC FUNCTION	TERM_ALLOC	GLO2999S
DATA SET INITIALIZED	DATASET_INIT	GLO2080I
DATA SET PROCESSED BY RULEDEF_TEST MODE	RULEDEF_TEST	GLO2190I, GLO3190I
INITIAL VOLUME PRIMARY SPACE ALLOCATION FAILURE RECOVERED	PRIM_REDUCE	GLO2118I, GLO3118I
INSUFFICIENT PDS DIRECTORY SPACE ERROR RECOVERED	PDS_DIR_INC	GLO2150I
INSUFFICIENT VOLUMES DEFINED ABEND RECOVERED (B37/E37)	VOL_ADD	GLO2128I, GLO3128I
NON-SMS VOLUME(S) SELECTED BY ADVANCED VOLUME SELECTION	Advanced Volume Selection (AVS)	GLO2148I, GLO3148I
NOT CATLGD 2 ERR PROCESSED WITH RENAME ON VOLSER *vvvvvv* AT *ttttt*	NOTCAT2 RENAME	GLO2085I, GLO2087I
NOT CATLGD 2 ERR PROCESSED WITH SCRATCH ON VOLSER *vvvvvv* AT *ttttt*	NOTCAT2 SCRATCH (UNCATALOG and DELETE)	GLO2083I, GLO2084I
NOT CATLGD 2 ERR PROCESSED WITH UNCATLG ON VOLSER *vvvvvv* AT *ttttt*	NOTCAT2 UNCATALOG	GLO2087I, GLO2089I
NOT CATLGD 2 ERROR DETECTED AND PROCESSED WITH JCLFAIL OPTION	NOTCAT2 JCLFAIL	GLO2051I, GLO2061I
POSSIBLE INSUFFICIENT SPACE ABEND AVOIDED (B37/E37)	SEC_INC	GLO2106I, GLO3106I
SPACE RELEASE OPTION ADDED TO PRIMARY OR SECONDARY ALLOCATION	SPACE_RELEASE	GLO2125I
SUBSEQUENT VOLUME PRIMARY SPACE ALLOCATION FAILURE RECOVERED	PRIM_REDUCE	GLO2144I, GLO3144I
UNDEFINED SECONDARY ALLOCATION SPACE ABEND AVOIDED (D37)	SEC_ALLOC	GLO2100I, GLO3100I
UNAVAILABLE SECONDARY SPACE ALLOCATION ABEND AVOIDED (B37/E37)	SEC_BEST	GLO2103I, GLO3103I
UNAVAILABLE SECONDARY SPACE ALLOCATION ABEND RECOVERED (B37/E37)	SEC_REDUCE	GLO2109I, GLO3109I

Chapter 11. SMF Reporting Utility

The Advanced Allocation Management SMF Reporting Utility generates a Projected Savings Report that you can use to help evaluate potential Advanced Allocation Management cost savings. The Advanced Allocation Management SMF Reporting Utility uses SMF record type 30 as input to the Projected Savings Report.

Note: Advanced Allocation Management does not need to be active to use the SMF Reporting Utility. The SMF Reporting Utility works with all versions of Advanced Allocation Management.

Configuring the SMF Reporting Utility JCL

To use the SMF Reporting Utility, you must configure the SMF Reporting Utility JCL. The sample JCL for the utility is generated during the installation process and is located in the GLOJSMFR member of the Advanced Allocation Management sample library.

Note:

- All lines are required.
- Numeric values must be specified as integers without commas or decimals. For example, to indicate an amount of 10,000, specify `10000`.

The sample GLOSMFRP SMF Reporting Utility JCL is shown below:

```
//*
//GLOSMFRP EXEC PGM=GLOSMFRP
//STEPLIB  DD   DSN=GLO.V3R3.SGLOLOAD,DISP=SHR
//CNTLOUT  DD   SYSOUT=*
//HISTOUT  DD   SYSOUT=*
//CNTLIN   DD   *
SMF_DATASET=SMF.DATA.SET1
MONETARY_UNIT=DOLLARS
CPU_TIME_COST_PER_HOUR=NNNNN
ELAPSED_TIME_COST_PER_HOUR=NNNNN
AVERAGE_ABEND_RECOVERY_TIME=NNNNN
ABEND_RECOVERY_COST_PER_HOUR=NNNNN
SORT_BY=DATETIME
```

To configure the JCL, complete the following steps:

1. Replace `GLO.V3R3` with the high-level qualifier of the installed Advanced Allocation Management load libraries.
2. Modify the required DD statements. For more information, see "Required DD statements."
3. Specify the names of one or more valid SMF data sets to search for Advanced Allocation Management SMF records.
4. Modify the input control statements. For more information, see "Valid input control statements" on page 308.
5. Submit the JCL to generate the Projected Savings Report.

Required DD statements

Modify the following required DD statements in the SMF Reporting Utility JCL.

HISTOUT
HISTOUT specifies where the abending jobs detail report is written. To ensure that the proper attributes are set (depending upon the functions that were requested), allow the utility to place file attributes on the data set at the time it is written.

CNTLOUT
CNTLOUT specifies where execution statistics and projected savings are written. It must have a record format of fixed (F) or fixed block (FB) and a record length (LRECL) of 80.

CNTLIN
CNTLIN specifies the location of the input control statements. It must have a record format of fixed (F) or fixed block (FB) and a record length (LRECL) of 80.

Valid input control statements

The following control statements are valid input to the SMF Reporting Utility. All statements are required unless otherwise noted.

Note:

- All control statements must start in card column one. (This restriction does not apply to continuation statements.)
- Blank lines are not permitted.

AVERAGE_ABEND_RECOVERY_TIME
Specify the average number of minutes it takes to recover each space abend. The maximum value is 99999.

ABEND_RECOVERY_COST_PER_HOUR
Specify the cost per hour of the person that is recovering each space abend. The maximum value is 99999.

CPU_TIME_COST_PER_HOUR
Specify the estimated cost for each hour of CPU time lost. The maximum value is 99999.

ELAPSED_TIME_COST_PER_HOUR
Specify the estimated cost of each hour of elapsed time lost. The maximum value is 99999.

MONETARY_UNIT
Specify the local currency unit. You can specify a maximum of 16 characters. The default setting is `MONETARY_UNIT=DOLLARS`.

SMF_DATASET
Specify the names of the SMF data sets to be searched for Advanced Allocation Management event records. A maximum of 32 data set names can be specified.

Each of the following SMF_DATASET= specifications are valid formats for specifying data set names using SMF_DATASET:

```
SMF_DATASET=SMF.data.set1

SMF_DATASET=(SMF.data.set1)

SMF_DATASET=(SMF.data.set1,SMF.data.set2)

SMF_DATASET=(SMF.data.set1,
       SMF.data.set2)
```

```
SMF_DATASET=(
       SMF.data.set1,
       SMF.data.set2
      )
```

Notes:

1. The SMF data set you specify must be a non-VSAM data set.
2. Separate multiple dsnames that are specified on a SMF_DATASET statement with commas, even if they are specified on separate lines. You do not need to specify a comma after the last dsname.

SORT_BY

The column by which the selected records are to be sorted. Specify one of the following columns: SYS, JOBNAME, STEPNAME, PROGRAM, CMP, DATETIME, CPUTIME, or ELAPSEDTIME.

Note: The SORT_BY value is optional. If the SORT_BY value is not specified, the records are sorted by the DATETIME column.

Example Projected Savings Report

The following information is provided in the Advanced Allocation Management Projected Savings Report.

Report options in effect

The REPORT OPTIONS IN EFFECT section displays the report options that were configured in the SMF Reporting Utility JCL. For example:

```
TIVOLI ADVANCED ALLOCATION MANAGEMENT PROJECTED SAVINGS REPORT

REPORT CREATED ON 2013/05/07 09:54:36

REPORT OPTIONS IN EFFECT:
  DATA SET CONTAINING SMF RECORDS
    SMF_DATASET                   : TSJRH.GLOSMFRP.SMFDATA.MAY0613
  MONEY
    MONETARY_UNIT                 : EUROS
  CPU TIME LOST COST BASIS
    CPU_TIME_COST_PER_HOUR        : 1000 EUROS
  ELAPSED TIME LOST COST BASIS
    ELAPSED_TIME_COST_PER_HOUR    : 1000 EUROS
  AVERAGE ABEND RECOVERY TIME
    AVERAGE_ABEND_RECOVERY_TIME   : 10 MINUTES
  ABEND RECOVERY COST BASIS
    ABEND_RECOVERY_COST_PER_HOUR  : 100 EUROS
  ABEND DETAIL REPORT SORTED BY
    SORT_BY                       : DATETIME
```

Totals for the reporting period

The TOTALS FOR THE REPORTING PERIOD section displays the total estimated monetary cost of the time that is lost due to recoverable space abends. For example:

```
TOTALS FOR THE REPORTING PERIOD
  TOTAL CPU TIME LOST             : 0:00:00.96
  TOTAL COST CPU TIME LOST        : 0 EUROS
  TOTAL ELAPSED TIME LOST         : 0:00:12.86
  TOTAL COST ELAPSED TIME LOST    : 4 EUROS
  TOTAL NUMBER RECOVERABLE ABENDS : 36
```

```
  TOTAL RECOVERY TIME LOST        : 6:00:00.00
  TOTAL COST RECOVERY TIME LOST   : 600 EUROS

  TOTAL COST FOR REPORTING PERIOD : 604 EUROS
```

Projected annualized savings

The PROJECTED ANNUALIZED SAVINGS section displays the total projected annualized monetary savings that are related to CPU time, elapsed time, and recovery time that might potentially be saved by avoiding space abends. For example:

```
PROJECTED ANNUALIZED SAVINGS BASED ON SMF DATA REPORTING
 PERIOD FROM 2013/05/06 07:49:43 TO 2013/05/06 19:23:57
  TOTAL YEARLY COST CPU LOST      : 202 EUROS
  TOTAL YEARLY COST ELAPSED LOST  : 2,704 EUROS
  TOTAL YEARLY COST RECOVERY LOST : 454,200 EUROS

  TOTAL COST FOR ALL TIME LOST    : 457,106 EUROS
```

Recoverable abends summary

The RECOVERABLE ABENDS SUMMARY section displays a breakdown of costs per abend type. For example:

```
RECOVERABLE ABENDS SUMMARY
  ABEND B37/E37 - INSUFFICIENT SPACE ABENDS
    TOTAL NUMBER RECOVERABLE ABENDS : 10
    TOTAL CPU TIME LOST             : 0:00:00.20
    TOTAL COST CPU TIME LOST        : 0 EUROS
    TOTAL ELAPSED TIME LOST         : 0:00:06.28
    TOTAL COST ELAPSED TIME LOST    : 2 EUROS
    TOTAL RECOVERY TIME LOST        : 1:40:00.0
    TOTAL COST RECOVERY TIME LOST   : 167 EUROS
    TOTAL COST FOR REPORTING PERIOD : 169 EUROS

  ABEND D37 - UNDEFINED SECONDARY SPACE ABENDS
    TOTAL NUMBER RECOVERABLE ABENDS : 14
    TOTAL CPU TIME LOST             : 0:00:00.14
    TOTAL COST CPU TIME LOST        : 0 EUROS
    TOTAL ELAPSED TIME LOST         : 0:00:01.02
    TOTAL COST ELAPSED TIME LOST    : 0 EUROS
    TOTAL RECOVERY TIME LOST        : 2:20:00.0
    TOTAL COST RECOVERY TIME LOST   : 233 EUROS
    TOTAL COST FOR REPORTING PERIOD : 233 EUROS
  ABEND B14 - PDS DIRECTORY ABENDS
    TOTAL NUMBER RECOVERABLE ABENDS : 12
    TOTAL CPU TIME LOST             : 0:00:00.62
    TOTAL COST CPU TIME LOST        : 0 EUROS
    TOTAL ELAPSED TIME LOST         : 0:00:05.56
    TOTAL COST ELAPSED TIME LOST    : 2 EUROS
    TOTAL RECOVERY TIME LOST        : 2:00:00.0
    TOTAL COST RECOVERY TIME LOST   : 200 EUROS
    TOTAL COST FOR REPORTING PERIOD : 202 EUROS


PROCESSING IS COMPLETE; RETURN CODE IS 00
```

Recoverable abend details report

The RECOVERABLE ABEND DETAILS REPORT section displays statistics that are based on the SMF records that are specified as input to the report on the SMF_DATASET parameter. For example:

```
SYS  JOBNAME  STEPNAME PROGRAM  CMP  DATETIME            CPUTIME     ELAPSEDTIME
---  -------  -------- -------  ---- --------            -------     -----------
TST1 TSJRHAB1 STEPJAN  IEBDG    SD37 2013/05/06 19:14:28 00:00:00.01 00:00:00.22
TST1 TSJRHAB2 STEPJAN  IEBDG    SD37 2013/05/06 19:14:36 00:00:00.01 00:00:00.06
TST1 TSJRHAB3 DIRINC   IEBDG    SB14 2013/05/06 19:14:44 00:00:00.07 00:00:00.75
TST1 TSJRHAB4 STEPJAN  IEBDG    SD37 2013/05/06 19:15:08 00:00:00.01 00:00:00.05
TST1 TSJRHAB5 STEPJAN  IEBDG    SD37 2013/05/06 19:15:17 00:00:00.01 00:00:00.09
TST1 TSJRHAB6 DIRINC   IEBDG    SB14 2013/05/06 19:15:25 00:00:00.05 00:00:00.66
TST1 TSJRHAB7 STEPJAN  IEBDG    SD37 2013/05/06 19:15:30 00:00:00.01 00:00:00.06
TST1 TSJRHAB8 STEPJAN  IEBDG    SD37 2013/05/06 19:15:35 00:00:00.01 00:00:00.07
TST1 TSJRHAB9 DIRINC   IEBDG    SB14 2013/05/06 19:15:41 00:00:00.05 00:00:00.44
TST1 TSJRHABA STEPJAN  IEBDG    SD37 2013/05/06 19:15:54 00:00:00.01 00:00:00.05
TST1 TSJRHABB STEPSB37 IEBDG    SB37 2013/05/06 19:15:58 00:00:00.02 00:00:00.77
TST1 TSJRHABC STEPSB37 IEBDG    SB37 2013/05/06 19:16:01 00:00:00.02 00:00:00.53
TST1 TSJRHAB1 STEPJAN  IEBDG    SD37 2013/05/06 19:17:08 00:00:00.01 00:00:00.08
TST1 TSJRHAB2 STEPJAN  IEBDG    SD37 2013/05/06 19:17:13 00:00:00.01 00:00:00.07
TST1 TSJRHAB3 DIRINC   IEBDG    SB14 2013/05/06 19:17:17 00:00:00.05 00:00:00.48
TST1 TSJRHAB4 STEPJAN  IEBDG    SD37 2013/05/06 19:17:25 00:00:00.01 00:00:00.05
TST1 TSJRHAB5 STEPJAN  IEBDG    SD37 2013/05/06 19:17:29 00:00:00.01 00:00:00.05
TST1 TSJRHAB6 DIRINC   IEBDG    SB14 2013/05/06 19:17:34 00:00:00.05 00:00:00.43
TST1 TSJRHAB7 STEPJAN  IEBDG    SD37 2013/05/06 19:17:40 00:00:00.01 00:00:00.06
TST1 TSJRHAB8 STEPJAN  IEBDG    SD37 2013/05/06 19:17:45 00:00:00.01 00:00:00.06
TST1 TSJRHAB9 DIRINC   IEBDG    SB14 2013/05/06 19:17:49 00:00:00.05 00:00:00.55
TST1 TSJRHABA STEPJAN  IEBDG    SD37 2013/05/06 19:17:57 00:00:00.01 00:00:00.05
TST1 TSJRHABB STEPSB37 IEBDG    SB37 2013/05/06 19:18:01 00:00:00.02 00:00:00.80
TST1 TSJRHABC STEPSB37 IEBDG    SB37 2013/05/06 19:18:06 00:00:00.02 00:00:00.48
TST1 TSJRHAB3 DIRINC   IEBDG    SB14 2013/05/06 19:18:46 00:00:00.05 00:00:00.44
TST1 TSJRHAB6 DIRINC   IEBDG    SB14 2013/05/06 19:18:51 00:00:00.05 00:00:00.36
TST1 TSJRHAB9 DIRINC   IEBDG    SB14 2013/05/06 19:18:55 00:00:00.05 00:00:00.28
TST1 TSJRHAB3 DIRINC   IEBDG    SB14 2013/05/06 19:19:03 00:00:00.05 00:00:00.39
TST1 TSJRHAB6 DIRINC   IEBDG    SB14 2013/05/06 19:19:11 00:00:00.05 00:00:00.28
TST1 TSJRHAB9 DIRINC   IEBDG    SB14 2013/05/06 19:19:17 00:00:00.05 00:00:00.50
TST1 TSJRHABB STEPSB37 IEBDG    SB37 2013/05/06 19:20:17 00:00:00.02 00:00:00.86
TST1 TSJRHABC STEPSB37 IEBDG    SB37 2013/05/06 19:20:21 00:00:00.02 00:00:00.46
TST1 TSJRHABB STEPSB37 IEBDG    SB37 2013/05/06 19:20:26 00:00:00.02 00:00:00.72
TST1 TSJRHABC STEPSB37 IEBDG    SB37 2013/05/06 19:20:29 00:00:00.02 00:00:00.51
TST1 TSJRHABB STEPSB37 IEBDG    SB37 2013/05/06 19:20:35 00:00:00.02 00:00:00.68
TST1 TSJRHABC STEPSB37 IEBDG    SB37 2013/05/06 19:20:40 00:00:00.02 00:00:00.47

PROCESSING IS COMPLETE; RETURN CODE IS 00
```

Chapter 12. Frequently asked questions

Consult the following questions and answers to help you understand more about using Advanced Allocation Management.

Installation and configuration

Consult the following information to help you understand more about installing and configuring Advanced Allocation Management.

Question: Are there special sizing considerations for the Advanced Allocation Management control data set?

Answer: Because the Advanced Allocation Management control data set contains relatively little data, there are no special sizing considerations. The space values that are supplied with the installation JCL are sufficient.

Definition management

Consult the following questions and answers to help you understand more about Advanced Allocation Management definition management.

Question: Is it true that an INCLUDE JOBNAME=*xxxxxxx* definition by itself will allow processing for that job name with all of the defaults that are specified at the beginning of the parameter member?

Answer: Yes. The only time that you must specify function keywords and values after selection criteria is when you want to specify a value other than the default values. Otherwise, all defaults are applied.

Question: Rule definitions are processed in a top-down order. Does this mean that as soon as a match is found, the process exits and does not process the remaining parameters?

Answer: Yes. The only exception is when you specify the CONTINUE keyword. The CONTINUE keyword specifies that RULEDEFS processing should continue even if the current rule definition encounters a match. Because all of the Advanced Allocation Management definitions are specified together with the selection criteria, there is no multiple-member evaluation.

Question: If you have multiple programs or job names that you want to include or exclude from Advanced Allocation Management processing, do you need separate INCLUDE and EXCLUDE statements for each one? In other words, you cannot combine more than one program in an INCLUDE or EXCLUDE statement.

Answer: Some selection criteria keywords support the specification of multiple operands. See “Selection criteria keywords” on page 180 to determine if the selection criterion you want to use supports multiple operands. For examples of INCLUDE and EXCLUDE statements, see “INCLUDE” on page 49 and “EXCLUDE” on page 52.

Question: If I specify a series of INCLUDE statements, and then add a function keyword (`DATASET_INIT=ENABLE`, for example) after the last one, does the function keyword apply to all of the preceding INCLUDE statements?

Answer: No. Function-related keywords must be inserted after each INCLUDE statement to which you want them to apply. In the situation you describe, any INCLUDE statement that matches the data set that is being allocated will, in most cases, cause Advanced Allocation Management to stop reading rule definitions and apply the functions that are specified in the DEFAULTS statement, because the INCLUDE statement does not specify a function that overrides the functions on the DEFAULTS statement.

Note: The CONTINUE function directs Advanced Allocation Management to perform applicable actions for a matched rule definition and then continue to look for additional matching rule definitions.

For more information, see "INCLUDE" on page 49 and "EXCLUDE" on page 52

Using Advanced Allocation Management

Consult the following questions and answers when you use Advanced Allocation Management.

Question: What would be the benefit (or need) of running multiple Advanced Allocation Management subsystems in a single system environment? Under what circumstances is the use of multiple Advanced Allocation Management subsystems beneficial?

Answer: The most common reason is to run production and test versions concurrently.

Question: The SMS data class has a Space Constraint Relief attribute that I can use to reduce space up to a specific percentage. This definition enables me to reduce primary allocation of the data sets. However, if I use the Advanced Allocation Management PRIM_REDUCE function for an SMS-managed data set, which definition is given priority, the SMS definition or the Advanced Allocation Management definition?

Answer: Advanced Allocation Management handles SMS and non-SMS-managed data sets. When you use SMS, think of Advanced Allocation Management as an extension to DFSMS. In other words, when SMS definitions are defined, those definitions are implemented first and then the Advanced Allocation Management definitions take effect.

Chapter 13. Troubleshooting

Consult the following troubleshooting topics to help diagnose and resolve any problems that you might encounter.

Disabling product-level intercepts

When the product-level intercepts are enabled, Advanced Allocation Management performs processing. When you disable the intercepts, the Advanced Allocation Management started task remain active without performing processing.

Disabling the intercepts can be helpful in situations when you do not want to shut down Advanced Allocation Management completely. For example, if Advanced Allocation Management is running and processing is not occurring as expected, you can disable the intercepts to help isolate the cause of the problem and collect information (such as dumping the address space) that you can use to troubleshoot the problem.

Note: Disabling or enabling the product-level intercepts disables or enables the intercepts for *all* active Advanced Allocation Management subsystems.

Bypassing specific processing

When Advanced Allocation Management does not operate as expected for a particular job or started task, and the process is critical to production processing, you can force Advanced Allocation Management to bypass the area of processing in question without shutting down the product entirely for all other processing.

To force Advanced Allocation Management to bypass processing for all of the data sets that are accessed by the step, add a bypass ddname to any job step JCL. The ddname is AOBYPASS and the format of the DD statement is:

```
//AOBYPASS DD DUMMY
```

Add the statement anywhere after the EXEC statement of a step and before the EXEC statement of the next step or the end of job indicator (//). For example:

```
//PDUSER#2 JOB (ACCT#),'PDUSER',
//         MSGCLASS=X,CLASS=A,NOTIFY=&SYSUID
//*
//JOBLIB  DD  DISP=SHR,
//            DSN=DSN.V710.SDSNLOAD
//*
//DSNTEP2 EXEC PGM=IKJEFT01,DYNAMNBR=20
//AOBYPASS DD DUMMY
//SYSTSPRT DD  SYSOUT=*
.
 -- PROCESSING TO BYPASS -----
.
//DSNTEP3 EXEC PGM=IKJEFT01,DYNAMNBR=20
```

By using the bypass ddname, you can avoid the processing in question without having to perform extensive time-consuming modifications to the rule definitions. You can also use the bypass ddname to systematically isolate specific areas of processing to locate the cause of a particular problem.

Note: Because DB2 performs space management for its own data sets, Advanced Allocation Management automatically bypasses processing for DB2 allocated data sets. This eliminates the creation of unnecessary control blocks and thus, the CPU time that is associated with managing those allocations.

Identifying a rule definition match

Specify the AOMATCH DD statement to easily identify a matching Advanced Allocation Management rule definition at the job step level.

The format of the statement is:

```
//AOMATCH DD DUMMY
```

Add the AOMATCH DD statement anywhere after the EXEC statement in a job step. For example:

```
//PDUSER#2 JOB (ACCT#),'PDUSER',
//         MSGCLASS=X,CLASS=A,NOTIFY=&SYSUID
//*
//JOBLIB  DD  DISP=SHR,
//            DSN=DSN.V710.SDSNLOAD
//*
//DSNTEP2 EXEC PGM=IKJEFT01,DYNAMNBR=20
//AOMATCH DD DUMMY
//SYSTSPRT DD  SYSOUT=*
.
.
//DSNTEP3 EXEC PGM=IKJEFT01,DYNAMNBR=20
```

During Advanced Allocation Management processing, Advanced Allocation Management generates a message that identifies the matching rule definition number for the first line of the matching rule definition for all of the data sets in that step.

For non-VSAM processing, the following messages are issued:

```
GLO2188I MATCHING INCLUDE RULEDEF LINE NUMBER = 280 - RG33
GLO2123I GLOJIVP1,SECALLOC,,SYSUT2,NSMS.PDABC.GLO.TEST.DATA
```

For VSAM processing, the messages are the same except that the message IDs are GLO3188I and GLO3123I.

To identify the rule definition that matched the event, look for the reported rule definition line number in the job log. The Advanced Allocation Management started task job log lists all of the activated rule definitions and volume group definitions and provides a line number for each line. For example:

```
GLO4199I    280  INCLUDE JOBNAME=GLOJIVP1                 * INCLUDE OR EXCLUDE          00370100 -RG33
GLO4199I    281                                                                        00370200 -RG33
GLO4199I    282          DDNAME=SYSUT2                   * DDNAME                     00370300 -RG33
GLO4199I    283          JOBTYPE=JOB                     * JOB TYPE                   00370400 -RG33
GLO4199I    284          PRIM_REDUCE=(ENABLE,ENABLE)     * PRIMARY ALLOCATION INITIAL 00370500 -RG33
GLO4199I    285                                          *    REDUCTION FOR INITIAL   00370600 -RG33
GLO4199I    286                                          *    AND VOLUME ADD ALLOCS   00370700 -RG33
GLO4199I    287          PRIM_REDUCE_LIM=(0,0)           * PRIMARY REDUCTION LIMIT FOR 00370800 -RG33
GLO4199I    288                                          *    INITIAL AND VOLUME ADD  00370900 -RG33
GLO4199I    289                                          *    ALLOCS                  00371000 -RG33
GLO4199I    290          PRIM_REDUCE_INC=(99,99)         * PRIMARY REDUCTION INCREMENT 00372000 -RG33
GLO4199I    291                                          *    FOR INITIAL AND VOLUME  00373000 -RG33
GLO4199I    292                                          *    ADD ALLOCS              00374000 -RG33
GLO4199I    293          SEC_ALLOC=ENABLE                * SECONDARY ALLOCATION       00375000 -RG33
```

```
| GLO4199I    294                                          *    ADDITION                 00376000 -RG33
| GLO4199I    295          SEC_ALLOC_AMT=100               * SECONDARY ALLOCATION       00377000 -RG33
| GLO4199I    296                                          *    AMOUNT AS A PERCENTAGE   00378000 -RG33
```

To determine the matching rule definition, match the line number that is reported in the message to the corresponding line for the last activated rule definition. (Because it is common for Advanced Allocation Management to receive control multiple times for a particular allocation, multiple instances of the messages might be issued.)

Generating trace data

When Advanced Allocation Management does not operate as expected for a particular job or started task, you can add a trace ddname to any job step JCL to generate trace data for diagnostic purposes.

The ddname is AOTRACE and the format of the DD statement is:

```
//AOTRACE DD DUMMY
```

Add the AOTRACE DD statement anywhere after the EXEC statement of a step and before the EXEC statement of the next step or the end of job indicator (//). For example:

```
//PDUSER#2 JOB (ACCT#),'PDUSER',
//         MSGCLASS=X,CLASS=A,NOTIFY=&SYSUID
//*
//JOBLIB  DD  DISP=SHR,
//            DSN=DSN.V710.SDSNLOAD
//*
//DSNTEP2 EXEC PGM=IKJEFT01,DYNAMNBR=20
//AOTRACE DD DUMMY
//SYSTSPRT DD  SYSOUT=*
.
.
//DSNTEP3 EXEC PGM=IKJEFT01,DYNAMNBR=20
```

Note: Use the AOTRACE DD only at the direction of IBM Software Support.

Gathering diagnostic information

Before you report a problem with Advanced Allocation Management to IBM Software Support, you need to gather the appropriate diagnostic information.

Provide the following information for all Advanced Allocation Management problems:

- A clear description of the problem and the steps that are required to re-create the problem
- All messages that were issued as a result of the problem
- Product release number and the number of the last program temporary fix (PTF) that was installed
- The version of z/OS that you are using and the type and version of the operating system that you are using

Provide additional information based on the type of problem that you experienced:

For online abends, provide the following information

- A screen shot of the panel that you were using when the abend occurred
- The job log from the TSO session that encountered the abend

- The job log from the server
- A description of the task that you were doing before the abend occurred

For errors in batch processing, provide the following information

- The complete job log
- Print output
- Contents of the any data sets that were used during the processing

Appendix A. Operator commands

Advanced Allocation Management operator commands enable authorized users to perform selected operations.

Advanced Allocation Management offers several different types of operator commands that can be used to display the status of Advanced Allocation Management, to enable and disable certain functions, and to dynamically alter processing without requiring a shutdown or quiesce of the product.

Note: The names of volume group definitions members can be specified as parameters on operator commands.

Issuing commands

Operator commands are issued using the MVS **MODIFY** command. Therefore, prefix the commands with MODIFY *stcname*, (where *stcname* is the name of the started task) followed immediately by the command.

Note: Operator commands must be entered from an MVS operator console or using a facility that issues MVS commands, such as SDSF.

The following are examples of operator commands:

MODIFY *stcname*,ACTIVATE RULEDEFS *member*

or

F *stcname*,ACTIVATE RULEDEFS *member*

Display commands

Display commands provide information about Advanced Allocation Management, such as the current subsystem and whether the product intercepts are enabled on the issuing operator's console. These commands do not affect Advanced Allocation Management configuration.

DIAGNOSE
Display diagnostic information.

Note: Use the **DIAGNOSE** command only at the direction of IBM Software Support.

DISPLAY SUBSYSTEM
Display information about the current Advanced Allocation Management subsystem.

DISPLAY SUBSYSTEM ALL
Display information about all Advanced Allocation Management subsystems on the current z/OS image.

DISPLAY RULEDEFS
Display information (the rule definitions member name that is active and the date and time it was activated) about the rule definitions for the current Advanced Allocation Management subsystem.

DISPLAY VGRPDEFS
Display information (the volume group definitions member name that is active and the date and time it was activated) about the volume group definitions for the current Advanced Allocation Management subsystem.

DISPLAY VARDEFS
Display information about the user-defined variable definitions for the current Advanced Allocation Management subsystem.

DISPLAY INTERCEPTS
Display whether the product-level intercepts are enabled or disabled.

Note: There is one set of intercepts for any and all subsystems on a z/OS image.

DISPLAY ALL
Perform the equivalent actions of the following operator commands:

- **DISPLAY INTERCEPTS**
- **DISPLAY SUBSYSTEM**
- **DISPLAY RULEDEFS**
- **DISPLAY VGRPDEFS**
- **DISPLAY VARDEFS**

Enable and disable commands

Enable and disable commands enable or disable specific Advanced Allocation Management functions.

ENABLE INTERCEPTS
Enable the product-level intercepts.

ENABLE SUBSYSTEM
Enable the subsystem for processing.

DISABLE INTERCEPTS
Disable the product-level intercepts.

DISABLE SUBSYSTEM
Disable the subsystem for processing. Disabling the subsystem prevents it from participating in new events, such as out-of-space conditions.

Dynamic configuration commands

With dynamic configuration commands, you can dynamically modify Advanced Allocation Management rule or volume group definitions at any time without performing a shutdown or quiesce of the product.

ACTIVATE RULEDEFS *member*
Activate a new or updated rule definitions member. Advanced Allocation Management reads the contents of the specified rule definitions member and then creates the appropriate control blocks.

ACTIVATE VGRPDEFS *member*
Activate a new or updated volume group definitions member. Advanced Allocation Management reads the contents of the specified volume group definitions member and then creates the appropriate control blocks.

SETVARDEFS *membername*
Specify the variables definitions member to use.

Appendix B. Subsystem options modification and display reference

The following table shows how the Advanced Allocation Management subsystem options can be modified and where the options are displayed (on the user interface or console).

Table 16. Subsystem options modification and display reference

Option	Modify from user interface?	Modify using operator cmd?	Specify in OPTIONS member?	Specify in start parameters?	Display in user interface?	Display on console?
DSN_PATTERN_MATCHING	No	No	Yes	No	No	Yes
S1_INTERCEPT	No	No	Yes	No	No	Yes
SSID	No	No	Yes	No	Yes	Yes
SECURITY_CLASS	No	No	Yes	No	No	No
(Name of active RULEDEFS)	Yes	Yes	Yes	No	Yes	Yes
(Name of active VGRPDEFS)	Yes	Yes	Yes	No	Yes	Yes
(Name of active VARDEFS)	Yes	Yes	Yes	No	Yes	Yes

Appendix C. Function and selection criteria environment compatibility

The following table shows the function and selection criteria that are permitted in each environment.

Table 17. Function and selection criteria environment compatibility

Keyword	ACS	ALLOCATE	EXTEND	UNALLOCATE	STOW	CATCHECK
ACCBIAS		x	x	x		
ADISP		x	x	x	x	x
AVS		x				
AVS_ALGORITHM		x				
AVS_MOUNT_USAGE		x				
AVS_UNIT		x				
AVS_VGRP		x				
AVS_VGRP_SEARCH		x				
AVS_VSER		x				
BLKSIZE		x	x	x	x	x
BUFSPC		x	x	x		
CISZ		x	x	x		
CISZ_DATA		x	x	x		
CISZ_INDEX		x	x	x		
CONTIGUOUS		x	x	x	x	x
CONTINUE	x	x	x	x	x	x
DATACLAS	x	x	x	x	x	x
DATASET_INIT				x		
DDNAME	*3	*3	x	x	x	x
DEFERMOUNT		x	x	x		x
DEVICECLAS		x	x	x	x	x
DIAG	x	x	x	x	x	x
DIRBLKS		x	x	x	x	x
DSNAME	x	x	x	x	x	x
DSNAME_DATA		x	x	x		
DSNAME_INDEX		x	x	x		
DSNAME_LASTNODE	x	x	x	x	x	x
DSNAME_QUALIFIER	x	x	x	x	x	x
DSNAME_TYPE	x	x	x	x	x	x
DSORG	x	x	x	x	x	x
ENVIRONMENT	x	x	x	x	x	x
EATTR	x	x	x	x		x
EXPDATE	x	x	x	x	x	x
EXTADDR		x	x	x		
FILESEQNO		x	x	x		x
FRLOG		x	x	x		
FSPC_CA		x	x	x		

Table 17. Function and selection criteria environment compatibility (continued)

Keyword	ACS	ALLOCATE	EXTEND	UNALLOCATE	STOW	CATCHECK
FSPC_CI		x	x	x		
GROUPID	x	x	x	x	x	x
GUARSPACE	x	x	x	x	x	x
ILOAD		x	x	x		
JOBACCTn	x	x	x	x	x	x
JOBCLASS	*1	x	x	x	x	x
JOBNAME	x	x	x	x	x	x
JOBTYPE	x	x	x	x	x	x
KEY_LEN		x	x	x		
KEY_OFF		x	x	x		
LOGGING		x	x	x		
LOGSTREAM_ID		x	x	x		
LRECL		x	x	x		
MAXSPACE	x	x	x	x	x	x
MESSAGES	x	x	x	x	x	x
MGMTCLAS	x	x	x	x	x	x
MOUNT_USAGE		x	x	x	x	x
MSG_DESTINATION	x	x	x	x	x	x
MSG_LEVEL	x	x	x	x	x	x
NDISP		x	x	x	x	x
NOTCAT2						x
NOTCAT2_CHECK						x
NOTCAT2_DYNAL						x
NOTCAT2_QUALIFIER						x
NOTCAT2_QUALIFIER_POSITION						x
NOTCAT2_SUFFIX						x
NUMVOLS		x	x	x		
NUMVOLS_DATA		x	x	x		
NUMVOLS_INDEX		x	x	x		
OWNERID		x	x	x	x	x
PDS_DIR_INC					x	
PRIM_REDUCE		x	x			
PRIM_REDUCE_INC		x	x			
PRIM_REDUCE_LIM		x	x			
PQTY	x	x	x	x	x	x
PQTY_DATA		x	x	x		
PQTY_INDEX		x	x	x		
PROCSTEPNAME	*2	x	x	x	x	x
PROGRAM	x	x	x	x	x	x
PSPACE_KB	x	x	x	x	x	x
PSPACE_MB	x	x	x	x	x	x
RECFM		x	x	x	x	x
RELEASESPACE		x	x	x	x	x
RETPD	x	x	x	x	x	x
REUSE		x	x	x		

Table 17. Function and selection criteria environment compatibility (continued)

Keyword	ACS	ALLOCATE	EXTEND	UNALLOCATE	STOW	CATCHECK
RULEDEF_TEST	x	x	x	x	x	x
SDISP		x	x	x	x	x
SEC_ALLOC			x			
SEC_ALLOC_AMT			x			
SEC_BEST			x			
SEC_BEST_LIM			x			
SEC_INC			x			
SEC_INC_AMT			x			
SEC_INC_VOLNO			x			
SEC_INC_XTNT			x			
SEC_REDUCE			x			
SEC_REDUCE_INC			x			
SEC_REDUCE_LIM			x			
SET_ACCBIAS		x				
SET_ADISP		x				
SET_BLKSIZE		x				
SET_BUFSPC		x				
SET_CISZ		x				
SET_CISZ_DATA		x				
SET_CISZ_INDEX		x				
SET_CONTIGUOUS		x				
SET_DATACLAS	x					
SET_DEFERMOUNT		x				
SET_DIRBLKS		x				
SET_DSNAME_TYPE		x				
SET_DSORG		x				
SET_EATTR		x				
SET_EXPDATE		x				
SET_EXTADDR		x				
SET_FILESEQNO		x				
SET_FRLOG		x				
SET_FSPC_CA		x				
SET_FSPC_CI		x				
SET_ILOAD		x				
SET_KEY_LEN		x				
SET_KEY_OFF		x				
SET_LOGSTREAM_ID		x				
SET_LRECL		x				
SET_MGMTCLAS	x					
SET_NDISP		x				
SET_NUMVOLS		x				
SET_NUMVOLS_DATA		x				
SET_NUMVOLS_INDEX		x				
SET_PQTY		x				
SET_PQTY_DATA		x				

Table 17. Function and selection criteria environment compatibility (continued)

Keyword	ACS	ALLOCATE	EXTEND	UNALLOCATE	STOW	CATCHECK
SET_PQTY_INDEX		x				
SET_RECFM		x				
SET_RETPD		x				
SET_REUSE		x				
SET_SHROPT_XRGN		x				
SET_SHROPT_XSYS		x				
SET_SPACEUNITS		x				
SET_SPACEUNITS_DATA		x				
SET_SPACEUNITS_INDEX		x				
SET_SPANNED		x				
SET_SQTY		x				
SET_SQTY_DATA		x				
SET_SQTY_INDEX		x				
SET_STORCLAS	x					
SET_STORGRP	x					
SET_TAPE_COMP		x				
SET_TAPE_LABEL		x				
SET_UNIT_AFFINITY		x				
SET_UNIT_COUNT		x				
SET_UNIT_NAME		x				
SET_VOLCT		x				
SET_VSAM_SEPARATE		x				
SHROPT_XRGN		x	x	x		
SHROPT_XSYS		x	x	x		
SMS_MANAGED	x	x	x	x	x	x
SPACE_RELEASE				x		
SPACEUNITS	x	x	x	x	x	x
SPACEUNITS_DATA		x	x	x		
SPACEUNITS_INDEX		x	x	x		
SPANNED		x	x	x		
SQTY	x	x	x	x	x	x
SQTY_DATA		x	x	x		
SQTY_INDEX		x	x	x		
STEPACCTn	x	x	x	x	x	x
STEPNAME	*2	x	x	x	x	x
STORCLAS	x	x	x	x	x	x
STORGRP	x	x	x	x	x	x
SUBSYS	x	x	x	x	x	x
SYSID	x	x	x	x	x	x
SYSPLEXNAME	x	x	x	x	x	x
TAPE_COMP		x		x		x
TAPE_LABEL		x	x	x		x
TEMPDS		x	x	x	x	x
TERM_ALLOC	x	x		x	x	x
UNIT_AFFINITY		x	x	x	x	x

Table 17. Function and selection criteria environment compatibility (continued)

Keyword	ACS	ALLOCATE	EXTEND	UNALLOCATE	STOW	CATCHECK
UNIT_COUNT	x	x	x	x	x	x
UNIT_NAME	x	x	x	x	x	x
UNIT_TYPE		x	x	x	x	x
USERID	x	x	x	x	x	x
VOL_ADD			x			
VOL_ADD_ALGORITHM			x			
VOL_ADD_LIM			x			
VOL_ADD_MOUNT_USAGE			x			
VOL_ADD_PROMPT			x			
VOL_ADD_SGRP			x			
VOL_ADD_UNIT			x			
VOL_ADD_VGRP			x			
VOL_ADD_VGRP_SEARCH			x			
VOL_ADD_VSER			x			
VOLCT		x	x	x	x	x
VOLSER_ANY	x	x	x	x	x	x
VOLSER_CURRENT		x	x	x	x	x
VOLSER_FIRST		x	x	x	x	x
VSAM_SEPARATE			x	x		
WRITE_MSG	x	x	x	x	x	x

Notes:

1. JOBCLASS is set to x'D0', the "}" character, when ACS is called during initial batch JCL processing.
2. PROCESTEPNAME and STEPNAME are not available during ACS processing when called during initial batch JCL processing.
3. DDNAME is not available during IDCAMS DEFINE processing.

Appendix D. Messages

Advanced Allocation Management messages adhere to the following format.

Message format description

GLO*innnnx*

Where:

GLO Indicates that the message was issued by Advanced Allocation Management

i Indicates the message identifier

nnn Indicates the message identification number

x Indicates the severity of the message:

- **A** Information message. User action required.
- **E** Error message. Some errors might be user-correctable, read the User Response text to determine the appropriate course of action.
- **F** Fatal error message. Read the User Response text to determine the appropriate course of action.
- **I** Information message. No user action required.
- **W** Warning message. Results might not be as expected.

Each message also includes the following information:

Explanation:
The Explanation section explains what the message text means, why it occurred, and what its variables represent.

User response:
The User response section describes whether a response is necessary, what the appropriate response is, and how the response will affect the system or program.

Finding messages in IBM Knowledge Center

Use the following tips to help you find messages and codes in IBM Knowledge Center.

- For negative SQL codes, enclose the SQL code in quotation marks. For example, enter `"-105"` instead of `-105` in the search field. When the minus sign is not enclosed in quotation marks, only pages that do not include the SQL code are returned in the search results. For example, a search term of `-105` returns all pages that do not include instances of 105.
- For positive SQL codes, enter the code with or without a leading plus sign (+). For example, enter `110` or `+110`.
- For messages, search for the complete message number, including the prefix (DSN) and the suffix (A, D, E, or I). For example, DSNU100I is a complete message number.
- For reason codes, search for the complete reason code number including the leading zeros. For example, 00C10203 is a complete reason code.

For more information about finding content in IBM Knowledge Center, see Finding information by searching.

User interface messages

These messages are related to Advanced Allocation Management panel entries or ISPF issues.

GLO002E Enter a valid option or X to exit.

Explanation: An incorrect selection was specified.

User response: Choose one of the valid options described in Chapter 8, "Managing control data set members and subsystems," on page 275.

GLO003E The data set name is not valid.

Explanation: The data set name that is specified in the **Data set name** field on the Tivoli Advanced Allocation Management Control Member Management panel is not valid.

User response: Provide a valid data set name of an existing data set. The data set name must be enclosed in single quotation marks.

GLO004E The data set does not exist. Specify the correct data set or specify an SSID.

Explanation: The data set specified in the **Data set name** field on the Tivoli Advanced Allocation Management Control Member Management could not be located.

User response: Take one of the following actions:

- Specify the name of an existing data set in the **Data set name** field. The data set name must be enclosed in single quotation marks.
- Specify the four character subsystem ID of an active Advanced Allocation Management subsystem in the **Active Subsystem ID** field and press Enter to retrieve the selected subsystem control data set name.

GLO005E Enter a valid option. Valid options are: V, E, R, D, S, or blank.

Explanation: An incorrect option was specified on the Tivoli Advanced Allocation Management Control Member Management panel.

User response: Specify **V** (View), **E** (Edit), **R** (Rename), **D** (Delete), **S** (Syntax-check), or blank (display list). See "Managing control data set members" on page 275 for more information.

GLO006E The member name is not valid. Enter a valid member name.

Explanation: The member name that was specified on the Tivoli Advanced Allocation Management Control Member Management panel or on the Tivoli Advanced Allocation Management Subsystem Settings panel was not valid.

User response: Specify the name of a control data set member. If the message is displayed on the Tivoli Advanced Allocation Management Control Member Management panel, see "Managing control data set members" on page 275 for information about specifying member names.

If the message is displayed on the Tivoli Advanced Allocation Management Subsystem Settings panel, see "Viewing and updating subsystem options" on page 278 for information about specifying member names.

GLO007E ISPF error: *zerrlm*.

Explanation: The text of this message varies depending upon the type of ISPF error.

User response: Contact IBM Software Support.

GLO008E The member already exists in this data set.

Explanation: The specified member exists in the data set.

User response: Specify a unique member name.

GLO009I The operation completed successfully.

Explanation: The operation completed without error.

User response: No action is required.

GLO010E No active subsystems found.

Explanation: Advanced Allocation Management could not locate an active subsystem.

An Advanced Allocation Management subsystem must be active. An Advanced Allocation Management subsystem is considered to be active when the Advanced Allocation Management started task that is associated with the subsystem ID is running and the product-level intercepts are enabled. For information about the relationship between the Advanced Allocation Management subsystem and started task, see "Architecture and processing" on page 12.

User response: For information about starting the Advanced Allocation Management starting task, see

"Starting the started task" on page 33. For information about modifying product-level intercepts through the Advanced Allocation Management user interface, see Chapter 8, "Managing control data set members and subsystems," on page 275.

GLO011E Active subsystem *ssid* not found.

Explanation: Advanced Allocation Management could not locate the subsystem (*ssid*) that is specified in the **Active Subsystem ID** field on the Advanced Allocation Management panel.

User response: Specify the four character subsystem ID of an active Advanced Allocation Management subsystem.

GLO012E The log data set does not exist for the active subsystem.

Explanation: The specified log data set does not exist for the Advanced Allocation Management subsystem.

User response: Contact IBM Software Support.

GLO013E Type S to select an active subsystem.

Explanation: An incorrect option was specified to select an active subsystem.

User response: Type S to select an active subsystem.

GLO014E Enter a valid option. Valid options are: E or D.

Explanation: An incorrect option was specified in the **Subsystem Status** field on the Tivoli Advanced Allocation Management Subsystem Settings panel.

User response: Specify **E (ENABLE)** to enable the subsystem for processing or **D (DISABLE)** to disable the subsystem for processing so that no processing occurs.

GLO016E GLOQUERY failed. Return code: *rcreg15*.

Explanation: A query operation failed. The return code (*rcreg15*) is provided. The following return codes are valid:

0 The request was successful. The requested updates were made.

4 A request to activate a rule definitions or volume group definitions member failed. To determine the syntax errors and correct them, specify the syntax-check function (option **S**).

8 Sufficient storage is not available for the request. Log on with a larger region size and attempt the operation again.

12 Advanced Allocation Management was not started on the system. No updates are possible. Start the product and try the operation again. For information about starting the Advanced Allocation Management started task, see "Starting the started task" on page 33.

16 The user ID is not authorized to perform update requests. For information about providing users with the ability to view and update selected subsystem options, see "Securing the user interface update functions" on page 31.

20 No Advanced Allocation Management ID was specified. Update processing requires an active subsystem.

24 No subsystems found. Start the product and attempt the operation again.

28 One or more subsystem fields were specified for update, but the specified subsystem was not found. Start the product and try the operation again.

32 One or more subsystem fields were specified for update, but the specified subsystem was not active. Start the previously active subsystem again and try the operation again.

User response: If the return code is greater than 1000 contact IBM Software Support.

GLO017E GLOUPDTE failed. Return code: *rcreg15*.

Explanation: An update operation failed. The return code (*rcreg15*) is provided. The following return codes are valid:

0 The request was successful. The requested updates were made.

4 A request to activate a rule definitions or volume group definitions member failed. See the Advanced Allocation Management job log for a description of the errors.

8 Sufficient storage is not available for the update request. Log on with a larger region size.

12 Advanced Allocation Management was not started on the system. No updates are possible. Start the product and try the operation again. For information about starting the Advanced Allocation Management started task, see "Starting the started task" on page 33.

User response: If the return code is greater than 1000 contact IBM Software Support.

GLO018E GLOSCHEK failed. Return code: *rcreg15*.

Explanation: A check operation failed. The return code is *rcreg15*.

User response: Contact IBM Software Support.

GLO019E The syntax DD could not be allocated. Report processing has been stopped.

Explanation: An error occurred during allocation.

User response: Contact IBM Software Support.

GLO020E Truncation error displaying panel in *pnlname* **-** *rc*.

Explanation: A truncation error occurred when the specified panel (*pnlname*) was displayed. The return code is *rc*.

User response: Contact IBM Software Support.

GLO021E Severe error displaying panel in *pnlname* **-** *rc*.

Explanation: An error occurred when the specified panel (*pnlname*) was displayed. The return code is *rc*.

User response: Contact IBM Software Support.

GLO022E Unexpected return code from panel in *pnlname*.

Explanation: Advanced Allocation Management encountered an unexpected return code from the specified panel (*pnlname*).

User response: Contact IBM Software Support.

GLO023I Log data set *logdsn* **is empty.**

Explanation: The specified log data set (*logdsn*) is empty.

User response: No action is required.

GLO024E The member is not in Rule Definition format. Verify the format.

Explanation: An attempt to change or define the current rule definition member using Option 2 (**View/Update Options**) failed because the first line of the member did not begin with the comment `* RULE DEFINITIONS`

User response: Add a comment as the first line of the rule definition member. The comment must contain the text "RULE DEFINITIONS".

GLO025E The member is not in VGRP definition format. Verify the format.

Explanation: An attempt to change or define the current VGRP member using Option 2 (**View/Update Options**) failed because the first line of the member did not begin with the comment `* VGRP DEFINITIONS`

User response: Add a comment as the first line of the VGRP member. The comment must contain the text "VGRP DEFINITIONS".

GLO026E The member is not in Rule, VGRP, or VAR definition format. The syntax check was not completed.

Explanation: An incorrect member type was specified. The syntax-check option is only valid for Rule Definition, VGRP Definition, and VAR Definition member types.

User response: Specify one of the valid member types: Rule Definition, VGRP, or VAR Definition. For information about member types, see “Components” on page 10.

GLO027E The SMF Source File field must not be blank.

Explanation: The **SMF Source File** field must contain the name of an SMF source file.

User response: Specify the SMF source file to be searched for Advanced Allocation Management event records. The SMF data set you specify must be a non-VSAM data set.

GLO028E The Specify additional SMF source files field must be blank or /.

Explanation: An incorrect option was specified.

User response: To specify additional SMF source files, type a forward slash (/) to display the Tivoli Advanced Allocation Management SMF Source Files panel.

GLO029E The Specify additional filter set criteria field must be blank or /.

Explanation: An incorrect option was specified.

User response: To specify additional filter set criteria, type a forward slash (/) to display the Tivoli Advanced Allocation Management Filter Set Criteria panel.

GLO030E Valid options for Detail Report are: Y, N, B, or A.

Explanation: An incorrect option was specified.

User response: Specify one of the following options:

Y (Default.) Include detail records in the report.

N Do not include detail records in the report.

B (BASIC) Include detail records in the report. (Specifying this option is the same as specifying Y.)

A (ALL) Generate an expanded report with additional values.

GLO031E Valid options for Summary Report are: Y or N.

Explanation: An incorrect option was specified.

User response: To generate a summary of the records that were processed, specify `Y`. If you do not want to generate a summary, specify `N`.

GLO032E Valid options for Format are: R or L.

Explanation: An incorrect option was specified.

User response: Specify R (report output) or L (list output).

GLO033E Valid options for Output Mode are: V or B.

Explanation: An incorrect option was specified.

User response: Specify V (View) or B (browse).

GLO034E The CNTLIN DD could not be allocated. Report processing has been stopped.

Explanation: An error was encountered during the allocation of CNTLIN DD.

User response: CNTLIN specifies the location of the input control statements. It must have a record format of fixed (F) or fixed block (FB) and a record length (LRECL) of 80. Ensure that at least 10 cylinders are available on at least one user-accessible DASD volume. Look for related allocation failure messages that contain additional information about the nature of the error. Log off, and then log back on and then attempt the function again. If the problem persists, contact IBM Software Support.

GLO035E The CNTLOUT DD could not be allocated. Report processing has been stopped.

Explanation: An error was encountered during the allocation of CNTLOUT DD.

User response: CNTLOUT specifies where execution statistics and diagnostics are written. It must have a record format of fixed (F) or fixed block (FB) and a record length (LRECL) of 80. Ensure that at least 10 cylinders are available on at least one user-accessible DASD volume. Look for related allocation failure messages that contain additional information about the nature of the error. Log off, and then log back on and then attempt the function again. If the problem persists, contact IBM Software Support.

GLO036E The HISTOUT DD could not be allocated. Report processing has been stopped.

Explanation: An error was encountered during the allocation of the HISTOUT DD.

User response: HISTOUT specifies where the resulting report output is written. Ensure that at least 10 cylinders are available on at least one user-accessible DASD volume. Look for related allocation failure messages that contain more information about the nature of the error. Log off, and then log back on and then attempt the function again. For more information about the HISTOUT DD, see "Configuring batch reporting" on page 293. If the problem persists, contact IBM Software Support.

GLO037E Cannot open CNTLOUT DD.

Explanation: Advanced Allocation Management could not open CNTLOUT DD.

User response: Log off and log back on and then try the function again. If the problem persists, contact IBM Software Support.

GLO038E Cannot open HISTOUT DD.

Explanation: Advanced Allocation Management could not open HISTOUT DD.

User response: Log off and log back on and then try the function again. If the problem persists, contact IBM Software Support.

GLO039E The SMF Record ID must be numeric.

Explanation: A numeric SMF record ID is required.

User response: Specify a numeric 1 to 3 digit record identifier with a value between 128 and 255.

GLO040E The SMF Record ID field must not be blank.

Explanation: The SMF record ID is required.

User response: Specify a numeric 1 to 3 digit record identifier with a value between 128 and 255.

GLO041E GLOUPDATE RACF authorization failed. RC: *rc*, RSN: *rsn*.

Explanation: An update failed because the user lacked the appropriate RACF authorization. The return code *rc* and reason code *rsn* from the RACROUTE REQUEST=AUTH macro are provided. For more

information, see the *z/OS Security Server RACROUTE Macro Reference*.

User response: Ensure that the user has the appropriate RACF update authority to perform the action. For more information, see "Securing the user interface update functions" on page 31.

GLO042E The member is not in VAR definition format. Verify the format.

Explanation: An attempt to change or define the current variable definitions using Option 2 (**View/Update Options**) failed because the first line of the VARDEFS member did not begin with the comment `* VAR DEFINITIONS`

User response: Ensure that the VARDEFS member contains the `* VAR DEFINITIONS` comment as the first line in the member.

GLO043E Invalid WORK DS prefix. Please verify the format.

Explanation: An invalid work dsname prefix was entered.

User response: Specify the work dsname prefix again and ensure that the data set name node meets the following requirements:

1. The first character must be an alphabetic or national ($, #, @) character.
2. Characters 2 - 8 must be an alphanumeric or national character, or a hyphen.

GLO044E Invalid listing data set name. Please verify the format.

Explanation: An invalid listing data set name was entered.

User response: Specify a valid listing data set name.

Subsystem and started task messages

These messages are related to issues with the Advanced Allocation Management subsystem or started task.

GLO1000E Insufficient operating system level; OS/390 R2.10 (SP6.10) or higher required.

Explanation: Product initialization determined that the level of the operating system is not at a level that is high enough to support the product.

User response: Run the product on a system with OS/390 at the 2.10 level or above.

GLO1001E Product is not running APF-authorized.

Explanation: Product initialization determined that the libraries from which the product is being loaded are not APF-authorized.

User response: Run the product from APF-authorized libraries. For more information, see "APF-authorizing the load libraries" on page 27. For more information about APF authorization, see the *MVS Programming: Authorized Assembler Services Guide*.

GLO1001I RULEDEFS activation successful –*ssss*.

Explanation: This message is issued to the operator console following successful activation of rule definitions using the **`ACTIVATE RULEDEFS`** operator command.

User response: No action is required.

GLO1002E Insufficient virtual storage for product processing.

Explanation: Product initialization was unable to obtain the required above-the-line storage.

User response: Increase the amount of available above-the-line storage and restart the product. If increasing the amount of available above-the-line storage is not successful, contact IBM Software Support.

GLO1003E Insufficient virtual storage for product processing.

Explanation: Product initialization was unable to obtain the required below-the-line storage.

User response: Increase the amount of available below-the-line storage and restart the product. If increasing the amount of available below-the-line storage is not successful, contact IBM Software Support.

GLO1004E Unable to locate required ddname - CONTROL.

Explanation: During product initialization, the CONTROL DD statement could not be located in the product started task procedure.

User response: Add the CONTROL DD statement to the Advanced Allocation Management started task procedure as described in "Configuring the started task JCL" on page 28 and try again.

GLO1004I VGRPDEFS activation successful –*ssss*.

Explanation: This message is issued to the operator console following successful activation of volume group definitions using the **ACTIVATE VGRPDEFS** operator command.

User response: No action is required.

GLO1005E Error occurred during SWAREQ processing for JFCB for ddname control, rc=*rrrrrrrr*.

Explanation: An internal error (*rrrrrrrr*) occurred while processing the CONTROL DD statement during product initialization.

User response: Make sure that the CONTROL DD statement points to a valid partitioned data set and try again. For more information, see "Configuring the started task JCL" on page 28. If the error persists, contact IBM Software Support.

GLO1006E Unable to locate required ddname - OPTIONS.

Explanation: During product initialization, the OPTIONS DD statement could not be located in the product started task procedure.

User response: Add the OPTIONS DD statement to the Advanced Allocation Management started task procedure as described in "Configuring the started task JCL" on page 28 and try again.

GLO1007E Error occurred during SWAREQ processing for JFCB for ddname OPTIONS, rc=*rrrrrrrr*.

Explanation: During product initialization, an internal error (*rrrrrrrr*) occurred while processing the OPTIONS DD statement.

User response: Make sure that the OPTIONS DD statement points to a valid data set as described in "Configuring the started task JCL" on page 28 and try again. If the error persists, contact IBM Software Support.

GLO1007I RULEDEFS activated on *mm/dd/yyyy* at *hh:mm:ss* from member *member* -*ssss*.

Explanation: This message is issued in response to the **DISPLAY RULEDEFS** operator command. *mm/dd/yyyy* indicates the month, day and year on which the rule definitions were activated, *hh:mm:ss* is the time in hours, minute and seconds at which the rule definitions were activated, and *member* is the member name from the control data set from which the rule definitions were processed.

User response: No action is required.

GLO1008I RULEDEFS not activated –*ssss*.

Explanation: This message is issued in response to the **DISPLAY RULEDEFS** operator command when no rule definitions are activated.

User response: No action is required.

GLO1009E OPEN failed for processing options member; default options used.

Explanation: OPEN processing was unsuccessful for the OPTIONS member so the default options were used.

User response: Make sure that the OPTIONS DD statement points to a valid data set as described in "Configuring the started task JCL" on page 28 and try again. If the error continues, contact IBM Software Support.

GLO1009I VGRPDEFS activated on *mm/dd/yyyy* at *hh:mm:ss* from member *member* -*ssss*.

Explanation: This message is issued in response to the **DISPLAY VGRPDEFS** operator command. *mm/dd/yyyy* indicates the month, day and year on which the volume group definitions were activated and *hh:mm:ss* indicates the time in hours, minutes and seconds, at which the volume group definitions were activated, and *member* is the member name from the control data set from which the volume group definitions were processed.

User response: No action is required.

GLO1010I Product intercepts have been enabled.

Explanation: This message is issued during product initialization and indicates successful installation of the product intercepts.

User response: No action is required.

GLO1012E Attach for GLOMAIN failed, rc=*rrrrrrrr*.

Explanation: During product initialization, the startup of an internal task failed. *rrrrrrrr* identifies the internal error code.

User response: Examine the other error messages that were issued at the same time as this message. If no cause can be determined, contact IBM Software Support.

GLO1013I Product termination is complete.

Explanation: At completion of termination processing, this message is issued in response to the product shutdown command.

User response: No action is required.

GLO1021E Invalid option specified - *pppppppp*.

Explanation: During product initialization, an incorrect keyword was encountered when processing the subsystem options in the OPTIONS member. *pppppppp* is the incorrect option that was encountered or the value (NONE) if blank options were specified.

User response: Correct the specified OPTION keyword and restart the product. For information about specifying subsystem options, see "Specifying subsystem options" on page 29.

GLO1022E Required delimiter "=" is missing - *pppppppp*.

Explanation: During product initialization, while processing the subsystem options in the OPTIONS member, a keyword required an equal sign (=) but it was missing. *pppppppp* is the associated keyword.

User response: Correct the specified keyword and restart the product. For information about specifying subsystem options, see "Specifying subsystem options" on page 29.

GLO1023E Invalid value specified for option - *pppppppp*.

Explanation: During product initialization, while processing the subsystem options in the OPTIONS member, a keyword was encountered with an incorrect value. *pppppppp* indicates the keyword that has the incorrect value.

User response: Correct the keyword value and restart the product. For information about specifying subsystem options, see "Specifying subsystem options" on page 29.

GLO1024I Processing option set - SSID=*ssss*.

Explanation: This message is issued during product initialization. It displays the value (*ssss*) that is specified for the SSID keyword in the OPTIONS member.

User response: No action is required.

GLO1025E Invalid value specified for option - SSID=*ssss*.

Explanation: During product initialization, while processing the subsystem options in the OPTIONS member, an incorrect value (*ssss*) was encountered for the SSID option.

User response: Correct the SSID value and restart the product. For information about specifying subsystem options, see "Specifying subsystem options" on page 29.

GLO1026I Processing option set - INITIAL_RULEDEF=*rrrrrrrr*.

Explanation: This message is issued during product initialization. It displays the value (*rrrrrrrr*) that is specified for the INITIAL_RULEDEF keyword in the OPTIONS member.

User response: No action is required.

GLO1027E Invalid value specified for option -INITIAL_RULEDEF=*rrrrrrrr*

Explanation: During product initialization, while processing the subsystem options in the OPTIONS member, an incorrect value (*rrrrrrrr*) was encountered for the INITIAL_RULEDEF option.

User response: Correct the INITIAL_RULEDEF value and restart the product. For information about specifying subsystem options, see "Specifying subsystem options" on page 29.

GLO1028I Processing option set - INITIAL_VGRPDEF=*vvvvvvvv*.

Explanation: This message is issued during product initialization. It displays the value (*vvvvvvvv*) that is specified for the INITIAL_VGRPDEF keyword in the OPTIONS member.

User response: No action is required.

GLO1029E Invalid value specified for option - INITIAL_VGRPDEF=*vvvvvvvv*.

Explanation: During product initialization, while processing the subsystem options in the OPTIONS member, an incorrect value (*vvvvvvvv*) was encountered for the INITIAL_VGRPDEF option.

User response: Correct the INITIAL_VGRPDEF value and restart the product. For information about specifying subsystem options, see "Specifying subsystem options" on page 29.

GLO1034E Error in name/token retrieval processing, rc=*rrrrrrrr*.

Explanation: During product initialization, an internal system error (*rrrrrrrr*) was encountered in establishing the product.

User response: Contact IBM Software Support.

GLO1035E Name/token already exists, but token is zero.

Explanation: During product initialization, an internal system error was encountered in establishing the product.

User response: Contact IBM Software Support.

GLO1036E Name/token already exists, but token does not point to a valid product block.

Explanation: During product initialization, an internal system error was encountered in establishing the product. The most likely cause of the error is that after the current IPL, a version of Advanced Allocation Management that is incompatible with Advanced Allocation Management 3.2 was started before starting Advanced Allocation Management 3.2.

User response: IPL the system before starting Advanced Allocation Management 3.2. If the problem persists, contact IBM Software Support.

GLO1038E Unable to obtain storage for product control block, rc=*rrrrrrrr*.

Explanation: During product initialization, above-the-line CSA storage could not be obtained for a product control block as indicated by the internal return code *rrrrrrrr*.

User response: Investigate and correct the shortage of above-the-line CSA storage and restart the product. If the problem persists, contact IBM Software Support.

GLO1040E Error in name/token create processing, rc=*rrrrrrrr*.

Explanation: During product initialization, an internal system error (*rrrrrrrr*) was encountered in establishing the product.

User response: Contact IBM Software Support.

GLO1041I Product intercepts have been established.

Explanation: All intercepts were successfully established.

User response: No action is required.

GLO1042E Unable to obtain storage for common area routine, rc=*rrrrrrrr*.

Explanation: During product initialization, above-the-line CSA storage could not be obtained for loading a required product routine as indicated by the internal return code *rrrrrrrr*.

User response: Investigate and correct the shortage of above-the-line CSA storage and restart the product. If the problem persists, contact IBM Software Support.

GLO1043E BLDL failed for *mmmmmmmm*, rc=*rrrrrrrr*.

Explanation: During product initialization, a required load module could not be located. *mmmmmmmm* identifies the load module and *rrrrrrrr* is the internal return code.

User response: Verify that the load modules for the product are accessible in a STEPLIB in the product started task or in the system LINKLIST concatenation, and then restart the product.

GLO1044E Unable to determine origin of *mmmmmmmm*.

Explanation: During product initialization, an error was encountered while processing the product load module *mmmmmmmm*.

User response: Verify that the load modules for the product are accessible in a STEPLIB in the product started task or in the system LINKLIST concatenation, and then restart the product.

GLO1046E Private load failed for *mmmmmmmm*.

Explanation: During product initialization, the processing of a product load module (*mmmmmmm*) to be located in above-the-line private storage, failed.

User response: Verify that the load modules for the product are accessible in a STEPLIB in the product started task, or in the system LINKLIST concatenation, and then restart the product. Also check the available amount of above-the-line private storage that is available for the Advanced Allocation Management started task. After correcting the problem, restart the product. If the error cannot be determined, contact IBM Software Support.

GLO1047E Common load failed for *mmmmmmmm*.

Explanation: During product initialization, the processing of a product load module (*mmmmmmm*) to be located in above-the-line common storage, failed.

User response: Verify that the load modules for the product are accessible in a STEPLIB in the product started task or in the system LINKLIST concatenation, and then restart the product. Also check the available amount of above-the-line common storage that is available for the Advanced Allocation Management started task. After correcting the problem, restart the product. If the error cannot be determined, contact IBM Software Support.

GLO1048E Unable to obtain storage for A1 intercept, rc=*rrrrrrrr*.

Explanation: During product initialization an error (*rrrrrrrr*) was encountered while attempting to obtain above-the-line common storage for a product control block.

User response: Investigate a potential shortage of above-the-line common storage and restart the product. If the problem continues, contact IBM Software Support.

GLO1050E Unable to locate LPDE for IEFW21SD.

Explanation: During product initialization, Advanced Allocation Management could not locate a required pointer to an operating system module.

User response: Contact IBM Software Support.

GLO1052E Unable to locate IEFAB434.

Explanation: During product initialization, Advanced Allocation Management could not locate a required operating system module.

User response: Contact IBM Software Support.

GLO1055E Page service list exhausted for A1 intercept.

Explanation: During product initialization, Advanced Allocation Management encountered an internal error while it was attempting to establish a product intercept.

User response: Contact IBM Software Support.

GLO1056E Unable to obtain storage for E1 intercept, rc=*rrrrrrrr*.

Explanation: During product initialization, an error (*rrrrrrrr*) was encountered while attempting to obtain below-the-line common storage for a product control block.

User response: Investigate a potential shortage of below-the-line common storage and restart the product. If the problem continues, contact IBM Software Support.

GLO1058E Unable to locate LPDE for IGC0005E.

Explanation: During product initialization, Advanced Allocation Management could not locate a required pointer to an operating system module.

User response: Contact IBM Software Support.

GLO1060E Unable to locate IFG0554P.

Explanation: During product initialization, Advanced Allocation Management could not located a required operating system module.

User response: Contact IBM Software Support.

GLO1063E Page service list exhausted for E1 intercept.

Explanation: During product initialization, Advanced Allocation Management encountered an internal error while it was attempting to establish a product intercept.

User response: Contact IBM Software Support.

GLO1064E Unable to locate LPDE for IGC0002A.

Explanation: During product initialization, Advanced Allocation Management could not locate a required pointer to an operating system module.

User response: Contact IBM Software Support.

GLO1065W Entry point in IGC0002A'S LPDE does not match VCON references.

Explanation: During product initialization, Advanced Allocation Management encountered an internal error while it was attempting to establish a product intercept.

User response: Contact IBM Software Support.

GLO1066E Unable to obtain storage for S1 intercept, rc=*rrrrrrrr*.

Explanation: During product initialization, an error (*rrrrrrrr*) was encountered while attempting to obtain above-the-line common storage for a product control block.

User response: Investigate a potential shortage of above-the-line common storage and restart the product. If the problem persists, contact IBM Software Support.

GLO1068E Unable to obtain storage for A4 intercept, rc=*rrrrrrrr*.

Explanation: During product initialization, an error (*rrrrrrrr*) was encountered while attempting to obtain above-the-line common storage for a product control block.

User response: Investigate a potential shortage of above-the-line common storage and restart the product. If the problem persists, contact IBM Software Support.

GLO1070E Unable to locate IGG021AB.

Explanation: During product initialization, Advanced Allocation Management could not locate a required operating system module.

User response: Contact IBM Software Support.

GLO1072E Unable to locate IGC0002F.

Explanation: During product initialization, Advanced Allocation Management could not locate a required operating system module.

User response: Contact IBM Software Support.

GLO1073E Page service list exhausted for S1 intercept.

Explanation: During product initialization, Advanced Allocation Management encountered an internal error while it was attempting to establish a product intercept.

User response: Contact IBM Software Support.

GLO1073W Maximum active subsystems exceeded (4).

Explanation: The current iteration of Advanced Allocation Management that is being started exceeds the limit of four concurrently active subsystems on a single z/OS system. Startup for the current iteration stops.

User response: If the current iteration of Advanced Allocation Management is required, shut down one of the active Advanced Allocation Management subsystems and then restart the current iteration.

To display all active Advanced Allocation Management subsystems use the `display,subsystems,all` command or select option 2 (**View/Update Options**) from the from the Tivoli Advanced Allocation Management Primary Option Menu.

GLO1074E Duplicate subsystem found for SSID=*ssss*.

Explanation: During product initialization, Advanced Allocation Management encountered a duplicate product control block for the subsystem ID *ssss*.

User response: Contact IBM Software Support.

GLO1077E Unable to obtain storage for A2 intercept, rc=*rrrrrrrr*.

Explanation: During product initialization, an error (*rrrrrrrr*) occurred while attempting to obtain above-the-line common storage for a product control block.

User response: Investigate a potential shortage of above-the-line common storage and restart the product. If the problem continues, contact IBM Software Support.

GLO1079E Unable to locate LPDE for IEFW21SD.

Explanation: During product initialization, Advanced Allocation Management could not locate a required pointer to an operating system module.

User response: Contact IBM Software Support.

GLO1081E Unable to locate IEFAB4A0.

Explanation: During product initialization, Advanced Allocation Management could not locate a required operating system module.

User response: Contact IBM Software Support.

GLO1084E Page service list exhausted for A2 intercept.

Explanation: During product initialization, Advanced Allocation Management encountered an internal error while it was attempting to establish a product intercept.

User response: Contact IBM Software Support.

GLO1085E Unable to locate LPDE for IEFAB4A0.

Explanation: During product initialization, Advanced Allocation Management could not locate a required pointer to an operating system module.

User response: Contact IBM Software Support.

GLO1086E Entry point in IEFAB4A0"S LPDE does not match VCON references.

Explanation: During product initialization, Advanced Allocation Management encountered an internal error while it was attempting to establish a product intercept.

User response: Contact IBM Software Support.

GLO1087E Unable to obtain storage for A3 intercept, rc=*rrrrrrrr*.

Explanation: During product initialization, an error (*rrrrrrrr*) was encountered while attempting to obtain above-the-line common storage for a product control block.

User response: Investigate a potential shortage of above-the-line common storage and restart the product. If the problem continues, contact IBM Software Support.

GLO1089E Unable to locate LPDE for IGDZILLA.

Explanation: During product initialization, Advanced Allocation Management could not locate a required pointer to an operating system module.

User response: Contact IBM Software Support.

GLO1091E Unable to locate IGDSSIRT.

Explanation: During product initialization, Advanced Allocation Management could not locate a required operating system module.

User response: Contact IBM Software Support.

GLO1094E Page service list exhausted for A3 intercept.

Explanation: During product initialization, Advanced Allocation Management encountered an internal error while it was attempting to establish a product intercept.

User response: Contact IBM Software Support.

GLO1095E Unable to locate SSCT for SMS.

Explanation: The process that sets the Advanced Allocation Management intercepts could not locate the subsystem control table for SMS.

User response: Contact IBM Software Support.

GLO1096E Function is not valid for subsystem.

Explanation: The process that sets the Advanced Allocation Management intercepts found that the function code for SMS was not valid in the subsystem vector table.

User response: Contact IBM Software Support.

GLO1097E Unable to locate LPDE for IGDSSIRT.

Explanation: During product initialization, Advanced Allocation Management could not locate a required pointer to an operating system module.

User response: Contact IBM Software Support.

GLO1098E Entry point in IGDSSIRT'S LPDE does not match VCON references.

Explanation: During product initialization, Advanced Allocation Management encountered an internal error while it was attempting to establish a product intercept.

User response: Contact IBM Software Support.

GLO1100E Acronym check failed for GSSB.

Explanation: During product initialization, an internal error occurred within the product.

User response: Contact IBM Software Support.

GLO1101E Insufficient virtual storage for product processing.

Explanation: Main task startup was unable to obtain sufficient above-the-line private storage to initialize.

User response: Increase the amount of above-the-line private storage. If the problem persists, contact IBM Software Support.

GLO1102E Error occurred in cross-memory initialization.

Explanation: An internal error occurred during the main task start-up.

User response: Contact IBM Software Support.

GLO1103E Attach for GLOLSRP failed, rc=*rrrrrrrr* -*ssss*.

Explanation: An internal error (*rrrrrrrr*) was encountered within the main product task during the attempt to attach a service subtask for subsystem ID *ssss*.

User response: Contact IBM Software Support.

GLO1105E Attach for GLODSRP failed, rc=*rrrrrrrr* -*ssss*.

Explanation: An internal error (*rrrrrrrr*) was encountered within the main product task during the attempt to attach a service subtask for subsystem ID *ssss*.

User response: Contact IBM Software Support.

GLO1105I Subsystem is active and enabled.

Explanation: The main product task was successfully started and is now active.

User response: No action is required.

GLO1106I Subsystem initialization is complete.

Explanation: The main product task successfully completed initialization processing.

User response: No action is required.

GLO1107I Product termination has been requested.

Explanation: A command to stop the product was successfully received.

User response: No action is required.

GLO1108E Unable to locate *modname*.

Explanation: During product initialization, Advanced Allocation Management encountered an internal error while it was attempting to establish a product intercept for the module *modname*. The error occurred during the attempt to locate the module header.

User response: Contact IBM Software Support.

GLO1111E Unable to obtain storage for common area routine, rc=*rrrrrrrr*.

Explanation: Product subsystem initialization could not obtain enough storage to load a required module.

User response: Check and increase the amount available above- and below-the-line storage and restart the product. If the error persists, contact IBM Software Support.

GLO1112E BLDL failed for *mmmmmmmm*, rc=*rrrrrrrr*.

Explanation: During product subsystem initialization, Advanced Allocation Management could not locate a required load module. *mmmmmmmm* identifies the load module and *rrrrrrrr* is the internal return code.

User response: Verify that the load modules for the product are accessible in a STEPLIB in the product started task or in the system LINKLIST concatenation, and then restart the product.

GLO1113E Unable to determine origin of *mmmmmmmm*.

Explanation: An error was encountered during product subsystem initialization during the processing of the product load module *mmmmmmmm*.

User response: Verify that the load modules for the product are accessible in a STEPLIB in the product started task or in the system LINKLIST concatenation, and then restart the product.

GLO1115E Initial load failed for *mmmmmmmm*.

Explanation: During product subsystem initialization, a required load module (*mmmmmmmm*) did not load successfully.

User response: Verify that the load modules for the product are accessible in a STEPLIB in the product started task or in the system LINKLIST concatenation, and then restart the product. Check the available amount of above-the-line private storage that is available for the product started task. After correcting the problem restart the product. If you cannot determine the cause of the error, contact IBM Software Support.

GLO1116E Directed load failed for *mmmmmmmm*.

Explanation: During product subsystem initialization, a required load module (*mmmmmmmm*) did not load successfully.

User response: Verify that the load modules for the product are accessible in a STEPLIB in the product started task or in the system LINKLIST concatenation, and then restart the product.

GLO1117E Non-zero return code from SYSEVENT, rc=*rrrrrrrr* -*ssss*.

Explanation: During product subsystem initialization, an error (*rrrrrrrr*) was encountered when attempting to make the product started task address space non-swappable for subsystem *ssss*.

User response: Contact IBM Software Support.

GLO1118E Attach for GLOVIRP failed, rc=*rrrrrrrr* -*ssss*.

Explanation: An internal error (*rrrrrrrr*) occurred within the main product task during the attempt to attach a service subtask for subsystem ID *ssss*.

User response: Contact IBM Software Support.

GLO1122E Invalid command specified - *cccccccc* -*ssss*.

Explanation: The product subsystem command parser received an error during the processing of the command (*cccccccc*) that was issued to the started task for subsystem ID *ssss*.

User response: Correct the command and issue the command again. See Appendix A, "Operator commands," on page 319 for information about Advanced Allocation Management commands.

GLO1123E Invalid command specified - *cccccccc* -*ssss*.

Explanation: An incorrect or null product subsystem command (*cccccccc*) was issued to the started task for subsystem ID *ssss*.

User response: Correct the command and issue the command again. See Appendix A, "Operator commands," on page 319 for information about Advanced Allocation Management commands.

GLO1123W Active subsystem(s) detected; product-level module(s) not re-initialized.

Explanation: Advanced Allocation Management issues this message when it initializes and determines that there are other active Advanced Allocation Management modules available for reuse. (These modules would typically be loaded into storage.)

User response: No action is required.

GLO1124E Excessive operands specified for command - *cccccccc* -*ssss*.

Explanation: More operands than are allowed were specified for the **DISPLAY** command (*cccccccc*) that was issued to the product started task for subsystem ID *ssss*.

User response: Issue the command again using the correct number of operands. For information about the **DISPLAY** command, see "Display commands" on page 319.

GLO1125E Insufficient operands specified for command - *cccccccc* **-***ssss***.**

Explanation: The command that was issued contains fewer operands than the minimum required. The command that was entered is *cccccccc*. The subsystem ID is *ssss*.

User response: Issue the command again using the correct number of operands. See Appendix A, "Operator commands," on page 319 for information about Advanced Allocation Management commands.

GLO1126E Invalid operand specified for command - *cccccccc* -*ssss*.

Explanation: The command that was issued contains an incorrect operand. The command that was entered is *cccccccc*. The subsystem ID is *ssss*.

User response: Correct the incorrect operand and issue the command again. See Appendix A, "Operator commands," on page 319 for information about Advanced Allocation Management commands.

GLO1127I Subsystem is active | inactive and enabled | disabled -*ssss***.**

Explanation: This message is issued in response to the **DISPLAY SUBSYSTEM** or **DISPLAY ALL** operator command and shows the ACTIVE or INACTIVE status of the product subsystem and whether the subsystem is enabled or disabled for subsystem *ssss*.

User response: No action is required.

GLO1129I There are currently no subsystems -*ssss***.**

Explanation: This message is issued in response to the product operator command **DISPLAY SUBSYSTEM ALL** when no subsystems are located.

User response: No action is required.

GLO1130I Subsystem *xxxx* is active | inactive and enabled | disabled -*ssss***.**

Explanation: This message is issued in response to the **DISPLAY SUBSYSTEM ALL** operator command issued to subsystem *ssss* and shows the active or inactive status of each product subsystem as identified by *xxxx* and whether the subsystem is enabled or disabled.

User response: No action is required.

GLO1131I RULEDEFS activated on *mm/dd/yyyy* **at** *hh:mm:ss* **from member** *mmmmmmmm* **-***ssss***.**

Explanation: This message is issued in response to the **DISPLAY RULEDEFS** operator command to subsystem ID *ssss*. It shows the date *mm/dd/yyyy* and time *hh:mm:ss* at which the active set of rule definitions was last activated and the member name (*mmmmmmmm*) from which they were activated.

User response: No action is required.

GLO1132I RULEDEFS not activated -*ssss***.**

Explanation: This message is issued in response to the **DISPLAY RULEDEFS** operator command to subsystem ID *ssss* when no activated rule definitions were found.

User response: No action is required.

GLO1133I VGRPDEFS activated on *mm/dd/yyyy* **at** *hh:mm:ss* **FROM MEMBER** *mmmmmmmm* **-***ssss***.**

Explanation: This message is issued in response to the **DISPLAY VGRPDEFS** operator command to subsystem ID *ssss*. It shows the date *mm/dd/yyyy* and time *hh:mm:ss* at which the active set of volume group definitions was last activated and the member name (*mmmmmmmm*) from which they were activated.

User response: No action is required.

GLO1134I VGRPDEFS not activated -*ssss***.**

Explanation: This message is issued in response to the **DISPLAY VGRPDEFS** operator command to subsystem ID *ssss* when no activated volume group definitions were found.

User response: No action is required.

GLO1135I Product intercepts are enabled | disabled -*ssss***.**

Explanation: This message is issued in response to the **DISPLAY INTERCEPTS** operator command to subsystem ID *ssss* and shows whether the intercepts are enabled or disabled.

User response: No action is required.

GLO1136I Product-level tracing is enabled | disabled -*ssss***.**

Explanation: This message is issued in response to the **DISPLAY TRACING** operator command to subsystem ID *ssss* and shows whether the product tracing facility is enabled or disabled.

User response: No action is required.

GLO1137I Subsystem-level tracing is enabled | disabled -*ssss***.**

Explanation: This message is issued in response to the **DISPLAY TRACING** operator command to subsystem ID *ssss* and shows whether the subsystem tracing facility is enabled or disabled.

User response: No action is required.

GLO1138E Excessive operands specified for command - *cccccccc* **-***ssss***.**

Explanation: More operands than are allowed were specified for the **ENABLE** command (*cccccccc*) issued to the Advanced Allocation Management started task for subsystem ID *ssss*.

User response: Issue the command again using the correct number of operands. See "Enable and disable commands" on page 320 for information about the **ENABLE** command.

GLO1139I VARDEFS defined on mm/dd/yyyy at hh:mm:ss from member *member* **–***ssid***.**

Explanation: This message provides the current VARDEFS member information.

User response: No action is required.

GLO1140I VARDEFS not defined –*ssid***.**

Explanation: This message is generated from a **DISPLAY VARDEFS** or **DISPLAY ALL** command when no VARDEFS member is defined.

User response: No action is required.

GLO1141I Subsystem is now enabled -*ssss***.**

Explanation: This message is issued in response to the **ENABLE SUBSYSTEM** operator command and indicates that the subsystem *ssss* was successfully enabled.

User response: No action is required.

GLO1142I Product intercepts are now enabled -*ssss***.**

Explanation: This message is issued in response to the **ENABLE INTERCEPTS** operator command and indicates that the intercepts for subsystem *ssss* were successfully enabled.

User response: No action is required.

GLO1144I Tracing for product is now enabled -*ssss***.**

Explanation: This message is issued in response to the **ENABLE TRACING** or **ENABLE TRACING ALL** operator command for subsystem ID *ssss* and indicates that product level tracing is now enabled.

User response: No action is required.

GLO1145I Tracing for subsystem is now enabled -*ssss***.**

Explanation: This message is issued in response to the **ENABLE TRACING ALL** operator command for subsystem ID *ssss* and indicates that subsystem level tracing is now enabled.

User response: No action is required.

GLO1146E Excessive operands specified for command - *cccccccc* **-***ssss***.**

Explanation: More operands than are allowed were specified for the **DISABLE** command (*cccccccc*) that was issued to the product started task for subsystem ID *ssss*.

User response: Issue the command again using the correct number of operands. For more information, see "Enable and disable commands" on page 320.

GLO1149I Subsystem is now disabled -*ssss***.**

Explanation: This message is issued in response to the **DISABLE SUBSYSTEM** operator command and indicates that the subsystem *ssss* was successfully disabled.

User response: No action is required.

GLO1150I Product intercepts are now disabled -*ssss***.**

Explanation: This message is issued in response to the **DISABLE INTERCEPTS** operator command and indicates that the intercepts for subsystem *ssss* were successfully disabled.

User response: No action is required.

GLO1152I Tracing for product is now disabled -*ssss***.**

Explanation: This message is issued in response to the **DISABLE TRACING** or **DISABLE TRACING ALL** operator command for subsystem ID *ssss* and indicates that product level tracing is now disabled.

User response: No action is required.

GLO1153I Tracing for subsystem is now disabled -*ssss***.**

Explanation: This message is issued in response to the **DISABLE TRACING ALL** operator command for subsystem ID *ssss* and indicates that subsystem level tracing is now disabled.

User response: No action is required.

GLO1154E Excessive operands specified for command - *cccccccc* **-***ssss***.**

Explanation: More operands than are allowed were specified for the **ACTIVATE** or **SETVARDEFS** command (*cccccccc*) that was issued to the Advanced Allocation Management started task for subsystem ID *ssss*.

User response: Issue the command again using the correct number of operands. For more information, see Appendix A, "Operator commands," on page 319.

GLO1163E **VARDEFS member** *member* **errors encountered - see JESYSMSG for details –***ssid***.**

Explanation: During the activation of the rule definitions or volume group definitions, Advanced Allocation Management detected errors in the VARDEFS member. The errors are detailed in JESSYSLOG. Activation of the rule definitions or volume group definitions fails.

User response: Review JESSYSLOG and correct the errors in the VARDEFS member. If you are unable to resolve the errors, contact IBM Software Support.

GLO1164E *dddddddd* **member** *mmmmmmmm* **not activated due to VARDEF errors –***ssid***.**

Explanation: A RULEDEFS or VGRPDEFS activation attempt was unsuccessful due to variable definition errors.

User response: Review JESSYSLOG and correct the errors in the VARDEFS member. If you are unable to resolve the errors, contact IBM Software Support.

GLO1170I **VARDEFS member JRHVAR32 definition successful –***ssid***.**

Explanation: The VARDEFS member was successfully defined.

User response: No action is required.

GLO1175I **RULEDEFS | VGRPDEFS member** *mmmmmmmm* **activation successful -***ssss***.**

Explanation: This message is issued in response to the **ACTIVATE RULEDEFS** operator command or the initial rule definitions activation (as indicated from the OPTIONS member for subsystem ID *ssss*) to show that the RULEDEFS or VGRPDEFS from member *mmmmmmmm* was successfully activated.

User response: No action is required.

GLO1176E *dddddddd* **member** *mmmmmmmm* **activation failed - see JESYSMSG for details -***ssss***.**

Explanation: This message is issued in response to the **ACTIVATE RULEDEFS** operator command or the initial rule definitions activation (as indicated from the OPTIONS member for subsystem ID *ssss*) to show that the RULEDEFS or VGRPDEFS from member *mmmmmmmm* was not successful due to syntax errors.

User response: Review the error messages in the JES SYSMSG output for the Advanced Allocation Management started task, and then correct the errors and activate the RULEDEFS or VGRPDEFS again.

GLO1177E *dddddddd* **member** *mmmmmmmm* **activation failed - failure code** *cccc* **-***ssss***.**

Explanation: This message is issued in response to the **ACTIVATE RULEDEFS** operator command, or the initial rule definitions activation (as indicated from the OPTIONS member for subsystem ID *ssss)*. It shows that the RULEDEFS or VGRPDEFS activation from member *mmmmmmmm* was not successful due to an internal error (*cccc*). *dddddddd* indicates the member type (RULDEFS or VGRPDEFS).

User response: To determine the cause of the error, review the error messages in the JES SYSMSG output or the console log for the Advanced Allocation Management started task. Correct the errors and activate the RULEDEFS or VGRPDEFS again. If the problem persists, contact IBM Software Support.

GLO1179I **Original VCON located at** *location***, via instruction sequence matching-** *ssss***.**

Explanation: During product initialization, Advanced Allocation Management determined that a pointer to an operating system module was altered. This situation usually occurs when another software product sets a hook for the module. Advanced Allocation Management continues to attempt to locate the operating system module by matching instruction sequences in the calling module.

User response: No action is required.

GLO1180E **Unable to locate IGC0002F VCON via instruction sequence matching-** *ssss***.**

Explanation: During product initialization, Advanced Allocation Management determined that a pointer to an operating system module was altered. This situation usually occurs when another software product sets a hook for the module. Advanced Allocation Management attempted to locate the operating system module by matching instruction sequences in the calling module, but was unsuccessful.

User response: Contact IBM Software Support.

GLO1181I **Processing option set - S1_intercept=ENABLE|DISABLE.**

Explanation: This message is issued during product initialization to display the value that was set for the S1_INTERCEPT parameter in the OPTIONS member.

User response: No action is required.

GLO1182E **Invalid value specified for option - S1_INTERCEPT=***xxxxxxx***.**

Explanation: During product initialization, while processing the subsystem options in the OPTIONS member, an incorrect value (*xxxxxxx*) was encountered for the S1_INTERCEPT parameter.

User response: The valid values for S1_INTERCEPT are ENABLE and DISABLE. Correct the value and restart the product.

GLO1183W S1 intercept initialization has been disabled. Dynamic PDS directory increase function is not available.

Explanation: The S1_INTERCEPT parameter in the OPTIONS member was set to DISABLE. The dynamic PDS directory increase function, which requires the S1 intercept, is not available.

User response: To re-enable the function, remove the S1_INTERCEPT parameter from the OPTIONS member or change it to ENABLE and then IPL the system and restart Advanced Allocation Management.

GLO1184E Command verb not unique - *cccccccc* **-***ssss***.**

Explanation: More than one command exists that matches the abbreviation specified (*cccccccc*) for the command verb. The Advanced Allocation Management subsystem processing the command was *ssss*.

User response: Issue the command again, using a command verb abbreviation that more uniquely specifies the intended command. See Appendix A, "Operator commands," on page 319 for information about Advanced Allocation Management commands.

GLO1185E Invalid command syntax specified - *ssss***.**

Explanation: The command that was entered contains incorrect syntax. The Advanced Allocation Management subsystem processing the command was *ssss*.

User response: Review the command that was entered and correct the syntax. For more information, see Appendix A, "Operator commands," on page 319 and Advanced Allocation Management.

GLO1189E 2.1 name/token exists, but token does not point to a valid product block.

Explanation: Advanced Allocation Management version 3.1 (or later) detected that a previous version is present on the system, but the required product blocks are not valid. In this situation, version 3.1 (or later) initialization is not possible.

User response: To start version 3.1 (or later), you must first perform an IPL.

GLO1190E Previous version GPB located, but version is not 2.1.

Explanation: Advanced Allocation Management version 3.1 (or later) detected that a previous version is present on the system, but the version is not 2.1 or later. Version 3.1 (or later) cannot be initialized once a Advanced Allocation Management version that is earlier than 2.1 is started on the system.

User response: To start version 3.1 (or later), you must first perform an IPL to remove the older version.

GLO1191E Invalid module name specified - *cccccccc***.**

Explanation: The command that was entered (*cccccccc*) specifies an incorrect module name.

User response: Issue the command again with a correct module name.

GLO1192I Module *mmmmmmmm vvvv ffffffff dddddddd ttttt***.**

Explanation: This message displays module header information. *mmmmmmmm* is the name of the module, *vvvv* is the version, *ffffffff* is the FMID, *dddddddd* is the assembly date, and *ttttt* is the assembly time.

User response: No action is required.

GLO1193I Module *mmmmmmmm* **located at** *aaaaaaaa* **(***stgloc***).**

Explanation: This message displays the module address with the offset, if specified. *mmmmmmmm* is the name of the module, *aaaaaaaa* is the virtual storage address, and *stgloc* is the storage location (PRIVATE or COMMON).

User response: No action is required.

GLO1194I Unable to determine O/S level; Processing continues.

Explanation: During product startup, Advanced Allocation Management could not determine the operating system level. Processing continues.

User response: No action is required.

GLO1195E Error occurred during FREEMAIN for GPB, rc=*rrrrrrrr***.**

Explanation: During initialization, Advanced Allocation Management encountered an error and determined that it must stop. As part of this process, an attempt was made to freemain the product control block, but the FREEMAIN service encountered an error. *rrrrrrrr* contains the value that was returned in register 15. The process of stopping the product continues.

User response: Contact IBM Software Support.

GLO1196E Unexpected VCON count for *ii* **intercept; expected=***eee***, found=***fff***.**

Explanation: While setting product intercept *ii*, Advanced Allocation Management encountered an unexpected VCON count for a particular CSECT. The

expected VCON count is *eee* and the VCON count is *fff*. This situation can indicate a potential problem. An SVC dump is created. Depending upon the particular intercept, product initialization might continue or stop.

User response: Contact IBM Software Support.

GLO1197E Error in name/token delete processing, rc=*rrrrrrrr*.

Explanation: During initialization, Advanced Allocation Management encountered an error and determined it must stop processing. As part of the process, an attempt was made to delete the NAME/TOKEN of the product, but the NAME/TOKEN DELETE service encountered an error. *rrrrrrrr* contains the value that was returned in register 15. The process of stopping the product continues.

User response: Contact IBM Software Support.

GLO1200E Unable to obtain virtual storage for workarea.

Explanation: A service task of the main product started task could not obtain the required amount of above-the-line storage.

User response: Increase the amount of above-the-line storage for the Advanced Allocation Management task. If the problem persists, contact IBM Software Support.

GLO1202E Unable to obtain virtual storage for workarea.

Explanation: A service task of the main product started task could not obtain the required amount of above-the-line storage.

User response: Increase the amount of above-the-line storage for the Advanced Allocation Management task. If the problem persists, contact IBM Software Support.

GLO1205I O/S mode: *mmmmm*.

Explanation: This message displays the current operating system (*mmmmm*).

User response: No action is required.

GLO1206I O/S level: *nnnnnnn fffffff* (*vvvvvvv*).

Explanation: This message displays operating system level information. *nnnnnnn* is the name of the control program, *fffffff* is the FMID of the control program, and *vvvvvvv* is the control program version and release.

User response: No action is required.

GLO1207I Job entry: *jjjj*.

Explanation: This message displays the name of the primary job entry subsystem (*jjjj*).

User response: No action is required.

GLO1208I SAF product: *ssss*.

Explanation: This message displays the name of the active security product (*ssss*).

User response: No action is required.

GLO1300E Unable to obtain virtual storage for workarea.

Explanation: A service task of the main product started task could not obtain the required amount of above-the-line storage.

User response: Increase the amount of above-the-line storage for the Advanced Allocation Management task. If the problem persists, contact IBM Software Support.

GLO1355E Page service list exhausted for A5 intercept.

Explanation: Advanced Allocation Management encountered an internal error while attempting to establish a product intercept during product initialization.

User response: Contact IBM Software Support.

GLO1360I Processing option set - VARDEFS_MEMBER=*member name*.

Explanation: This message is issued during product initialization to display the member name (*member name*) that was specified for the VARDEFS_MEMBER keyword in the OPTIONS member.

User response: No action is required.

GLO1361E Invalid value specified for option - VARDEFS_MEMBER=*vvvv*.

Explanation: During product initialization, Advanced Allocation Management encountered an incorrect value *vvvv* for the VARDEFS_MEMBER option while processing the subsystem options in the OPTIONS member.

User response: Correct the VARDEFS_MEMBER value and restart the product. For information about subsystem options, see “Specifying subsystem options” on page 29.

GLO1366E Unable to obtain storage for Z1 intercept, rc=*rrrrrrrr*.

Explanation: During product initialization, an error (*rrrrrrrr*) was encountered while attempting to obtain above-the-line common storage for a product control block.

User response: Investigate a potential shortage of above-the-line common storage and restart the product. If the problem persists, contact IBM Software Support.

GLO1370E Unable to locate LPDE for IDA0192A.

Explanation: During product initialization, Advanced Allocation Management could not locate a required pointer to an operating system module.

User response: Contact IBM Software Support.

GLO1372E Unable to locate IDAOCTBL.

Explanation: During product initialization, Advanced Allocation Management could not locate a required operating system module.

User response: Contact IBM Software Support.

GLO1373E Unable to locate IDAO557B.

Explanation: During product initialization, Advanced Allocation Management could not locate a required operating system module.

User response: Contact IBM Software Support.

GLO1376I Processing option set – DSN_PATTERN_MATCHING=*oooooooo*.

Explanation: This message is issued during product initialization to display the value (BASIC, EXTENDED1, or EXTENDED2) that is specified for the DSN_PATTERN_MATCHING keyword in the OPTIONS member.

User response: No action is required.

GLO1378E Invalid value specified for option – DSN_PATTERN_MATCHING=*oooooooo*.

Explanation: During product initialization, Advanced Allocation Management encountered an incorrect value (*oooooooo*) for the DSN_PATTERN_MATCHING option while processing the subsystem options in the OPTIONS member.

User response: Correct the DSN_PATTERN_MATCHING value and restart the product. For more information, see "Specifying subsystem options" on page 29.

GLO1380E Unable to obtain storage for a6 intercept, rc=*rrrrrrrr*.

Explanation: During product initialization an error (*rrrrrrrr*) was encountered while attempting to obtain above-the-line common storage for a product control block.

User response: Investigate a potential shortage of above-the-line common storage and restart the product. If the problem persists, contact IBM Software Support.

GLO1382E Unable to locate ISTRING.

Explanation: During product initialization, Advanced Allocation Management could not locate an internal control block.

User response: Contact IBM Software Support.

GLO1383E Unable to locate LPDE for IDA0192A.

Explanation: During product initialization, Advanced Allocation Management could not locate a required pointer to an operating system module.

User response: Contact IBM Software Support.

GLO1384E Unable to locate IDAOCTBL.

Explanation: During product initialization, Advanced Allocation Management could not locate a required operating system module.

User response: Contact IBM Software Support.

GLO1385E Unable to locate IDAO557F.

Explanation: During product initialization, Advanced Allocation Management could not locate a required operating system module.

User response: Contact IBM Software Support.

GLO1388E Page service list exhausted for A6 intercept.

Explanation: During product initialization, while attempting to establish a product intercept, Advanced Allocation Management encountered an internal error.

User response: Contact IBM Software Support.

GLO1391E Internal error in GIRWB000, rc=*rrrrrrrr*.

Explanation: An internal error occurred in routine GIRWB000. The return code is *rrrrrrrr*.

User response: Contact IBM Software Support.

GLO1393E Unable to determine ILEVEL.

Explanation: During product initialization, Advanced Allocation Management could not determine an internal control block level.

User response: Contact IBM Software Support.

GLO1396W Z1 intercept initialization has been disabled.

Explanation: To enable the DATASET_INIT function, the Z1_INTERCEPT option must be set to ENABLE and then the system must be IPLed and Advanced Allocation Management must be restarted. For information about the Z1_INTERCEPT option, see "Specifying subsystem options" on page 29.

User response: Contact IBM Software Support.

GLO1397W Data set initialization function is not available.

Explanation: This message accompanies GLO1396W. See the explanation for GLO1396W.

User response: See the user response for GLO1396W.

GLO1398I Processing option set - Z1_INTERCEPT=*oooooooo*.

Explanation: This message is issued during product initialization to display the value (*oooooooo*) that is specified for the Z1_INTERCEPT keyword in the OPTIONS member.

User response: No action is required.

GLO1399E Invalid value specified for option - Z1_INTERCEPT=*oooooooo*.

Explanation: During product initialization, while processing the subsystem options in the OPTIONS member, an incorrect value (*oooooooo*) was encountered for the Z1_INTERCEPT option.

User response: Correct the Z1_INTERCEPT value and restart the product. For information about the Z1_INTERCEPT option, see "Specifying subsystem options" on page 29.

GLO1400E Unable to obtain storage for E2 intercept, rc=*rrrrrrrr*.

Explanation: During product initialization, an error (*rrrrrrrr*) was encountered while attempting to obtain below-the-line common storage for a product control block.

User response: Investigate a potential shortage of below-the-line common storage and restart the product. If the problem persists, contact IBM Software Support.

GLO1402E Unable to locate LPDE for IGC00020.

Explanation: During product initialization, Advanced Allocation Management could not locate a required pointer to an operating system module.

User response: Contact IBM Software Support.

GLO1403E Unable to locate IFG0202J.

Explanation: During product initialization, Advanced Allocation Management could not locate a required operating system module.

User response: Contact IBM Software Support.

GLO1406E Page service list exhausted for E2 intercept.

Explanation: During product initialization, Advanced Allocation Management encountered an internal error when it attempted to establish a product intercept.

User response: Contact IBM Software Support.

GLO1407E Unable to obtain storage for E3 intercept, rc=*rrrrrrrr*.

Explanation: During product initialization, an error (*rrrrrrrr*) was encountered while attempting to obtain below-the-line common storage for a product control block.

User response: Investigate a potential shortage of below-the-line common storage and restart the product. If the problem persists, contact IBM Software Support.

GLO1408E Unable to obtain storage for E4 intercept, rc=*rrrrrrrr*.

Explanation: During product initialization, an error (*rrrrrrrr*) was encountered while attempting to obtain below-the-line common storage for a product control block.

User response: Investigate a potential shortage of below-the-line common storage and restart the product. If the problem persists, contact IBM Software Support.

GLO1411E Unable to locate LPDE for IGC00020.

Explanation: During product initialization, Advanced Allocation Management could not locate a required pointer to an operating system module.

User response: Contact IBM Software Support.

GLO1412E Unable to locate LPDE for IGC00020.

Explanation: During product initialization, Advanced Allocation Management could not locate a required pointer to an operating system module.

User response: Contact IBM Software Support.

GLO1415E Unable to locate IFG0202J.

Explanation: During product initialization, Advanced Allocation Management could not locate a required operating system module.

User response: Contact IBM Software Support.

GLO1416E Unable to locate IFG0202I.

Explanation: During product initialization, Advanced Allocation Management could not locate a required operating system module.

User response: Contact IBM Software Support.

GLO1421E Page service list exhausted for E3 intercept.

Explanation: During product initialization, Advanced Allocation Management attempted to establish a product intercept and encountered an internal error.

User response: Contact IBM Software Support.

GLO1422E Page service list exhausted for E4 intercept.

Explanation: During product initialization, Advanced Allocation Management attempted to establish a product intercept and encountered an internal error.

User response: Contact IBM Software Support.

GLO1423E Unable to locate IFG055ZZ call.

Explanation: An internal error was encountered during Advanced Allocation Management initialization. Initialization does not continue.

User response: Contact IBM Software Support.

GLO1424E Unable to locate IFG055ZZ return.

Explanation: An internal error was encountered during Advanced Allocation Management initialization. Initialization does not continue.

User response: Contact IBM Software Support.

GLO1425E Unable to locate IFG055ZZ compare.

Explanation: An internal error was encountered during Advanced Allocation Management initialization. Initialization does not continue.

User response: Contact IBM Software Support.

GLO1426E Unable to locate IFG055ZZ branch.

Explanation: An internal error was encountered during Advanced Allocation Management initialization. Initialization does not continue.

User response: Contact IBM Software Support.

GLO1427E Unable to locate IFG055ZZ string.

Explanation: An internal error was encountered during Advanced Allocation Management initialization. Initialization does not continue.

User response: Contact IBM Software Support.

GLO1428E Invalid IFG055ZZ branch offset.

Explanation: An internal error was encountered during Advanced Allocation Management initialization. Initialization does not continue.

User response: Contact IBM Software Support.

GLO1450E Unexpected return code from enqueue test function, r15=*rc*.

Explanation: During the dynamic PDS directory increase function, an unexpected error code was returned when testing the ownership of the data set resource. Recovery cannot continue.

User response: Note the return code *rc* and contact IBM Software Support with the diagnostic information.

GLO1451W Unable to obtain storage for GQSCAN workarea.

Explanation: During the dynamic PDS directory increase function, Advanced Allocation Management was unable to obtain necessary storage for an MVS service. Recovery cannot continue.

User response: Increase the region size and run the process again. If the error still occurs, contact IBM Software Support.

GLO1452W Directory increase requires exclusive control, but the data set is being shared.

Explanation: During the dynamic PDS directory increase function, Advanced Allocation Management required exclusive control of the data set to maintain the integrity of the PDS directory, but was unable to obtain it. Recovery cannot continue.

User response: If you still want the directory increase, stop any processes that have shared ownership of the data set and run the process again.

GLO1453W Directory increase requires exclusive control, but elevation failed; r15=*rc*.

Explanation: During the dynamic PDS directory increase function, Advanced Allocation Management had shared ownership of the data set, but required exclusive control to maintain integrity of the PDS directory. An attempt to elevate the ownership to exclusive was unsuccessful. Recovery cannot continue.

User response: If you still want the directory increase, stop any processes that have ownership of the data set and run the process again.

GLO1454W Directory increase requires exclusive control, but data set is shared by one or more other tasks (1).

Explanation: During the dynamic PDS directory increase function, Advanced Allocation Management required exclusive control of the data set to maintain the integrity of the PDS directory, but was unable to obtain it. Recovery cannot continue.

User response: If you still want the directory increase, stop any processes that have shared ownership of the data set and run the process again.

GLO1455W Directory increase requires exclusive control, but was unable to determine the owning TCB.

Explanation: During the dynamic PDS directory increase function, Advanced Allocation Management required exclusive control of the data set to maintain the integrity of the PDS directory, but was unable to determine data set ownership. Recovery cannot continue.

User response: Contact IBM Software Support.

GLO1456W Directory increase requires exclusive control, but data set is shared by one or more other tasks (2).

Explanation: During the dynamic PDS directory increase function, Advanced Allocation Management required exclusive control of the data set to maintain the integrity of the PDS directory, but was unable to obtain it. Recovery cannot continue.

User response: If you still want the directory increase, stop any processes that have shared ownership of the data set and run the process again.

GLO1457E Non-zero return from GQSCAN service, r15=*rc*.

Explanation: During the dynamic PDS directory increase function, an error code was returned when determining the ownership of the data set resource. Recovery cannot continue.

User response: Note the return code *rc* and contact IBM Software Support with the diagnostic information.

GLO1458W Error during GQSCAN termination processing, r15=*rc*.

Explanation: During the dynamic PDS directory increase function, an error code was returned when determining the ownership of the data set resource. Recovery cannot continue.

User response: Note the return code *rc* and contact IBM Software Support with the diagnostic information.

GLO1459W Directory increase requires exclusive control, but the data set is being shared.

Explanation: During the dynamic PDS directory increase function, Advanced Allocation Management required exclusive control of the data set to maintain the integrity of the PDS directory, but was unable to obtain it. Recovery cannot continue.

User response: If you still want the directory increase, stop any processes that have shared ownership of the data set and run the process again.

GLO1460E Directory increase requires exclusive control, but was unable to obtain it; r15=*rc*.

Explanation: During the dynamic PDS directory increase function, Advanced Allocation Management required exclusive control of the data set to maintain the integrity of the PDS directory, but an attempt to obtain it failed. Recovery cannot continue.

User response: Note the return code *rc* and contact IBM Software Support with the diagnostic information.

GLO1480E Unable to obtain storage for A7 intercept, rc=*rrrrrrrr*.

Explanation: During product initialization an error (*rrrrrrrr*) was encountered while attempting to obtain above-the-line common storage for a product control block.

User response: Investigate a potential shortage of above-the-line common storage and restart the product. If the problem persists, contact IBM Software Support.

GLO1482E Unable to locate LPDE for IDA0192A.

Explanation: During product initialization, Advanced Allocation Management could not locate a required pointer to an operating system module.

User response: Contact IBM Software Support.

GLO1484E Unable to locate IDAOCTBL.

Explanation: During product initialization, Advanced Allocation Management could not locate a required operating system module.

User response: Contact IBM Software Support.

GLO1485E Unable to locate IDAO557X.

Explanation: During product initialization, Advanced Allocation Management could not locate a required operating system module.

User response: Contact IBM Software Support.

GLO1488E Page service list exhausted for A7 intercept.

Explanation: During product initialization, Advanced Allocation Management encountered an internal error when it attempted to establish a product intercept.

User response: Contact IBM Software Support.

Function messages (non-VSAM, EAM, and JCL-allocated VSAM)

These messages are related to the Advanced Allocation Management functions that support non-VSAM, EAM, and JCL-allocated VSAM data sets.

GLO2000E Insufficient virtual storage for product processing.

Explanation: During an attempt to intercept an allocation request, Advanced Allocation Management could not obtain enough above-the-line or below-the-line storage to perform processing.

User response: Increase the amount of above-the-line and below-the-line storage for the job that encountered the error. If the error persists, contact IBM Software Support.

GLO2001E Storage obtain failure *xxxxxxxx* - Function bypassed -*ssss*.

Explanation: In support of the NOTCAT2 function, the product could not obtain the required amount of below-the-line storage for the Advanced Allocation Management subsystem *ssss*. The field *xxxxxxxx* contains diagnostic information.

User response: Increase the amount of below-the-line storage for the job step. If the problem persists, contact IBM Software Support.

GLO2002E Error occurred during SWAREQ processing for SCT, rc=*rrrrrrrr*.

Explanation: During interception of a primary allocation request, an internal error (*rrrrrrrr*) was encountered while attempting to access a system control block.

User response: Contact IBM Software Support.

GLO2008E Error occurred during SWAREQ processing for SIOT, rc=*rrrrrrrr*.

Explanation: During interception of a secondary allocation request, an internal error (*rrrrrrrr*) was encountered while attempting to access a system control block.

User response: Contact IBM Software Support.

GLO2009E Error occurred during SWAREQ processing for SCT, rc=*rrrrrrrr*.

Explanation: During interception of a secondary allocation request, an internal error (*rrrrrrrr*) was encountered while attempting to access a system control block.

User response: Contact IBM Software Support.

GLO2013E Error occurred during SWAREQ processing for SCT, rc=*rrrrrrrr*.

Explanation: During interception of a directory STOW request an internal error (*rrrrrrrr*) was encountered while attempting to access a system control block.

User response: Contact IBM Software Support.

GLO2019E Unable to locate IRWB in queue - *ssss*.

Explanation: During the processing of an allocation request, the product encountered an internal error for subsystem ID *ssss*.

User response: Contact IBM Software Support.

GLO2023E Error in name/token delete processing, rc=*rrrrrrrr*.

Explanation: During processing of an allocation request, the product encountered an internal system error (*rrrrrrrr*).

User response: Contact IBM Software Support.

GLO2025E Error in name/token create processing, rc=*rrrrrrrr*.

Explanation: During processing of an allocation request, the product encountered an internal system error (*rrrrrrrr*).

User response: Contact IBM Software Support.

GLO2031E Internal error in GIRWB000, rc=*rc*.

Explanation: An internal error occurred in routine GIRWB000. The return code is *rc*.

User response: Contact IBM Software Support.

GLO2032E Error occurred during SWAREQ processing for SCT, rc=*rrrrrrrr*.

Explanation: During interception of a primary allocation request, an internal error (*rrrrrrrr*) was encountered while attempting to access a system control block.

User response: Contact IBM Software Support.

GLO2038E Internal error in GIRWB000, rc=*rc*.

Explanation: An internal error occurred in routine GIRWB000. The return code is *rc*.

User response: Contact IBM Software Support.

GLO2040E Error occurred during SWAREQ processing for JCT, rc=*rrrrrrrr*.

Explanation: During interception of an allocation request, an internal error (*rrrrrrrr*) was encountered while attempting to access a system control block.

User response: Contact IBM Software Support.

GLO2041E Error occurred during SWAREQ processing for JMR, rc=*rrrrrrrr*.

Explanation: During interception of an allocation request, an internal error (*rrrrrrrr*) was encountered while attempting to access a system control block.

User response: Contact IBM Software Support.

GLO2043E Internal error in GIRWB000, rc=*rc*.

Explanation: An internal error occurred in routine GIRWB000 with a return code of *rc*.

User response: Contact IBM Software Support.

GLO2048E Error occurred during SWAREQ processing for SCT, rc=*rrrrrrrr*.

Explanation: During interception of an allocation request, an internal error (*rrrrrrrr*) was encountered while attempting to access a system control block.

User response: Contact IBM Software Support.

GLO2049I Duplicate data set but DSORG is VSAM - NOTCAT2 function skipped -*ssss*.

Explanation: A duplicate data set name was found for the NOTCAT2 function for Advanced Allocation Management subsystem *ssss*. Because the data set DSORG is VSAM, the NOTCAT2 function was skipped.

User response: No action is required.

GLO2051I Job set to fail due to NOT CATLGD 2 on volser *vvvvvv* -*ssss*.

Explanation: The current job step and subsequent steps were failed for Advanced Allocation Management subsystem *ssss* in support of the NOTCAT2=JCFAIL function. The steps were failed upon the detection of a NOT CATLGD 2 error at step termination during the attempt to catalog the data set. The data set is identified by message GLO2123I with its first volser of *vvvvvv*.

User response: No action is required.

GLO2061I Job set to fail due to duplicate data set name on *vvvvvv* at allocation -*ssss*.

Explanation: The current job step and subsequent steps were failed for Advanced Allocation Management subsystem *ssss* in support of the NOTCAT2=JCLFAIL function. The steps were failed upon the detection of a duplicate data set name at allocation time for the data set identified by message GLO2123I with its first volser of *vvvvvv*.

User response: No action is required.

GLO2070E Error occurred during SWAREQ processing for SIOT, rc=*rrrrrrrr*.

Explanation: During interception of an allocation request, an internal error (*rrrrrrrr*) was encountered during the attempt to access a system control block.

User response: Contact IBM Software Support.

GLO2074E DDPB queue error *xxxxxxxx*- SDUMP taken - Queue recovery successful.

Explanation: During processing, Advanced Allocation Management detected a problem with an internal control block queue. *xxxxxxxx* is an internal error code that is used by IBM Software Support.

User response: An SVCDUMP is created within the address space that detected the error. The title of the dump is "DDPB QUEUE ERROR." After the dump occurs, Advanced Allocation Management corrects the queue problem and continues processing. Save the SVCDUMP and contact IBM Software Support.

GLO2075E Error occurred during SWAREQ processing for SIOT, rc=*rrrrrrrr*.

Explanation: During interception of a step termination, an internal error (*rrrrrrrr*) was encountered during the attempt to access a system control block.

User response: Contact IBM Software Support.

GLO2076E Error occurred during RESMGR add processing, rc=*rrrrrrrr*.

Explanation: An attempt to add a resource manager was unsuccessful. The return code from the RESMGR invocation is *rrrrrrrr*.

User response: Contact IBM Software Support.

GLO2077I Waiting for RESMGR DELETE.

Explanation: During the attempt to delete a resource manager, the system might require the issuing program (Advanced Allocation Management) to wait before the deletion is complete.This message is issued in case the wait time becomes excessive.

User response: If Advanced Allocation Management returns from the wait relatively quickly, no action is required. If the wait time seems excessive, contact IBM Software Support.

GLO2078E Error occurred during RESMGR DELETE processing, rc=*rrrrrrrr*.

Explanation: An attempt to delete a resource manager was unsuccessful. The return code from the RESMGR invocation is *rrrrrrrr*.

User response: Contact IBM Software Support.

GLO2079E Internal error in GIRWB000, rc=*rrrrrrrr*.

Explanation: An internal error occurred in routine GIRWB000 with a return code of *rrrrrrrr*.

User response: Contact IBM Software Support.

GLO2080I Initializing data set.

Explanation: Through rule definition statements, Advanced Allocation Management was instructed to initialize a newly allocated data set. The data set initialization was successfully performed. Message GLO2123I accompanies this message and displays the job name, ddname, and data set name for the data set that is being initialized.

User response: No action is required.

GLO2081E Error occurred during OPEN processing for ddname=*dddddddd*, rc=*rrrrrrrr*.

Explanation: A request was made to initialize the data set represented by ddname *dddddddd*. During the OPEN portion of initialization, an error occurred. The return code from the OPEN operation is *rrrrrrrr*.

User response: Contact IBM Software Support.

GLO2082E OPEN unsuccessful for ddname=*dddddddd*.

Explanation: A request was made to initialize the data set represented by the ddname *dddddddd*. While the OPEN portion of initialization ended with a return code of zero, the OPEN operation was not successful.

User response: Contact IBM Software Support.

GLO2083E Error occurred during CLOSE processing for ddname=*dddddddd*, rc=*rrrrrrrr*.

Explanation: A request was made to initialize the data set represented by ddname *dddddddd*. During the CLOSE portion of initialization, an error occurred. The return code from the CLOSE operation is *rrrrrrrr*.

User response: Contact IBM Software Support.

GLO2083I Data set on volser *oooooo* scratched due to NOT CATLGD 2 on volser *vvvvvv* -*ssss*.

Explanation: The data set that is named in message GLO2123I with its first volser on volume *oooooo* was deleted in support of the NOTCAT2=SCRATCH function. The data set was deleted at step termination time for the new data set with its first volser located on volume *vvvvvv* on behalf of the Advanced Allocation Management subsystem *ssss*.

User response: Contact IBM Software Support.

GLO2084E Error occurred during OBTAIN processing for ddname=*dddddddd*, rc=*rrrrrrrr*.

Explanation: A request was made to initialize the data set represented by ddname *dddddddd*. An error occurred during OBTAIN processing for the data set. The return code from OBTAIN is *rrrrrrrr*.

User response: Contact IBM Software Support.

GLO2084I Data set on volser *oooooo* scratched at allocation time -*ssss*.

Explanation: The data set that is named in message GLO2123I with its first volser on volume *oooooo* was deleted in support of the NOTCAT2=SCRATCH function at allocation time on behalf of the Advanced Allocation Management subsystem *ssss*.

User response: No action is required.

GLO2085E Abend occurred during DATASET_INIT processing for ddname=*dddddddd* (*ccccc*).

Explanation: A request was made to initialize the data set represented by ddname *dddddddd*. During initialization processing an abend occurred. *ccccc* contains the abend type and code.

User response: Contact IBM Software Support.

GLO2085I **Data set on volser** *oooooo* **renamed due to NOT CATLGD 2 on volser** *vvvvvv* **-***ssss***.**

Explanation: The data set that is named in message GLO2123I with its first volser on volume *oooooo* was renamed in support of the NOTCAT2=RENAME function. The data set was renamed at step termination time for the new data set with its first volser located on volume *vvvvvv* on behalf of the Advanced Allocation Management subsystem *ssss*. The new name of the renamed data set is specified in message GLO2086I.

User response: No action is required.

GLO2086I **New name:** *datasetname* **-***ssss***.**

Explanation: This message displays the new name (*datasetname*) of a data set that was renamed as a result of the NOTCAT2=RENAME function.

User response: No action is required.

GLO2087I **Data set on volser** *oooooo* **renamed at allocation time -***ssss***.**

Explanation: The data set that is named in message GLO2123I with its first volser on volume *oooooo* was renamed in support of the NOTCAT2=RENAME function at allocation time on behalf of the Advanced Allocation Management subsystem *ssss*. The new name of the renamed data set is specified in message GLO2086I.

User response: No action is required.

GLO2088I **Data set on volser** *oooooo* **uncataloged due to NOT CATLGD 2 on volser** *vvvvvv* **-***ssss***.**

Explanation: The data set that is named in message GLO2123I with its first volser on volume *oooooo* was uncataloged in support of the NOTCAT2=UNCATALOG function. The data set was uncataloged at step termination time for the new data set with its first volser located on volume *vvvvvv* on behalf of the Advanced Allocation Management subsystem *ssss*.

User response: No action is required.

GLO2089I **Data set on volser** *oooooo* **uncataloged at allocation time -***ssss***.**

Explanation: The data set that is named in message GLO2123I with its first volser on volume *oooooo* was uncataloged in support of the NOTCAT2=UNCATALOG function at allocation time on behalf of the Advanced Allocation Management subsystem *ssss*.

User response: No action is required.

GLO2090E **NOTCAT2** *ffffffff* **failure in service routine, rc=***rcrcrcrc* **-***ssss***.**

Explanation: A failure (*ffffffff*) occurred in support of the NOTCAT2 function with the return code *rcrcrcrc*. The Advanced Allocation Management subsystem is *ssss*.

User response: Contact IBM Software Support.

GLO2091E **NOTCAT2** *ffffffff* **failure in invoked service - rc=***rcrcrcrc***, R15=***aaaaaaaa***, R0=***bbbbbbbb***, offset=***oooooooo* **-***ssss***.**

Explanation: A failure (*ffffffff*) occurred in support of the NOTCAT2 function with the return code *rcrcrcrc*. The Advanced Allocation Management subsystem is *ssss*. *aaaaaaaa*, *bbbbbbbb*, and *oooooooo* contain diagnostic information for IBM Software Support.

User response: Contact IBM Software Support.

GLO2092E **Volume list retrieval error** *eeeeeeee* **- NOTCAT2 processing skipped -***ssss***.**

Explanation: A failure identified by the failure code in *eeeeeeee* occurred in support of the NOTCAT2 function. The failure occurred during the attempt to perform a catalog LOCATE operation on behalf of Advanced Allocation Management subsystem *ssss*.

User response: Contact IBM Software Support.

GLO2100I **Setting secondary allocation to** *nnnnnq uuuu -ssss***.**

Explanation: Advanced Allocation Management is setting the secondary allocation to the value *nnnnn* for the DDNAME that is referenced in message GLO2123I on behalf of subsystem *ssss*. *q* corresponds to the JCL parameter AVGREC and indicates K (kilobytes) or M (megabytes) and is only present when SMS managed data sets are used and when the allocation unit is RCDS. *uuuu* indicates CYL, TRKS, BLKS, or RCDS. This message is issued in response to the SEC_ALLOC function and is followed by message GLO2123I.

User response: No action is required.

GLO2101E **SBST0500 non-zero return from VAE, R15=***rcrcrcrc* **–** *ssss***.**

Explanation: Advanced Allocation Management encountered an internal error (*rcrcrcrc*) while it was performing the SEC_BEST function on behalf of subsystem *ssss*.

User response: Contact IBM Software Support.

GLO2102E SBST0600 non-zero return from VAE, R15=*rcrcrcrc*, R0=*rsrsrsrs*, at offset *oooooooo* – *ssss*.

Explanation: Advanced Allocation Management encountered an internal error that is identified by return code *rcrcrcrc*, reason code *rsrsrsrs*, and offset *oooooooo*, while it was performing the SEC_BEST function on behalf of subsystem *ssss*.

User response: Contact IBM Software Support.

GLO2103I Extent-matching secondary allocation to *nnnnnq uuuu -ssss*.

Explanation: Advanced Allocation Management is setting the secondary allocation to the value *nnnnn* for the DDNAME that is referenced in message GLO2123I on behalf of subsystem *ssss*. *q* corresponds to the JCL parameter AVGREC and indicates K (kilobytes) or M (megabytes) and is only present when SMS managed data sets are used and when the allocation unit is RCDS. *uuuu* indicates CYL, TRKS, BLKS, or RCDS. This message is issued in response to the SEC_BEST function and is followed by message GLO2123I.

User response: No action is required.

GLO2104E SINC0500 non-zero return from VAE, R15=*rcrcrcrc* – *ssss*.

Explanation: Advanced Allocation Management encountered an internal error (*rcrcrcrc*) while it was performing the SEC_INC function on behalf of subsystem *ssss*.

User response: Contact IBM Software Support.

GLO2105E SINC0600 non-zero return from VAE, R15=*rcrcrcrc*, R0=*rsrsrsrs*, at offset *oooooooo* – *ssss*.

Explanation: Advanced Allocation Management encountered an internal error that is identified by the return code *rcrcrcrc*, reason code *rsrsrsrs,* and offset *oooooooo*, while it was performing the SEC_INC function on behalf of subsystem *ssss*.

User response: Contact IBM Software Support.

GLO2106I Increasing secondary allocation to *nnnnnq uuuu - ssss*.

Explanation: Advanced Allocation Management is setting the secondary allocation to the value *nnnnn* for the DDNAME that is referenced in message GLO2123I on behalf of subsystem *ssss*. *q* corresponds to the JCL parameter AVGREC and indicates K (kilobytes) or M (megabytes) and is only present when SMS managed data sets are used and when the allocation unit is RCDS. *uuuu* indicates CYL, TRKS, BLKS, or RCDS. This message is issued in response to the SEC_INC function and is followed by message GLO2123I.

User response: No action is required.

GLO2107E SRED0500 non-zero return from VAE, R15=*rcrcrcrc* – *ssss*.

Explanation: Advanced Allocation Management encountered an internal error (*rcrcrcrc*) while it was performing the SEC_REDUCE function on behalf of subsystem *ssss*.

User response: Contact IBM Software Support.

GLO2108E SRED0600 non-zero return from VAE, R15=*rcrcrcrc*, R0=*rsrsrsrs*, AT OFFSET *oooooooo* – *ssss*.

Explanation: Advanced Allocation Management encountered an internal error that is identified by the return code *rcrcrcrc*, reason code *rsrsrsrs*, and offset *oooooooo*, while it was performing the SEC_REDUCE function on behalf of subsystem *ssss*.

User response: Contact IBM Software Support.

GLO2109I Reducing secondary allocation to *nnnnnq uuuu -ssss*.

Explanation: Advanced Allocation Management is setting the secondary allocation to the value *nnnnn* for the DDNAME that is referenced in message GLO2123I on behalf of subsystem *ssss*. *q* corresponds to the JCL parameter AVGREC and indicates K (kilobytes) or M (megabytes) and is only present when SMS managed data sets are used and when the allocation unit is RCDS. *uuu* indicates CYL, TRKS, BLKS, or RCDS. This message is issued in response to the SEC_REDUCE function and is followed by message GLO2123I.

User response: No action is required.

GLO2112I Allocation successfully recovered for dsn=*dsname* – *ssss*.

Explanation: The primary allocation for data set *dsname* was successfully reduced on behalf of subsystem *ssss*. This message is issued in response to the PRIM_REDUCE function.

User response: No action is required.

GLO2113E SPCHK010 non-zero return from BVIRB000, rc=*rrrrrrrr* – *ssss*.

Explanation: Advanced Allocation Management encountered an internal error (*rrrrrrrr*) while processing the primary reduction function on behalf of subsystem *ssss*.

User response: Contact IBM Software Support.

GLO2114E SPCHK060 non-zero return from VAE, R15=*rcrcrcrc* **–** *ssss***.**

Explanation: Advanced Allocation Management encountered an internal error (*rcrcrcrc*) while it was performing the PRIM_REDUCE function on behalf of subsystem *ssss*.

User response: Contact IBM Software Support.

GLO2115E SPCHK070 non-zero return from VAE, R15=*rcrcrcrc***, R0=***rsrsrsrs***, AT OFFSET** *oooooooo* **–** *ssss***.**

Explanation: Advanced Allocation Management encountered an internal error that is identified by return code *rcrcrcrc*, reason code *rsrsrsrs*, and offset *oooooooo*, while it was performing the PRIM_REDUCE function on behalf of subsystem *ssss*.

User response: Contact IBM Software Support.

GLO2116I Unable to reduce primary space quantity; current value is 0 or 1 *ssss***.**

Explanation: While performing the PRIM_REDUCE function on behalf of subsystem *ssss*, Advanced Allocation Management was not able to reduce the primary quantity further because the current value is 0. Message GLO2123I identifies the DDNAME that encountered the error.

User response: Contact IBM Software Support.

GLO2117I Space reduction of *ppp%* **would reduce primary quantity beyond minimum limit** *ssss***.**

Explanation: Advanced Allocation Management could not reduce the primary quantity further by the percentage value *ppp* as requested by the PRIM_REDUCE function on behalf of subsystem *ssss* because doing so would lower the primary quantity below the limit value that is defined by PRIM_REDUCE_LIM. Message GLO2123I identifies the DDNAME that encountered the situation.

User response: No action is required.

GLO2118I Reducing primary allocation to *nnnnn uuuuuuuuu* **-***ssss***.**

Explanation: Advanced Allocation Management is reducing the primary allocation quantity to the value *nnnnn* on behalf of subsystem *ssss* where *uuuuuuuu* indicates TRACKS, CYLINDERS, or BLKS/RCDS. This message is issued in response to the PRIM_REDUCE function. Message GLO2123I identifies the DDNAME for which the reduction is being performed.

User response: No action is required.

GLO2119E JFCB validation failed *ssss***.**

Explanation: An internal error occurred while performing the PRIM_REDUCE function on behalf of subsystem *ssss*. Message GLO2123I identifies the DDNAME that encountered the error.

User response: Contact IBM Software Support.

GLO2123I *jjjjjjjj***,***ssssssss***,***pppppppp***,***dddddddd***,***dsn ssss*

Explanation: This message identifies the jobname (*jjjjjjjj*), stepname (*ssssssss*), procedure stepname (*pppppppp*), ddname (*dddddddd*), and data set name (*dsn*) associated with the preceding message that is associated with subsystem *ssss*.

User response: No action is required.

GLO2124E Unrecognized intercept ID encountered (*nn***) -***ssss***.**

Explanation: Unrecognizable control information (*nn)* was passed from an intercept point to the intercept handling routine for subsystem *ssss*.

User response: Contact IBM Software Support.

GLO2125I Space release recovered *nnnnnnnnn* **tracks -***ssss***.**

Explanation: The space release option, which is equivalent to the RLSE option of the SPACE keyword of the DD statement, was set for the data set that is identified by the message GLO2123I on behalf of subsystem *ssss*. *nnnnnnnnn* is the number of tracks that are recovered and is associated with the Advanced Allocation Management SPACE_RELEASE function.

User response: No action is required.

GLO2127E Error occurred during SWAREQ processing for SIOT, rc=*rrrrrrrr* **-** *ssss***.**

Explanation: While performing the VOL_ADD function, Advanced Allocation Management encountered an internal error (*rrrrrrrr*) during the attempt to access a system control block for subsystem *ssss*.

User response: Contact IBM Software Support.

GLO2128I Data set successfully extended to volume *vvvvvv* **-***ssss***, Total VOLCNT=***ccc***.**

Explanation: The data set that is identified by the message GLO2123I successfully processed the VOL_ADD function and extended the data set to the volume serial number *vvvvvv* on behalf of subsystem *ssss*. Total VOLCNT is the total volume count.

User response: No action is required.

GLO2129E Error occurred during SWAREQ processing for AMPX, rc=*rrrrrrrr* -*ssss*.

Explanation: During SELECTION processing, an internal error (*rrrrrrrr*) was encountered during the attempt to access a system control block.

User response: Contact IBM Software Support.

GLO2130E Error occurred during SWAREQ processing for JFCBX, rc=*rrrrrrrr* -*ssss*.

Explanation: During SELECTION processing, an internal error (*rrrrrrrr*) was encountered during the attempt to access a system control block.

User response: Contact IBM Software Support.

GLO2130E Unrecognized intercept ID encountered (*nn*).

Explanation: Unrecognizable control information (*nn*) was passed from an intercept point to the intercept handling routine.

User response: Contact IBM Software Support.

GLO2131E Volume not added due to volume add limit reached -*ssss*.

Explanation: A volume addition could not be performed because the operating system limit of 59 volumes or the volume limit that is specified by the VOL_ADD_LIM keyword was reached. Message GLO2123I identifies the data set.

User response: Increase the size of each extent or raise the limit that is specified by the VOL_ADD_LIM keyword.

GLO2133E Unexpected return from VAE, R15=*rcrcrcrc* - *ssss*.

Explanation: Advanced Allocation Management encountered an internal error that is identified by the return code *rcrcrcrc* was while it was performing the VOL_ADD function on behalf of subsystem *ssss*.

User response: Contact IBM Software Support.

GLO2134E Non-zero return from VAE, R15=*rcrcrcrc*, R0=*rsrsrsrs*, at offset *oooooooo* - *ssss*.

Explanation: Advanced Allocation Management encountered an internal error that is identified by the return code *rcrcrcrc*, reason code *rsrsrsrs* and offset *oooooooo* while it was performing the VOL_ADD function on behalf of subsystem *ssss*.

User response: Contact IBM Software Support.

GLO2135E SVOL2240 non-zero return from VAE, R15=*rcrcrcrc* - *ssss*.

Explanation: Advanced Allocation Management encountered an internal error that is identified by the return code *rcrcrcrc* while it was performing the VOL_ADD function on behalf of subsystem *ssss*.

User response: Contact IBM Software Support.

GLO2136E Non-zero return from VAE, R15=*rcrcrcrc*, R0=*rsrsrsrs*, at offset *oooooooo* - *ssss*.

Explanation: Advanced Allocation Management encountered an internal error that is identified by the return code *rcrcrcrc*, reason code *rsrsrsrs*, and offset *oooooooo*, while it was performing the VOL_ADD function on behalf of subsystem *ssss*.

User response: Contact IBM Software Support.

GLO2137A Specify volser for *jobname*, *stepname*, *procstepname*, *ddname* or reply CANCEL -*ssss*.

Explanation: In support of the VOL_ADD_PROMPT keyword, specify a volume serial number to extend the data set that is associated with the job *jobname*, the step *stepname*, the procedure stepname *procstepname*, and the DDNAME *ddname* for subsystem *ssss*.

User response: Specify a six character volume serial number to use to extend the data set, or reply with the **CANCEL** command to fail the allocation.

GLO2138I Operator responded to volume request with *vvvvvv* -*ssss*.

Explanation: In response to message GLO2137A that is associated with the VOL_ADD_PROMPT function, the operator responded with volume serial number *vvvvvvv* for subsystem *ssss*.

User response: No action is required.

GLO2139E Volume add not performed - unable to locate suitable candidate volume -*ssss*.

Explanation: After searching all of the volumes to be used for locating a new volume, no volumes with adequate space were found to extend the data set to another volume. *ssss* is the Advanced Allocation Management subsystem ID. Message GLO2123I identifies the data set.

User response: Add more volumes to be searched, free up space on the existing volumes, or include the VOL_ADD_PROMPT keyword and supply a volume with adequate space.

GLO2141E Unexpected error *rcrcrcrc* during volume add -*sss*.

Explanation: An internal error (*rcrcrcrc*) occurred during processing of the VOL_ADD function for subsystem *ssss*.

User response: Contact IBM Software Support.

GLO2142E PRED0700 non-zero return from VAE, R15=*rcrcrcrc* - *ssss*.

Explanation: Advanced Allocation Management encountered an internal error that is identified by the return code *rcrcrcrc* while it was performing the PRIM_REDUCE function for a secondary volume on behalf of subsystem *ssss*.

User response: Contact IBM Software Support.

GLO2143E PRED4000 non-zero return from VAE, R15=*rcrcrcrc*, R0=*rsrsrsrs*, at offset *oooooooo* - *ssss*.

Explanation: Advanced Allocation Management encountered an internal error that is identified by the return code *rcrcrcrc*, reason code *rsrsrsrs*, and offset *oooooooo,* while it was performing the PRIM_REDUCE function for a secondary volume on behalf of subsystem *ssss*.

User response: Contact IBM Software Support.

GLO2144I Setting initial allocation on volume *vvvvvv* to *nnnnnq uuuu* -*ssss*.

Explanation: In response to successful completion of the PRIM_REDUCE function on a secondary volume that is identified by *vvvvvv* for subsystem *ssss*, the initial allocation was set to *nnnnn*. *q* corresponds to the JCL parameter AVGREC and indicates K (kilobytes) or M (megabytes) and is present only when SMS managed data sets are used and when the allocation unit is RCDS. *uuuu* indicates CYL, TRKS, BLKS, or RCDS. Message GLO2123I, identifies the data set.

User response: No action is required.

GLO2145I Volume ADD not performed due to multiple opens against data set.

Explanation: A VOL_ADD operation was requested, but could not be performed because more than one DCB is open concurrently against the non-VSAM data set and `BYPASS_CHECK=MULTOPEN` was not specified. Performing a VOL_ADD might compromise data set integrity or cause unpredictable results.

User response: If you are certain that the executing program can properly handle the multiple concurrent OPENs against the data set, specify `BYPASS_CHECK=MULTOPEN` to bypass the integrity check. For more information, see "Bypass integrity check processing (BYPASS_CHECK)" on page 65.

GLO2146E AVS not performed - Unable to locate suitable candidate volume(s)-*xxxxx* -*ssys*.

Explanation: While Advanced Allocation Management was performing AVS processing for a non-VSAM non-SMS data set, it was unable to locate candidate volumes to fulfill the requested number of volumes for the data set based on the chosen AVS algorithm. Normal system allocation is used to allocate the data set. *ssys* is the Advanced Allocation Management subsystem identifier.

User response: Verify the following items:

- A volume group, unit name list, or volser list was specified in the rule definition for the allocation
- Specified volume groups are defined in a volume group definition
- Adequate space exists in the specified volumes, based on the chosen AVS algorithm

GLO2147I Volume ADD not performed due to multiple DDs assigned to data set.

Explanation: A VOL_ADD operation was requested, but could not be performed because more than one DD was allocated to the non-VSAM data set and `BYPASS_CHECK=MULTDD` was not specified. Performing a VOL_ADD might compromise data set integrity or cause unpredictable results.

User response: If you are certain that the executing program can properly handle multiple DDs allocated to the data set, specify `BYPASS_CHECK=MULTDD` to bypass the integrity check. For more information, see "Bypass integrity check processing (BYPASS_CHECK)" on page 65.

GLO2148I AVS has successfully selected all volumes-*xxxxx* -*ssys*.

Explanation: AVS located all of the requested volumes for the non-VSAM non-SMS data set as requested using the specified AVS algorithm. Message GLO2156I displays the selected volumes.

User response: No action is required.

GLO2149E AVS unable to fulfill volume count request due to insufficient volumes-*xxxxx* -*ssys*.

Explanation: AVS was unable to locate the requested number of volumes for a non-VSAM non-SMS data set as requested using the specified AVS algorithm. The volume list for the component is truncated to the number of volumes that were located. *ssys* is the Advanced Allocation Management subsystem identifier. Message GLO2156I displays the selected volumes.

User response: Verify the following items:

- A volume group, unit name list, or volser list was specified in the rule definition for the allocation.
- Specified volume groups are defined in a volume group definition.
- Adequate space exists in the specified volumes, based on the chosen AVS algorithm.

GLO2150I Increasing PDS directory size -*ssss*.

Explanation: Advanced Allocation Management detected and successfully recovered a DIRECTORY FULL condition. The directory size was dynamically increased so that STOW processing for the current member could be successfully completed.

User response: No action is required.

GLO2151W Insufficient space in primary extent to increase PDS directory size -*ssss*.

Explanation: During an attempt to recover from a DIRECTORY FULL condition, Advanced Allocation Management determined that there was insufficient space in the data set's primary extent to increase the size of the directory. The recovery was unsuccessful and system error processing ended the process.

User response: Reallocate the data set and manually increase the size of the directory. Run the process that originally attempted to add the member to the data set again.

GLO2152W Insufficient space in data set to increase PDS directory size -*ssss*.

Explanation: During an attempt to recover from a DIRECTORY FULL condition, Advanced Allocation Management determined that member data must be moved to make space for the new directory block, but insufficient space in the data set did not permit the member data to be moved. The recovery was unsuccessful and system error processing ended the process.

User response: Compressing the data set might free enough space so that the member data can be moved. If so, compress the data set and then run the process that originally attempted to add the member to the data set again. Advanced Allocation Management will detect the DIRECTORY FULL condition, move the necessary members, and then dynamically increase the directory size.

If compressing the data set does not free enough space so that the member data can be moved, reallocate the data set with more primary and secondary space, if needed. Increase the number of directory blocks at the same time. Run the process that originally attempted to add the member to the data set again.

GLO2153W Data set not supported for PDS directory increase due to (*reason*) -*ssss*.

Explanation: During an attempt to recover from a DIRECTORY FULL condition, Advanced Allocation Management detected one of the following unsupported conditions as indicated by *reason*:

- TTR LIST IN USER DATA
- TEXT IN USER DATA
- NOTE LIST IN USER DATA
- OVERLAY USAGE
- ALIAS USAGE

The recovery process stops and system error processing ends the process.

User response: The PDS directory increase function is not supported with the type of PDS usage. To prevent the message from being displayed, add an exclusion for the entity that is being processed to the Advanced Allocation Management rule definitions member.

GLO2154E I/O error detected during PDS directory increase function -*ssss*.

Explanation: During an attempt to recover from a DIRECTORY FULL condition, Advanced Allocation Management determined that member data must be moved to make space for the new directory block. While moving the member data, an I/O error occurred and prevented the process from successfully completing. The recovery process stops and system error processing ends the process. Message GLO2155E describes the I/O error.

User response: Verify that the data in the members is valid. If so, reallocate the data set and increase the size of the directory. If the data in the existing members is not valid, restore the data set from a backup. Then, reallocate the data set, increase the size of the directory, and run the process that originally attempted to add the member to the data set again.

GLO2155E (text from the system SYNAD routine) -*ssss nnnn -ssss*

Explanation: This message accompanies message GLO2154E and provides details about the I/O error that was encountered.

User response: Perform the actions described in the User Response for message GLO2154E, record the contents of this message, and contact IBM Software Support.

GLO2156I Volser(s): *vvvvvv vvvvvvv vvvvvvv vvvvvvv vvvvvvv vvvvvvv vvvvvvv -ssys*

Explanation: This message lists a maximum of seven volume serial numbers as chosen by AVS processing for a non-VSAM data set. If more than seven volumes

were allocated, multiple GLO2156I messages are displayed.

User response: No action is required.

GLO2157I Multiple open check bypassed for volume add.

Explanation: Multiple concurrent OPENs were found during a VOL_ADD operation for a non-VSAM data set and `BYPASS_CHECK=MULTOPEN` was specified. The VOL_ADD operation is attempted.

User response: No action is required.

GLO2158I Multiple DD check bypassed for volume add.

Explanation: Multiple DDs assigned to the data set were found during a VOL_ADD operation for a non-VSAM data set and `BYPASS_CHECK=MULTDD` was specified. The VOL_ADD operation is attempted.

User response: No action is required.

GLO2160E Error occurred during SWAREQ processing for JCT, rc=*rrrrrrrr*.

Explanation: During interception of an allocation request, an internal error (*rrrrrrrr*) was encountered while attempting to access a system control block.

User response: Contact IBM Software Support.

GLO2161E Error occurred during SWAREQ processing for JMR, rc=*rrrrrrrr*.

Explanation: During interception of an allocation request, an internal error (*rrrrrrrr*) was encountered during the attempt to access a system control block.

User response: Contact IBM Software Support.

GLO2162E Error occurred during SWAREQ processing for SCT, rc=*rrrrrrrr*.

Explanation: During interception of an allocation request, an internal error (*rrrrrrrr*) was encountered during the attempt to access a system control block.

User response: Contact IBM Software Support.

GLO2163E Error occurred during SWAREQ processing for SIOT, rc=*rrrrrrrr*.

Explanation: During interception of an allocation request, an internal error (*rrrrrrrr*) was encountered during the attempt to access a system control block.

User response: Contact IBM Software Support.

GLO2165E Internal error in GIRWB000, rc=*rrrrrrrr*.

Explanation: An internal error occurred in routine GIRWB000 with a return code of *rrrrrrrr*.

User response: Contact IBM Software Support.

GLO2188I Matching *xxxxxxx* RULEDEF line number = *nnnnn* -*ssss*.

Explanation: Advanced Allocation Management displays this message upon a rule definition match during various points during non-VSAM processing in response to the MSG_LEVEL=RULEDEF option or the specification of the AOMATCH DD statement. *xxxxxxx* is the text "INCLUDE" or "EXCLUDE". INCLUDE denotes a match on an INCLUDE rule definition. EXCLUDE denotes a match on an EXCLUDE rule definition. *nnnnn* is the rule definition line number and *ssss* is the Advanced Allocation Management subsystem identifier.

User response: No action is required.

GLO2189I Waiting on resources for 15 minutes - retrying -*ssss*.

Explanation: A resource that Advanced Allocation Management requires to perform a volume addition was unavailable for fifteen minutes. *ssss* is the Advanced Allocation Management subsystem ID. Advanced Allocation Management continues to attempt to acquire the resource.

User response: No action is required.

GLO2190I Entity processed by RULEDEF_TEST mode; RULEDEF line number *nnnnn* *-ssss* **.**

Explanation: The data set that is being processed matches a rule definition statement that designates simulate mode and identifies the line number of the matching statement in the active rule definitions. No recovery action is performed for the data set. This message can be displayed more than once during the life of an allocation through unallocation for a data set. Message GLO2123I identifies the data set that is involved. This message applies to non-VSAM data sets only.

User response: No action is required.

GLO2194I Set processing complete; *keyword*=*newvalue* (old value=*oldvalue*).

Explanation: This message is issued when Advanced Allocation Management successfully processes a SET request. *keyword* is the keyword that is being processed. *newvalue* is the new value for the keyword, and *oldvalue* is the previous value of the keyword.

Response: No action is required.

GLO2195E Failure during consistency check for set processing; reason code *xxx*.

Explanation: During SET processing, a consistency check failed with reason code *xx*, where *xx* is one of the following codes:

01 One or more of the following parameters were implicitly or explicitly set to dataclas-override (DCO), but not all of them were set to DCO and the original allocation did not specify AVGREC and average-record-length:
- SPACEUNITS
- PQTY
- SQTY

User response: Change the SET values for SPACEUNITS, PQTY, and (or) SQTY so that DCO is used for all of the parameters (or none of them) or change the original allocation to use AVGREC and average-record-length.

02 SPACEUNITS, PQTY, and SQTY were implicitly or explicitly set to dataclas-override (DCO), but AVGVALUE, PQTY, and (or) SQTY was not defined in the dataclas.

User response: Add values to the dataclas for AVGVALUE, PQTY, and (or) SQTY, or change the SET statements so that DCO is not used for SPACEUNITS, PQTY, or SQTY.

03 One or more of the following parameters were set to $NULL, but not all of them were set to $NULL:
- SPACEUNITS
- PQTY
- SQTY

User response: Change the SET values for SPACEUNITS, PQTY, and SQTY, so that $NULL is used for all of them (or none of them).

04 SPACEUNITS is being implicitly or explicitly changed, but no corresponding PQTY change was specified.

User response: Add a SET statement for PQTY.

05 SPACEUNITS is being implicitly or explicitly changed, but no corresponding SQTY change was specified.

User response: Add a SET statement for SQTY.

06 SPACEUNITS is being implicitly or explicitly set to AVGRECU, AVGRECK, or AVGRECM, but the record-size was not specified.

User response: Specify the record-size subparameter on the SET_SPACEUNITS statement.

Message GLO2123I accompanies this message. For more information, see "Enhanced Allocation Management functions" on page 126.

GLO2197E Error returned from UCBPIN/UNPIN for UCB at *uuuuuuuu*, rc=*rrrrrrrr*, offset=*oooooooo* -*ssss*.

Explanation: An failure occurred in UCBPIN/UNPIN processing for the UCB at virtual storage location *uuuuuuuu*. The return code is *rrrrrrrr* and the internal error offset is *oooooooo* (for internal diagnostic use only). The Advanced Allocation Management subsystem is *ssss*.

User response: Contact IBM Software Support.

GLO2200E Unable to perform update for *xxxxxxxx* -*ssss*.

Explanation: During SET processing for allocation variable *xxxxxxxx*, a system service did not complete successfully. SET processing was not performed.

User response: Contact IBM Software Support.

GLO2201E Unable to create JFCBX(S) during SET_VOLCT processing, rc=*cccccccc*, rs=*ssssssss* -*ssss*.

Explanation: During SET processing for VOLCT, an error was returned by the SWAREQ ASSIGN system service. The return code is *cccccccc* and the reason code is *ssssssss*.

User response: Contact IBM Software Support.

GLO2202E Non-zero return from SJFREQ RETRIEVE, rc=*cccccccc*, rs=*ssssssss* -*ssss*.

Explanation: During SELECTION processing, an error was returned by the SJFREQ RETRIEVE system service. The return code is *cccccccc* and the reason code is *ssssssss*.

User response: Contact IBM Software Support.

GLO2203E Non-zero return from SJFREQ update, rc=*cccccccc*, rs=*ssssssss* -*ssss*.

Explanation: During SET processing, an error was returned by the SJFREQ UPDATE system service. The return code is *cccccccc* and the reason code is *ssssssss*.

User response: Contact IBM Software Support.

GLO2204E Non-zero return from SJFREQ ERASE, rc=*cccccccc*, rs=*ssssssss* -*ssss*.

Explanation: During SET processing, an error was returned by the SJFREQ ERASE system service. The return code is *cccccccc* and the reason code is *ssssssss*.

User response: Contact IBM Software Support.

GLO2900E Invalid storage request for control block *nnnn* -*ssss*.

Explanation: An internal error occurred while attempting to obtain a control block identified by *nnnn* subsystem ID *ssss*.

User response: Contact IBM Software Support.

GLO2901E Insufficient virtual storage for control block *nnnn* -*ssss*.

Explanation: Sufficient storage was not available to obtain a required control block that is identified by *nnnn* subsystem ID *ssss*.

User response: Attempt to increase above-the-line or below-the-line storage for the job that received the error message. If the error persists, contact IBM Software Support.

GLO2902E Acronym check failed while attempting to free *nnnn*, data=*dddd* -*ssss*.

Explanation: An internal error occurred during the attempt to free control block *nnnn* with the incorrect data *dddd* for subsystem ID *ssss*.

User response: Contact IBM Software Support.

GLO2903E Failure occurred during FREEMAIN for *nnnn* -*ssss*.

Explanation: An internal error occurred during the attempt to free the control block *nnnn* for subsystem ID *ssss*.

User response: Contact IBM Software Support.

GLO2904E Error occurred during SWAREQ processing for SIOT, rc=*rrrrrrrr*.

Explanation: During interception of a secondary allocation request, an internal error (*rrrrrrrr*) was encountered during the attempt to access a system control block.

User response: Contact IBM Software Support.

GLO2999I Allocation terminated as requested by RULEDEF line *nnnnn* - *ssid*.

Explanation: In accordance with the TERM_ALLOC function, the data set allocation was stopped as requested by rule definition line number *nnnnn* for the Advanced Allocation Management subsystem *ssid*.

User response: No action is required.

Function messages (VSAM)

These messages are related to the Advanced Allocation Management functions that support VSAM data sets.

GLO3100I Setting secondary allocation to *nnnnn uuuu*.

Explanation: Advanced Allocation Management is setting the secondary allocation to the value *nnnnn* for the DDNAME that is referenced in message GLO3123I on behalf of subsystem *ssss*. *uuuu* indicates CYL S or TRKS. This message is issued in response to the SEC_ALLOC function for VSAM and is followed by message GLO3123I.

User response: No action is required.

GLO3103I Extent-matching *ttttt* secondary allocation to *nnnnn uuuuu* -*ssss*.

Explanation: Advanced Allocation Management is setting the secondary allocation to the value *nnnnn* for the DDNAME that is referenced in message GLO3123I on behalf of subsystem *ssss*. *uuuu* indicates CYL S or TRKS. *ttttt* indicates the component (INDEX or DATA) for which the action was performed. This message is issued in response to the SEC_BEST function for VSAM and is followed by message GLO3123I.

User response: No action is required.

GLO3109I Reducing *ttttt* secondary allocation to *nnnnn uuuu* -*ssss*.

Explanation: Advanced Allocation Management is setting the secondary allocation to the value *nnnnn* for the DDNAME that is referenced in message GLO3123I on behalf of subsystem *ssss*. *uuuu* indicates CYL S or TRKS. The value *ttttt* is INDEX or DATA, which indicates the component for which the action was performed. This message is issued in response to the SEC_REDUCE function for VSAM and is followed by message GLO3123I.

User response: No action is required.

GLO3116I Unable to reduce primary space quantity; current value is 0 or 1 -*ssss*.

Explanation: While performing the PRIM_REDUCE function for a VSAM data set on behalf of subsystem *ssss*, Advanced Allocation Management was not able to reduce the primary quantity further because the current value is 0. Message GLO3123I identifies the ddname that encountered the error.

User response: No action is required.

GLO3117I Space reduction of *ppp*% would reduce primary quantity beyond minimum limit -*ssss*.

Explanation: Advanced Allocation Management could not reduce the primary quantity for a VSAM data set further by the percentage value *ppp* as requested by the PRIM_REDUCE function on behalf of subsystem *ssss* because doing so would lower the primary quantity below the limit value that is defined by the PRIM_REDUCE_LIM option. Message GLO3123I identifies the ddname that encountered the situation.

User response: No action is required.

GLO3118I Reducing primary allocation to *nnnnn uuuuuuuuuu* -*ssss*.

Explanation: Advanced Allocation Management is reducing the primary allocation quantity for a VSAM data set to the value *nnnnn* on behalf of subsystem *ssss*. *uuuuuuuu* indicates TRACKS, CYLINDERS, RECORDS, MEGABYTES, or KILOBYTES. This message is issued in response to the PRIM_REDUCE function. Message GLO3123I identifies the ddname for which reduction is being performed.

User response: No action is required.

GLO3123I *jobname*, *stepname*, *pstepname*, *ddname*, *dsname*, -*ssss*

Explanation: This message identifies the job name (*jobname*), step name (*stepname*), procedure step name (*pstepname*), ddname (*ddname*), and data set name (*dsname*) associated with the preceding message that is associated with subsystem -*ssss* for a VSAM data set.

User response: No action is required.

GLO3124E Unrecognized intercept ID encountered (*xx*).

Explanation: Unrecognizable control information (*xx*) was passed from an intercept point to the intercept handling routine.

User response: Contact IBM Software Support.

GLO3126E VADD1260 non-zero return from SVC99, R15=*rrrrrrrr* -*ssss*.

Explanation: An internal error (*rrrrrrrr*) occurred during processing of the VOL_ADD function for a VSAM data set on behalf of subsystem *ssss*.

User response: Contact IBM Software Support.

GLO3127E Error occurred during SWAREQ processing for SIOT, rc=*rrrrrrrr* -*ssss*.

Explanation: During interception of a VSAM allocation request, an internal error (*rrrrrrrr*) was encountered during the attempt to access a system control block.

User response: Contact IBM Software Support.

GLO3128I *ttttt* successfully extended to volume *vvvvvv* -*ssss*, Total VOLCNT=*ccc*

Explanation: *ttttt*, identified by the message GLO3123I, was successfully extended to volume serial number *vvvvvv* on behalf of subsystem *ssss* in response to the VOL_ADD function.

Where *ttttt* is one of the following:

- DATA
- INDEX
- AIX-D (an AIX for the data component was processed)
- AIX-I (an AIX for the index component was processed)
- BAS-D
- BAS-I (which indicates that the VOL_ADD was performed for the base cluster data or index component due to an add through an UPDATE path)

Total VOLCNT is the total volume count.

User response: No action is required.

GLO3131E Volume not added due to volume add limit reached -*ssss*.

Explanation: A volume addition was not performed for the VSAM data set because the operating system limit of 59 volumes or the volume limit that was specified by the VOL_ADD_LIM keyword, was reached. Message GLO3123I identifies the data set.

User response: Increase the size of each extent or raise the limit that is specified by the VOL_ADD_LIM keyword.

GLO3135E SVOL2240 non-zero return from VAE, R15=*rrrrrrrr* -*ssss*.

Explanation: Advanced Allocation Management encountered an internal error (*rrrrrrrr*) while it was performing the VOL_ADD function for a VSAM data set on behalf of subsystem *ssss*.

User response: Contact IBM Software Support.

GLO3136E Non-zero return from VAE, R15=*rrrrrrrr*, R0=*ssssssss -ssss*.

Explanation: Advanced Allocation Management encountered an internal error that is identified by the return code *rrrrrrr* and the reason code *ssssssss* while it was performing the VOL_ADD function for a VSAM data set on behalf of subsystem *ssss*.

User response: Contact IBM Software Support.

GLO3137A Specify volser for *jobname,stepname,pstepname,ddname* or reply CANCEL -*ssss*.

Explanation: In support of the VOL_ADD_PROMPT keyword for a VSAM data set, specify a volume serial number to use to extend the data set that is associated with the job *jobname*, the step *stepname*, the procedure step name *procstepname*, and the ddname *ddname* for subsystem *ssss*.

User response: Specify a six character volume serial number to use to extend the data set, or reply with the **CANCEL** command to fail the allocation.

GLO3138I Operator responded to volume request with *vvvvvvv* -*ssss*.

Explanation: In response to message GLO3137A that is associated with the VOL_ADD_PROMPT function for a VSAM data set, the operator responded with volume serial number *vvvvvvv* for subsystem *ssss*.

User response: No action is required.

GLO3139E Volume add not performed - unable to locate suitable candidate volume -*ssss*.

Explanation: After searching all of the specified volumes to be used for locating a new volume for a VSAM data set, no volumes with adequate space were found to extend the data set to another volume. *ssss* identifies the Advanced Allocation Management subsystem ID. Message GLO3123I identifies the data set.

User response: Add more volumes to be searched, free up space on the existing volumes, or include the VOL_ADD_PROMPT keyword and supply a volume with adequate space.

GLO3141E Unexpected error *rrrrrrrr* during volume add -*ssss*.

Explanation: An internal error (*rrrrrrrr*) occurred during processing of the VOL_ADD function for a VSAM data set on behalf of subsystem *ssss*.

User response: Contact IBM Software Support.

GLO3144I Setting initial allocation on new volume to *nnnnn uuuu* -*sssss*.

Explanation: In response to successful completion of the PRIM_REDUCE function on a secondary volume for a VSAM data set (*vvvvvv*) for subsystem *ssss*, the initial allocation was set to *nnnnn*. *uuuu* indicates CYL S or TRKS. Message GLO3123I identifies the data set.

User response: No action is required.

GLO3145I Volume add not performed due to multiple opens against data set.

Explanation: A VOL_ADD operation was requested, but was not performed because more than one ACB is open concurrently against the VSAM data set and `BYPASS_CHECK=MULTOPEN` was not specified. Performing a VOL_ADD might compromise data set integrity or cause unpredictable results.

User response: If you are certain that the executing program can properly handle the multiple concurrent opens against the data set, specify `BYPASS_CHECK=MULTOPEN` to bypass the integrity check. For more information, see "Bypass integrity check processing (BYPASS_CHECK)" on page 65.

GLO3146E AVS not performed - unable to locate suitable candidate volume(s)-*xxxxx* -*ssys*.

Explanation: While Advanced Allocation Management was performing AVS processing for a non-SMS VSAM cluster component, it was unable to locate candidate volumes to fulfill the requested number of volumes for the data set based on the chosen AVS algorithm. The volume list for the component is emptied, which results in an IDCAMS DEFINE failure to prevent the allocation of the data set. *xxxxx* is the VSAM cluster component type (DATA or INDEX), and *ssys* is the Advanced Allocation Management subsystem identifier.

User response: Verify the following items:

- A volume group, unit name list, or volser list was specified in the rule definition for the allocation
- Specified volume groups are defined in a volume group definition
- Enough space exists in the specified volumes, based on the chosen AVS algorithm

GLO3147I Volume add not performed due to multiple DDs assigned to data set.

Explanation: A VOL_ADD operation was requested, but was not performed because more than one DD is allocated to the VSAM data set and `BYPASS_CHECK=MULTDD` was not specified. Performing a VOL_ADD might compromise data set integrity or cause unpredictable results.

User response: If you are certain that the executing program can properly handle multiple DDs allocated to

the data set, specify BYPASS_CHECK=MULTDD to bypass the integrity check. For more information, see "Bypass integrity check processing (BYPASS_CHECK)" on page 65.

GLO3148I AVS has successfully selected all volumes-*xxxxx* -*ssys*.

Explanation: AVS located all of the requested volumes for the non-SMS VSAM cluster component that is identified by the variable *xxxxx* as DATA or INDEX as requested using the specified AVS algorithm. Message GLO3156I displays the selected volumes.

User response: No action is required.

GLO3149E AVS unable to fulfill volume count request due to insufficient volumes-*xxxxx* -*ssys*.

Explanation: AVS was unable to locate the requested number of volumes for a non-SMS VSAM cluster component (*xxxxx*) as DATA or INDEX as requested using the specified AVS algorithm. The volume list for the component is truncated to the number of volumes that were located. *ssys* is the Advanced Allocation Management subsystem identifier. Message GLO3156I displays the volumes that were selected.

User response: Verify the following items:

- The volume group, unit name list, or volser list was specified in the rule definition for the allocation
- The specified volume groups are defined in a volume group definition
- Enough space exists in the volumes that were specified, based on the chosen AVS algorithm

GLO3156I Volser(s): *vvvvvv vvvvvvv vvvvvvv vvvvvvv vvvvvvv vvvvvvv vvvvvvv* **-***ssys*

Explanation: This message lists a maximum of seven volume serial numbers as chosen by AVS processing for a non-SMS VSAM cluster component. If more than seven volumes were allocated, multiple GLO3156I messages are displayed.

User response: No action is required.

GLO3160I Multiple open check bypassed for volume add.

Explanation: Multiple concurrent OPENs were found during a VOL_ADD operation for a VSAM data set. BYPASS_CHECK=MULTOPEN was specified. The VOL_ADD is attempted.

User response: No action is required.

GLO3161I Multiple DD check bypassed for volume add.

Explanation: During a VOL_ADD operation for a VSAM data set, multiple DDs assigned to the data set were found. BYPASS_CHECK=MULTDD was specified. The VOL_ADD is attempted.

User response: No action is required.

GLO3186I Maximum extents of 255 per cluster reached -*ssss*.

Explanation: The maximum allowable number of extents was reached for the VSAM cluster and no further extends can be performed.

User response: No action is required.

GLO3187I Maximum extents of 123 per volume reached -*ssss*.

Explanation: The maximum allowable number of extents for a VSAM component was reached and no further extends can be performed.

User response: No action is required.

GLO3188I Matching RULEDEF line number = *nnnnn* - *ssss*.

Explanation: In response to the MSG_LEVEL=RULEDEF option or the specification of the AOMATCH DD statement, Advanced Allocation Management displays this message upon a rule definition match during various points during VSAM processing. *xxxxxxx* is the text "INCLUDE" or "EXCLUDE". INCLUDE indicates a match on an INCLUDE rule definition. EXCLUDE indicates a match on an EXCLUDE rule definition. *nnnnn* is the rule definition line number and *ssss* is the Advanced Allocation Management subsystem identifier.

User response: No action is required.

GLO3188S Matching *xxxxxxx* RULEDEF line number = *nnnnn* -*ssss*.

Explanation: In response to the MSG_LEVEL=RULEDEF option or the specification of the AOMATCH DD statement, Advanced Allocation Management displays this message upon a rule definition match during various points during VSAM processing. *xxxxxxx* is the text "INCLUDE" or "EXCLUDE". INCLUDE indicates a match on an INCLUDE rule definition. EXCLUDE indicates a match on an EXCLUDE rule definition. *nnnnn* is the rule definition line number and *ssss* is the Advanced Allocation Management subsystem identifier.

User response: No action is required.

GLO3190I Entity processed by RULEDEF_TEST mode; RULEDEF line number *nnnnn* **-***ssss***.**

Explanation: The data set that is being processed matches a rule definition statement that designates simulate mode and identifies the line number of the matching statement in the active RULEDEFS definition. No recovery action is performed for the data set. This message can be displayed more than once during the life of an allocation through unallocation for a data set. Message GLO2123I identifies the data set that is involved. This message applies to VSAM data sets only.

User response: No action is required.

GLO3194I Set processing complete; *keyword***=***newvalue* **(old value=***oldvalue***).**

Explanation: This message is issued when Advanced Allocation Management successfully processes a SET request. *keyword* is the keyword that is being processed. *newvalue* is the new value for the keyword and *oldvalue* is the previous value of the keyword.

Response: No action is required.

GLO3195E Failure during consistency check for set processing; reason code *xxx***.**

Explanation: During SET processing, a consistency check failed with one of the following reason codes *xx*:

01 One or more of the following parameters were implicitly or explicitly set to dataclas-override (DCO), but not all of them were set to DCO:

- SPACEUNITS
- PQTY
- SQTY

User response: Change the SET values for SPACEUNITS, PQTY, and (or) SQTY, so that DCO is used for all of the parameters (or none of them).

02 SPACEUNITS, PQTY, and SQTY were implicitly or explicitly set to dataclas-override (DCO), but AVGVALUE, PQTY, and (or) SQTY was not defined in the dataclas.

User response: Add values to the dataclas for AVGVALUE, PQTY, and (or) SQTY, or change the SET statements so that DCO is not used for SPACEUNITS, PQTY, or SQTY.

03 One or more of the following parameters were set to $NULL, but not all of them were set to $NULL:

- SPACEUNITS
- PQTY
- SQTY

User response: Change the SET values for SPACEUNITS, PQTY, and (or) SQTY, so that $NULL is used for all of them (or none of them).

04 SPACEUNITS is being implicitly or explicitly changed, but no corresponding PQTY change was specified.

User response: Add a SET statement for PQTY.

05 SPACEUNITS is being implicitly or explicitly changed, but no corresponding SQTY change was specified.

User response: Add a SET statement for SQTY.

06 SPACEUNITS is being implicitly or explicitly set to AVGRECU, AVGRECK, or AVGRECM, but the record-size was not specified.

User response: Specify the record-size subparameter on the SET_SPACEUNITS statement.

Message GLO3123I accompanies this message. For more information, see "Enhanced Allocation Management functions" on page 126.

GLO3199E Error occurred during SWAREQ processing for JFCBX, rc=*rrrrrrrr* **-***ssss***.**

Explanation: During SELECTION processing, an internal error (*rrrrrrrr*) was encountered during the attempt to access a system control block.

User response: Contact IBM Software Support.

GLO3200E Unable to chain control block for field list - *xxxxxxxx* **-***ssss***.**

Explanation: During SET processing for an IDCAMS DEFINE, an internal error occurred while processing a catalog field list for allocation variable *xxxxxxxx*.

User response: Contact IBM Software Support.

GLO3201E Unable to acquire space to build field list - *xxxxxxxx* **-***ssss***.**

Explanation: During SET processing for an IDCAMS DEFINE, an internal error occurred while processing a catalog field list for allocation variable *xxxxxxxx*.

User response: Contact IBM Software Support.

GLO3202E Unable to chain control block for field list -*xxxxxxx* **-***ssss***.**

Explanation: During SET processing for an IDCAMS DEFINE, an internal error occurred while processing a catalog field list for allocation variable *xxxxxxxx*.

User response: Contact IBM Software Support.

GLO3220W Unsupported EDB encountered.

Explanation: A 64 bit EDB was encountered.

Processing is bypassed for the current data set.

User response: Contact IBM Software Support.

Syntax-check messages

These messages are related to issues with the Advanced Allocation Management syntax-checking feature.

GLO4001W End-of-data encountered *ssid*.

Explanation: An internal error occurred during rule or volume group definition activation or syntax checking. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Contact IBM Software Support.

GLO4002E String address is zero *ssid*.

Explanation: An internal error occurred during rule or volume group definition activation or syntax checking. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Contact IBM Software Support.

GLO4003E Processing mode not specified *ssid*.

Explanation: An internal error occurred during rule or volume group definition activation or syntax checking. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Contact IBM Software Support.

GLO4004E Read routine address missing *ssid*.

Explanation: An internal error occurred during rule or volume group definition activation or syntax checking. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Contact IBM Software Support.

GLO4006E Record area address missing *ssid*.

Explanation: An internal error occurred during rule or volume group definition activation or syntax checking. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Contact IBM Software Support.

GLO4007E Output area address missing *ssid*.

Explanation: An internal error occurred during rule or volume group definition activation or syntax checking. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Contact IBM Software Support.

GLO4008E Parse table address missing *ssid*.

Explanation: An internal error occurred during rule or volume group definition activation or syntax checking. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Contact IBM Software Support.

GLO4009E Internal error - RKDFTYPE invalid *ssid*.

Explanation: An internal error occurred during rule or volume group definition activation or syntax checking. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Contact IBM Software Support.

GLO4010E Keyword should not have parameters *keyword ssid*.

Explanation: The keyword (*keyword*) was specified with parameters. The keyword does not require parameters. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Remove the parameters from the keyword and then use the Advanced Allocation Management syntax-check option to verify the syntax. For more information, see "Syntax-checking control data set members" on page 277.

GLO4011E Illegal placement of equal sign *keyword ssid*.

Explanation: The specified keyword (*keyword*) has an equal sign that is specified in an incorrect location. This error is usually caused by one or more spaces before, or after, the equal sign.

User response: Correct the syntax and then use the Advanced Allocation Management syntax-check option to verify the syntax. For more information, see "Syntax-checking control data set members" on page 277.

GLO4012E Excessive number of keyword parms specified *keyword ssid*.

Explanation: More parameters were specified for the (*keyword*) than are allowed. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Use the Advanced Allocation

Management syntax-check option to verify the syntax for the appropriate number of parameters and then remove or add parameters as required. For more information, see "Syntax-checking control data set members" on page 277.

GLO4013E Unbalanced parentheses *keyword ssid*.

Explanation: The specified keyword (*keyword*) does not contain an equal number of matching left and right parentheses. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Use the Advanced Allocation Management syntax-check option to verify the syntax and to determine if you need to add a right or left parentheses to enclose the operands. For more information, see "Syntax-checking control data set members" on page 277.

GLO4014E Misplaced comma *keyword ssid*.

Explanation: The keyword (*keyword*) has a comma specified in an incorrect location. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Correct the syntax and use the Advanced Allocation Management syntax-check option to verify the syntax. For more information, see "Syntax-checking control data set members" on page 277.

GLO4015E Misplaced equal sign *keyword ssid*.

Explanation: The keyword (*keyword*) has an equal sign in an incorrect location. This message is issued when two equal signs are adjacent to each other. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Correct the syntax and then use the Advanced Allocation Management syntax-check option to verify the syntax. For more information, see "Syntax-checking control data set members" on page 277.

GLO4016E Keyword length exceeds 256 byte internal limit *ssid*.

Explanation: An internal error occurred during rule or volume group definition activation or syntax checking. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Contact IBM Software Support.

GLO4017E Keyword value not valid *keyword ssid*.

Explanation: The keyword (*keyword*) specified an operand that is not one of the allowable values. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Use the Advanced Allocation Management syntax-check option to verify the syntax. For more information, see "Syntax-checking control data set members" on page 277.

GLO4018E Keyword value not alphabetic *keyword ssid*.

Explanation: The keyword (*keyword*) value is not entirely alphabetic as required. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Use the Advanced Allocation Management syntax-check option to verify the syntax. For more information, see "Syntax-checking control data set members" on page 277.

GLO4019E Keyword value not numeric *keyword ssid*.

Explanation: The keyword (*keyword*) value is not numeric as required. The *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Correct the syntax and use the Advanced Allocation Management syntax-check option to verify the syntax. For more information, see "Syntax-checking control data set members" on page 277.

GLO4020E Keyword value does not begin with alphabetic *keyword ssid*.

Explanation: The keyword (*keyword*) value does not begin with an alphabetic character as required. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Correct the syntax and use the Advanced Allocation Management syntax-check option to verify the syntax. For more information, see "Syntax-checking control data set members" on page 277.

GLO4021E Keyword value exceeds maximum allowed length *keyword ssid*.

Explanation: The length of the value for the keyword (*keyword*) is longer than the maximum length allowed. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Correct the syntax and use the Advanced Allocation Management syntax-check option to verify the syntax. For more information, see "Syntax-checking control data set members" on page 277.

GLO4022E Keyword value does not meet minimum length *keyword ssid***.**

Explanation: The length of the value for the keyword (*keyword*) is shorter than the minimum length allowed. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Correct the syntax and use the Advanced Allocation Management syntax-check option to verify the syntax. For more information, see "Syntax-checking control data set members" on page 277.

GLO4023E Unable to expand variable area *keyword ssid***.**

Explanation: While processing the keyword (*keyword*) an internal storage-related error occurred. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Increase the above-the-line region size for the Advanced Allocation Management started task. If increasing the above-the-line region size does not resolve the error, contact IBM Software Support.

GLO4024E Numeric value exceeds internal limit of 999,999,999,999,999 *keyword ssid***.**

Explanation: The value for the keyword (*keyword*) is greater than the maximum allowed numeric value of 999,999,999,999,999. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Correct the syntax and use the Advanced Allocation Management syntax-check option to verify the syntax. For more information, see "Syntax-checking control data set members" on page 277.

GLO4025E Numeric value is below minimum allowed *keyword ssid***.**

Explanation: The value for the keyword (*keyword*) is less than the minimum value allowed. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Correct the syntax and use the Advanced Allocation Management syntax-check option to verify the syntax. For more information, see "Syntax-checking control data set members" on page 277.

GLO4026E Numeric value exceeds maximum allowed *keyword ssid***.**

Explanation: The value for the keyword (*keyword*) is greater than the maximum value allowed. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Correct the syntax and use the Advanced Allocation Management syntax-check option to verify the syntax. For more information, see "Syntax-checking control data set members" on page 277.

GLO4027E Missing left parenthesis *keyword ssid***.**

Explanation: More than one parameter value was specified for the keyword (*keyword*). The required left parenthesis was not found. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Correct the syntax and use the Advanced Allocation Management syntax-check option to verify the syntax. For more information, see "Syntax-checking control data set members" on page 277.

GLO4028E Misplaced left parenthesis *keyword ssid***.**

Explanation: A left parenthesis was specified in an inappropriate location for the keyword (*keyword*). *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Correct the syntax and use the Advanced Allocation Management syntax-check option to verify the syntax. For more information, see "Syntax-checking control data set members" on page 277.

GLO4029E Missing address of variable area *ssid***.**

Explanation: An internal error occurred during rule or volume group definition activation or syntax checking. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Contact IBM Software Support.

GLO4030E Invalid keyword *keyword ssid***.**

Explanation: The keyword (*keyword*) is not a recognized keyword. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Correct the syntax and use the Advanced Allocation Management syntax-check option to verify the syntax. For more information, see "Syntax-checking control data set members" on page 277.

GLO4031E Secondary keyword not preceded by primary keyword *keyword ssid***.**

Explanation: The keyword (*keyword*) was found without a preceding DEFAULTS, INCLUDE, or EXCLUDE keyword. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Correct the syntax. Use the Advanced Allocation Management syntax-check option to verify

the syntax. For more information, see "Syntax-checking control data set members" on page 277.

GLO4032E Unexpected end of data *keyword ssid***.**

Explanation: An internal error occurred during rule or volume group definition activation or syntax checking while processing the keyword (*keyword*). *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Contact IBM Software Support.

GLO4033E Expected an equal sign *keyword ssid***.**

Explanation: An internal error occurred during rule or volume group definition activation or syntax checking while processing the keyword (*keyword*). *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Contact IBM Software Support.

GLO4036E Keyword value does not begin with ampersand - *variablename – ssid***.**

Explanation: This message is displayed in the JESSYSLOG when a VARDEFS member variable definition VARIABLE_NAME value does not begin with the required ampersand. The rule or volume group definitions are not activated.

User response: Review the JESSYSLOG and correct the errors. For more information, see "Creating user-defined variable definitions" on page 271.

GLO4037E Invalid imbedded variable in value list- *variablename – ssid***.**

Explanation: This message is displayed in the JESSYSLOG when a VARDEFS member variables name VALUE contains an embedded variable. User-defined variables cannot be embedded in other variables. The rule or volume group definitions are not activated.

User response: Review the JESSYSLOG and correct the errors. For more information, see "Creating user-defined variable definitions" on page 271.

GLO4038E Substitution variable not defined - *varname –ssid***.**

Explanation: This message is displayed in the JESSYSLOG when a rule or volume group definition contains a variable that is not defined in the current VARDEFS member.

User response: Review the JESSYSLOG and correct the errors in the VARDEFS member. For more information, see "Creating user-defined variable definitions" on page 271.

GLO4039E Variable not allowed for keyword- *keywordname – ssid***.**

Explanation: This message is displayed in the JESSYSLOG when a rule or volume group definition keyword contains a variable, but the keyword does not support variables. The rule or volume group definitions are not activated.

User response: Review the JESSYSLOG and correct the errors.

GLO4040E Keyword value null - *keyword - ssid***.**

Explanation: This message is displayed in JESSYSLOG during activation of rule or volume group definitions when a null keyword is found in the current VARDEFS member. *keyword* is VARIABLE_NAME or VALUE. The activation of the rule or volume group definitions fails.

User response: Review the JESSYSLOG and correct the VARDEFS error and then activate the associated RULEDEFS or VGRPDEFS member that is affected. For more information, see "Creating user-defined variable definitions" on page 271.

GLO4100F Fatal error encountered - parsing terminated *ssid***.**

Explanation: An internal error occurred during rule or volume group definition activation or syntax checking. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Contact IBM Software Support.

GLO4101F Unable to GETMAIN CDSB - parsing terminated *ssid***.**

Explanation: An internal error occurred. Advanced Allocation Management was unable to obtain enough storage in the Extended CSA for a required control block. *ssid* indicates the subsystem ID that is assigned to Advanced Allocation Management.

User response: Check for available space in the ECSA and increase the region size above-the-line for the Advanced Allocation Management started task. If the problem persists, contact IBM Software Support.

GLO41021E Control data RECFM error - processing terminated *ssid***.**

Explanation: The CONTROL DD must point to a partitioned data set with fixed format records that have a record length of 80 bytes. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Review the contents of the started task JCL to ensure that it includes a DD statement with a DDNAME of CONTROL that points to a valid partitioned data set that has fixed format records with a record length of 80 bytes. For more information, see

"Configuring the started task JCL" on page 28.

GLO4102F Unable to GETMAIN GCDB - parsing terminated *ssid***.**

Explanation: An internal error occurred. Advanced Allocation Management was unable to obtain enough storage in the Extended CSA for a required control block. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Check the available space in the ECSA and increase the region size above-the-line for the Advanced Allocation Management started task. If the problem persists, contact IBM Software Support.

GLO4103W No data processed - parsing terminated *ssid***.**

Explanation: The control data set member did not contain valid keywords. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Ensure that the control data set member contains valid syntax. Use the Advanced Allocation Management syntax-check option to verify the syntax. For more information, see "Syntax-checking control data set members" on page 277.

GLO4104E Errors encountered - rule(s) not activated *ssid***.**

Explanation: Errors were encountered while processing the control data set member. The rule definitions were not activated. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: See the JES SYSMSG list data set for a detailed list of the errors.

GLO4105E VARDEF errors encountered - RULE(S)/VGRPS(S) not activated - *ssid***.**

Explanation: This message is displayed in the JESSYSLOG when a VARDEFS variable definition member contains errors. It indicates that the rule definitions or volume group definitions were not activated and is displayed at the end of the GLO4199I messages that detail the contents of the VARDEFS member.

User response: Review the JESSYSLOG, correct the variable definition errors, and then activate the appropriate RULEDEFS or VGRPDEFS member that is affected. For information about defining variables, see "Creating user-defined variable definitions" on page 271.

GLO4105I Rule definitions successfully activated *ssid***.**

Explanation: No errors were encountered during activation and the rule definitions were successfully activated. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: No action is required.

GLO4106E Defaults definition missing from member – processing terminated *ssid***.**

Explanation: The DEFAULTS keyword was not found before an INCLUDE or EXCLUDE keyword. The DEFAULTS keyword and its operands (if any) must precede any INCLUDE or EXCLUDE keywords. The *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Make sure that the DEFAULTS keyword precedes the INCLUDE or EXCLUDE keywords. Use the Advanced Allocation Management syntax-check option to verify the syntax. For more information, see "Syntax-checking control data set members" on page 277 and "DEFAULTS" on page 47.

GLO4107E Excessive number of defaults definitions - processing terminated *ssid***.**

Explanation: Only one DEFAULTS keyword is permitted in the rules definition control data set member. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Remove the unneeded DEFAULTS keywords and then use the Advanced Allocation Management syntax-check option to verify the syntax. For more information, see "Syntax-checking control data set members" on page 277.

GLO4108E Errors encountered - VGRP(S) not activated *ssid***.**

Explanation: Errors were encountered while processing the control data set member. The volume group definitions were not activated. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: For a detailed list of the errors that were encountered, see the JES SYSMSG list data set.

GLO4109I VGRP definitions successfully activated *ssid*

Explanation: No errors were encountered during activation and the volume group definitions were successfully activated. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: No action is required.

GLO4110E Unable to GETMAIN < 16M workarea – processing terminated *ssid***.**

Explanation: An internal error occurred during rule or volume group definition activation or syntax checking. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Increase the region size below-the-line for the Advanced Allocation Management started task. If the problem persists, contact IBM Software Support.

GLO4110I Var definitions in effect for following RULE/VGRP activation –*ssid***.**

Explanation: This message is issued in the JESSYSLOG when rule or volume group definitions are activated. This message is issued before the rule or volume group definitions are described. The contents of the VARDEFS member are displayed in the message GLO4199I before this message is issued.

User response: No action is required.

GLO4111E Unable to establish estae - processing terminated *ssid***.**

Explanation: An internal error occurred. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Contact IBM Software Support.

GLO4112E Invalid parm block (type) - processing terminated *ssid***.**

Explanation: An internal error occurred during rule or volume group definition activation or syntax checking. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Contact IBM Software Support.

GLO4113E Invalid parm block (SRCE) - processing terminated *ssid***.**

Explanation: An internal error occurred. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Contact IBM Software Support.

GLO4114E Invalid parm block (DEFN) - processing terminated *ssid***.**

Explanation: An internal error occurred during rule or volume group definition activation or syntax checking. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Contact IBM Software Support.

GLO4115E Unable to locate CONTROL DD - processing terminated *ssid***.**

Explanation: A DD statement with a DDNAME of CONTROL must be present within the Advanced Allocation Management started task. The DD statement must point to a partitioned data set with fixed-format records that have a record length of 80 bytes. The control data set contains members that provide parameters that control Advanced Allocation Management processing. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Review the contents of the started task JCL to ensure that it includes a DD statement with a DDNAME of CONTROL that points to a partitioned data set with fixed-format records that have a record length of 80 bytes. For more information, see "Configuring the started task JCL" on page 28.

GLO4116E SWAREQ failure - processing terminated *ssid***.**

Explanation: An internal error occurred during rule or volume group definition activation or syntax checking. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Contact IBM Software Support.

GLO4117E DYNALLOC failure - processing terminated *ssid***.**

Explanation: An internal error occurred during rule definition or volume group definition activation or syntax checking. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Contact IBM Software Support.

GLO4118E Control data set must be DSORG of PO – Processing terminated *ssid***.**

Explanation: The CONTROL DD must point to a partitioned data set with fixed-format records that have a record length of 80 bytes. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Review the contents of the Advanced Allocation Management started task JCL to ensure that it includes a DD statement with a DDNAME of CONTROL that points to a partitioned data set with fixed-format records with a record length of 80 bytes. For more information, see "Configuring the started task JCL" on page 28.

GLO4119E Control data OPEN error - Processing terminated *ssid***.**

Explanation: An error was encountered during OPEN processing for the data set referenced by the CONTROL DD statement. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Review the contents of the started task JCL to ensure that it includes a DD statement with a DDNAME of CONTROL that points to a partitioned data set. Ensure that the data set is a valid partitioned data set with fixed-format records with a record length of 80 bytes. For more information, see “Configuring the started task JCL” on page 28.

GLO4120E Control data LRECL ERROR - Processing terminated *ssid*.

Explanation: The CONTROL DD must point to a partitioned data set with fixed-format records with a record length of 80 bytes. The *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Review the contents of the started task JCL to ensure that it includes a DD statement with a DDNAME of CONTROL that points to a valid partitioned data set with fixed-format records with a record length of 80 bytes. For more information, see “Configuring the started task JCL” on page 28.

GLO4122E Unable to locate control member – Processing terminated *ssid*.

Explanation: The member name that is specified on, or defaulted to, in an **ACTIVATE RULEDEFS** or **ACTIVATE VGRPDEFS** command, was not found. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Ensure that the specified member exists in the data set that is pointed to by the CONTROL DD statement in the Advanced Allocation Management started task procedure.

GLO4123E Unable to GETMAIN buffer area - Processing terminated *ssid*.

Explanation: An internal error occurred. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Increase the region size above-the-line for the Advanced Allocation Management started task. If the problem persists, contact IBM Software Support.

GLO4124E Invalid parm block (STRA) - Processing terminated *ssid*.

Explanation: An internal error occurred during rule or volume group definition activation or syntax checking. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Contact IBM Software Support.

GLO4125E Invalid parm block (STRD) - Processing terminated *ssid*.

Explanation: An internal error occurred during rule or volume group definition activation or syntax checking. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Contact IBM Software Support.

GLO4126E Unable to GETMAIN variable area – Processing terminated *ssid*.

Explanation: An internal error occurred during rule or volume group definition activation or syntax checking. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Increase the region size above-the-line for the Advanced Allocation Management started task. If the problem persists, contact IBM Software Support.

GLO4127E Read routine address missing - Processing terminated.

Explanation: An internal error in processing occurred during rule or volume group definition activation or syntax checking.

User response: Contact IBM Software Support.

GLO4128E Write routine address missing - Processing terminated.

Explanation: An internal error in processing occurred during rule or volume group definition activation or syntax checking.

User response: Contact IBM Software Support.

GLO4129E First record must be comment card.

Explanation: The first logical record in a rule or volume group definition was not a comment card.

User response: Add a comment card as the first record of the rule or volume group definitions. The comment must contain the characters "RULE DEFINITION" or "VGRP DEFINITION" as appropriate. For more information, see “Creating rule definitions” on page 45 and “Creating volume group definitions” on page 267.

GLO4130I No syntax errors found.

Explanation: This message is issued at the end of syntax check processing that was invoked using option **S** on the Tivoli Advanced Allocation Management Control Member Management panel. The message indicates that syntax checking was successful and no errors were found.

User response: No action is required.

GLO4131E Syntax errors encountered.

Explanation: This message is issued at the end of the syntax check processing that was invoked using option **S** on the Tivoli Advanced Allocation Management Control Member Management panel. It indicates that errors were found while checking the syntax for the associated rule or volume group definitions.

User response: Check the listing for other error messages that provide information about the errors, and then correct the errors.

GLO4132E Unable to GETMAIN variable area 2 - Processing terminated *ssid*.

Explanation: An internal error occurred during rule group definition activation or syntax checking. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: Increase the region size above-the-line for the Advanced Allocation Management started task. If the problem persists, contact IBM Software Support.

GLO4199I *linenum linedata – ssid*

Explanation: This message displays each record that was read from the control data set member. *linenum* is the record number and *linedata* is the record image. *ssid* is the subsystem ID that is assigned to Advanced Allocation Management.

User response: No action is required.

Event recording and reporting messages

These messages are related to issues with the Advanced Allocation Management event recording and reporting utility.

GLO7000E Insufficient above-the-line virtual storage available.

Explanation: During event reporting processing, Advanced Allocation Management was not able to obtain the required storage above the 16 MB line.

User response: Increase the region size for the report utility and run the utility again. If the problem persists, contact IBM Software Support.

GLO7001E Insufficient below-the-line virtual storage available.

Explanation: During event reporting processing, Advanced Allocation Management was unable to obtain the required storage below the 16 MB line.

User response: Increase the region size for the report utility and run the utility again. If the problem persists, contact IBM Software Support.

GLO7002E OPEN failed for CNTLOUT file.

Explanation: OPEN processing was unsuccessful for the CNTLOUT file.

User response: Ensure that the CNTLOUT DD statement is present in the JCL and that it points to a valid data set. For more information, see "Configuring batch reporting" on page 293. Look in the job log for data management (IEC*xxx*) messages that contain more information about the error. If the problem persists, contact IBM Software Support.

GLO7003W Tape mount requires APF-authorization, tape input skipped.

Explanation: One or more specified SMF data sets are on tape. To request tapes to be mounted, you must run the GLOLOGRP utility with APF authorization. If you receive this message when you run the utility through the ISPF interface **View Event History** option (option 4), maintaining this authorization is not possible due to the nature of TSO environments. Therefore, SMF data sets that are on tape cannot be processed using the ISPF interface.

User response: If you receive this message when you run the GLOLOGRP utility in batch, ensure that the load library that you are running the utility from is APF-authorized. When you run the utility using the ISPF interface, first copy the SMF data sets that reside on tape to disk, or run the utility in batch instead.

GLO7010E OPEN failed for CNTLIN file.

Explanation: OPEN processing was unsuccessful for the CNTLIN file.

User response: Ensure that the CNTLIN DD statement is present in the JCL and that it points to a valid data set. For more information, see "Configuring batch reporting" on page 293. Look in the job log for data management (IEC*xxx*) messages that contain more information about the error. If the problem persists, contact IBM Software Support.

GLO7011E Maximum of one SMFIN statement has been exceeded.

Explanation: More than one SMFIN control statement was specified. One SMFIN control statement is permitted.

User response: Specify one SMFIN control statement.

GLO7012W One or more SMF records were encountered of an unknown version.

Explanation: Advanced Allocation Management encountered one or more SMF records whose record type matched the type that was specified, but it did not recognize the version value within it.

User response: Verify that the version of the report utility is the same as the version of the product that recorded the records to SMF. If problems persist, contact IBM Software Support.

GLO7013E Error(s) encountered during control statement processing.

Explanation: Errors were encountered while processing the report utility control statements.

User response: Review the control statements that are provided and verify that the syntax is correct. For more information, see "Configuring batch reporting" on page 293.

GLO7014E Error occurred during allocation of temporary work file.

Explanation: An error occurred during the allocation of the temporary work file that is used during event reporting processing. Message GLO7049E provides diagnostic information about the error.

User response: Contact IBM Software Support.

GLO7015E OPEN failed for TEMPFILE for output processing.

Explanation: OPEN processing was unsuccessful for the temporary work file that is used during event reporting output processing.

User response: Look in the job log for data management (IEC*xxx*) messages that contain more information about the error. If the problem persists, contact IBM Software Support.

GLO7016E Error occurred during allocation of SMF data set.

Explanation: An error occurred during the allocation of the SMF data set that is specified using the SMFIN control statement. Messages GLO7018E and GLO7049E accompany this message and indicate the SMF data set name in error and provide diagnostic information about the error.

User response: Verify that the specified data set name exists and is cataloged.

GLO7017E OPEN failed for SMF data set -

Explanation: OPEN processing was unsuccessful for the specified SMF data set. Message GLO7018E indicates the SMF data set name in error.

User response: Look in the job log for data management (IEC*xxx*) messages that contain more information about the error. If the problem persists, contact IBM Software Support.

GLO7018E *data set name*

Explanation: This message accompanies message GLO7016E or message GLO7017E and indicates the name of the SMF data set for which the error occurred.

User response: See the accompanying message for instructions.

GLO7019E No SMF records were found in any input file.

Explanation: No SMF records were found in the specified SMF data sets to be searched for Advanced Allocation Management event records.

User response: Ensure that the data sets that are specified by the SMFIN control statement contain valid SMF data.

GLO7020E No SMF records of the specified type were found in any input file.

Explanation: No SMF records of the type specified by the SMF_RECORD_ID parameter were found in the SMF data sets to be searched for Advanced Allocation Management event records.

User response: Ensure that the proper set-up steps for event recording were performed and that Advanced Allocation Management is active. Also ensure that the RULEDEFS are set up properly so that Advanced Allocation Management is actively preventing and recovering space-related conditions on your system. Finally, ensure that the value specified on the SMF_RECORD_ID control statement for the report utility matches the value that is specified on the SMF_RECORD_ID control statement in the OPTIONS member of the Advanced Allocation Management control data set.

GLO7021E No SMF records of the specified type passed filtering.

Explanation: No SMF records in the specified SMF data sets passed the filtering that is specified by the FILTER_SET control statements.

User response: Ensure that the filter criteria values were specified correctly and that the entities specified in the filter criteria had Advanced Allocation Management processing performed on their behalf. For information about filter criteria, see "Report filter criteria and filter sets" on page 288.

GLO7022E OPEN failed for HISTOUT DCB.

Explanation: OPEN processing was unsuccessful for the HISTOUT DCB.

User response: Ensure that the HISTOUT DD statement is present in the JCL and that it points to a valid data set. Look in the job log for data management (IEC*xxx*) messages that contain more information about the error. If the problem persists, contact IBM Software Support.

GLO7023E DETAIL=NO and SUMMARY=NO are not valid together.

Explanation: The report option combination DETAIL=NO AND SUMMARY=NO is not valid.

User response: Specify a valid combination of report options. For more information, see "Report option combinations and results" on page 289.

GLO7024E SUMMARY=YES and FORMAT=LIST are not valid together.

Explanation: The report option combination SUMMARY=YES AND FORMAT=LIST is not valid.

User response: Specify a valid combination of report options. For more information, see "Report options" on page 289.

GLO7025E Above control statement contains one or more errors.

Explanation: A control statement that was specified as input to the report utility contains an error.

User response: Review the specified control statement and verify that the keyword, value, and syntax are valid. For more information, see "Configuring batch reporting" on page 293.

GLO7026E No data set names were specified for SMFIN.

Explanation: A data set name was not specified for the SMFIN parameter in the report utility JCL. You must specify a minimum of one data set for event data reporting to occur.

User response: Specify one or more names of SMF data sets to be searched for Advanced Allocation Management event records. You can specify a maximum of 32 data sets. For more information, see "Configuring batch reporting" on page 293.

GLO7027E SMF_RECORD_ID is required, but was not specified.

Explanation: The Advanced Allocation Management report utility requires an SMF record ID to determine the SMF records to process.

User response: Add an SMF_RECORD_ID control statement to the report utility input and specify the value that Advanced Allocation Management was configured to use for event recording in the OPTIONS member of the Advanced Allocation Management control data set. For more information, see "Specifying subsystem options" on page 29.

GLO7028I Parenthesis unmatched before end-of-file on CNTLIN data set.

Explanation: During the processing of the CNTLIN data set, a required matching parenthesis was not found to be missing.

User response: Review the input control statements, add the required parenthesis to the appropriate control statement, and run the report utility again. For more information, see "Configuring batch reporting" on page 293.

GLO7030E FORMAT=LIST and DETAIL=NO are not valid together.

Explanation: The report option combination FORMAT=LIST AND DETAIL=NO is not valid.

User response: Specify a valid combination of report options. For more information, see "Report options" on page 289.

GLO7031E Maximum number of SMFIN files (32) exceeded.

Explanation: The number of SMF data sets that are specified through the SMFIN control statement is greater than the maximum number allowed.

User response: Modify the SMFIN control statement to specify no more than 32 SMF data sets and run the report utility again.

GLO7032E Maximum number of filter criteria for a filterset (32) exceeded.

Explanation: The number of filter criteria specified is greater than the maximum number of allowed filter criteria allowed for a single filterset.

User response: Modify the FILTER_SET control statement to specify no more than 32 filter criteria and run the report utility again.

GLO7033E Maximum number of filtersets (32) exceeded.

Explanation: The number of FILTER_SET control statements specified is greater than the maximum number allowed.

User response: Change the control statements to specify no more than 32 FILTER_SETs and run the report utility again.

GLO7034E Unable to obtain storage for FCCA.

Explanation: During event reporting processing, insufficient storage above the 16 MB line was available for an internal product control block.

User response: Increase the region size for the report utility and run the utility again. If the problem persists, contact IBM Software Support.

GLO7035E Error occurred in CONVTOD service.

Explanation: A non-zero return code was returned from the system time-of-day conversion routine (CONVTOD). The routine is required for the report utility to work properly.

User response: Contact IBM Software Support.

GLO7036E Error occurred in STCKCONV service.

Explanation: A non-zero return code was returned from the system store-clock conversion routine (STCKCONV). The routine is required for the report utility to work properly.

User response: Contact IBM Software Support.

GLO7037E Required delimiter "=" is missing in above control statement.

Explanation: During the processing of the control statements for the report utility, a control statement was encountered which requires an equal sign (=) but it was found to be missing.

User response: Add the required delimiter to the appropriate control statement and run the report utility again. For more information, see "Configuring batch reporting" on page 293.

GLO7038E Data set specified by SMFIN is not cataloged -

Explanation: One or more of the data sets that are specified by the SMFIN parameter are not cataloged. Message GLO7018E accompanies this message and indicates the SMF data set name that was not found in the catalog.

User response: Check the data sets that are specified by the SMFIN parameter and verify that all of them are cataloged.

GLO7039E OPEN failed for TEMPFILE file for input processing.

Explanation: OPEN processing was unsuccessful for the TEMPFILE file that is required for input processing.

User response: Look in the job log for data management (IEC*xxx*) messages that contain more information about the error. If the problem persists, contact IBM Software Support.

GLO7040E Unable to locate required ddname - SMFIN -

Explanation: During event reporting processing, Advanced Allocation Management could not locate the SMF input data set ddname SMFIN in the task input/output table (TIOT). Message GLO7018E accompanies this message and indicates the SMF data set name that is being processed at the time of the error. This is an internal error.

User response: Contact IBM Software Support.

GLO7041E Error occurred during SWAREQ JFCB retrieval for SMFIN, rc=*rrrrrrrr*,

Explanation: During event reporting processing, Advanced Allocation Management issued a SWAREQ request to retrieve the job file control block (JFCB) for the SMF input data set ddname SMFIN. The SWAREQ request was unsuccessful and returned return code *rrrrrrrr*. Message GLO7018E accompanies this message and indicates the SMF data set name that is being processed at the time of the error. This is an internal error.

User response: Contact IBM Software Support.

GLO7042E Invalid filter criteria specified in filterset-

Explanation: An incorrect filter criteria was encountered on a FILTER_SET control statement. Message GLO7043E accompanies this message and indicates the incorrect filter criteria.

User response: Review the specified filter criteria and verify that the keyword, value, and syntax are valid. For more information, see Chapter 10, "Event recording and reporting," on page 287. If the problem persists, contact IBM Software Support.

GLO7044E Invalid DSORG (VSAM) for data set specified by SMFIN -

Explanation: One or more data sets that are specified by the SMFIN parameter do not have a valid DSORG (PS). Message GLO7018E accompanies this message and indicates the SMF data set name that has the incorrect DSORG.

User response: The Advanced Allocation Management event reporting utility works only with SMF data that was previously unloaded from the system SMF (MAN) data sets. Ensure that all SMF data sets that are specified by the SMFIN parameter have a DSORG of PS.

GLO7049E DYNALLOC return code=*retcode*, error code=*errcode*, info code=*infocode*.

Explanation: During event reporting processing, a dynamic allocation request failed. This message accompanies a previous message and provides diagnostic information about the error. *retcode* is the value that is returned in register 15 by the dynamic allocation service. *errcode* and *infocode* are the error and info code values that are returned in fields S99ERROR and S99INFO in the dynamic allocation request block. These values are documented in the *MVS Programming: Authorized Assembler Services Guide*.

User response: The *retcode*, *errcode*, and *infocode* values can be found in the *MVS Programming: Authorized Assembler Services Guide*, which can help to determine the exact nature of the problem. Contact IBM Software Support, if needed.

GLO7050E Unable to load sort routine.

Explanation: During event reporting processing, the SORT_BY parameter was specified which caused Advanced Allocation Management to attempt to load a module with the name of SORT, but the load operation was unsuccessful.

User response: If the SORT module does not reside in the link list or LPALIB on your system, add a DD to the STEPLIB concatenation of the report utility step and specify the name of the loadlib in which SORT resides. If the SORT module resides in the link list or LPALIB on your system, or you add a DD to the STEPLIB concatenation of the report utility step and the problems persists, contact IBM Software Support.

GLO7051E Unable to locate required ddname - TEMPFILE

Explanation: The SORT_BY parameter was specified for event reporting processing. During an attempt to perform sort processing, Advanced Allocation Management was unable to locate its temporary processing DD (TEMPFILE) in the TIOT. This error is an internal error.

User response: Contact IBM Software Support.

GLO7052E Error occurred during SWAREQ JFCB retrieval for TEMPFILE, rc=*rrrrrrrr*.

Explanation: The SORT_BY parameter was specified for event reporting processing. During an attempt to perform sort processing, Advanced Allocation Management issued a SWAREQ request to retrieve the JFCB for its temporary processing DD (TEMPFILE). The SWAREQ request was unsuccessful and returned the return code *rrrrrrrr*. This error is an internal error.

User response: Contact IBM Software Support.

GLO7053E ddname SYSIN unavailable; already allocated.

Explanation: The SORT_BY parameter was specified for event reporting processing. To perform sort processing, Advanced Allocation Management must allocate ddname SYSIN to communicate with the sort routine, but discovered that the ddname was already allocated.

User response: If you are invoking the report utility in TSO through the user interface, perform the following steps:

1. Exit the Advanced Allocation Management user interface.
2. From the TSO command line, issue the command: `"TSO FREE F(SYSIN)."`
3. Invoke the Advanced Allocation Management user interface again.
4. Invoke the report utility from the user interface again.

If the problem persists, contact IBM Software Support.

GLO7054E DDname SYSOUT unavailable; already allocated.

Explanation: The SORT_BY parameter was specified for event reporting processing. To perform sort processing, Advanced Allocation Management must allocate ddname SYSOUT to communicate with the sort routine, but discovered that the ddname was already allocated.

User response: If you are invoking the report utility in TSO through the user interface, perform the following steps:

1. Exit the Advanced Allocation Management user interface.
2. From the TSO command line, issue the command: `"TSO FREE F(SYSOUT)."`
3. Invoke the Advanced Allocation Management user interface again.
4. Invoke the report utility from the user interface again.

If the problem persists, contact IBM Software Support.

GLO7055E Error occurred during allocation for SORTIN.

Explanation: The SORT_BY parameter was specified for event reporting processing. To perform sort processing, Advanced Allocation Management must allocate ddname SORTIN to communicate with the sort routine, but the allocation was unsuccessful.

User response: Contact IBM Software Support.

GLO7056E Error occurred during allocation for SORTOUT.

Explanation: The SORT_BY parameter was specified for event reporting processing. To perform sort processing, Advanced Allocation Management must allocate ddname SORTOUT to communicate with the sort routine, but the allocation was unsuccessful. Message GLO7049E accompanies this message and provides diagnostic information regarding the error.

User response: Contact IBM Software Support.

GLO7057E Error occurred during allocation for SYSOUT.

Explanation: The SORT_BY parameter was specified for event reporting processing. To perform sort processing, Advanced Allocation Management must allocate ddname SYSOUT to communicate with the sort routine, but the allocation was unsuccessful. Message GLO7049E accompanies this message and provides diagnostic information about the error.

User response: Contact IBM Software Support.

GLO7058E Error occurred during allocation for SYSIN.

Explanation: The SORT_BY parameter was specified for event reporting processing. To perform sort processing, Advanced Allocation Management must allocate ddname SYSIN to communicate with the sort routine, but the allocation was unsuccessful. Message GLO7049E accompanies this message and provides diagnostic information regarding the error.

User response: Contact IBM Software Support.

GLO7059E OPEN failed for SYSIN file.

Explanation: The SORT_BY parameter was specified for event reporting processing. To perform sort processing, Advanced Allocation Management must open the ddname SYSIN to communicate with the sort routine, but the OPEN operation was unsuccessful.

User response: Look in the job log for data management (IEC*xxx*) messages that contain more information about the error. If the problem persists, contact IBM Software Support.

GLO7060E OPEN failed for SYSOUT file.

Explanation: The SORT_BY parameter was specified for event reporting processing. To perform sort processing, Advanced Allocation Management must open the ddname SYSOUT to communicate with the sort routine, but the OPEN operation was unsuccessful.

User response: Look in the job log for data management (IEC*xxx*) messages that contain more information about the error. If the problem persists, contact IBM Software Support.

GLO7061E Error occurred during sort processing; sort messages follow-

Explanation: The SORT_BY parameter was specified for event reporting processing. Advanced Allocation Management invoked the sort routine, but the sort was unsuccessful. Advanced Allocation Management copies the messages from the sort to the CNTLOUT file following this message.

User response: Evaluate the sort messages that follow and attempt to correct the problem. If you cannot correct the problem or if the problem persists, contact IBM Software Support.

GLO7080E Invalid value specified for keyword STARTTIME.

Explanation: The value specified for the STARTTIME filter criteria is not valid. The value must be in the format: yyyy/mm/dd-hh:mm:ss.

User response: Verify the value that is specified for the STARTTIME filter criteria and ensure that it meets the required format. For more information, see "Event recording and reporting filter criteria descriptions" on page 302.

GLO7081E Invalid value specified for keyword ENDTIME.

Explanation: The value that is specified for the ENDTIME filter criteria is not valid. The value must be in the format: yyyy/mm/dd-hh:mm:ss.

User response: Verify the value that is specified for the ENDTIME filter criteria and ensure that it meets the required format. For more information, see "Event recording and reporting filter criteria descriptions" on page 302.

GLO7082E Invalid value specified for keyword JOBTYPE-

Explanation: The value that is specified for the JOBTYPE filter criteria is not valid.

User response: Verify the value that is specified for the JOBTYPE filter criteria and ensure that it specifies one of the valid values. For more information, see "Event recording and reporting filter criteria descriptions" on page 302 and "Event recording and reporting sort field descriptions" on page 305.

GLO7083E Invalid value specified for keyword DSORG.

Explanation: The value that is specified for the DSORG filter criteria is not valid.

User response: Verify the value that is specified for

the DSORG filter criteria and ensure that it specifies one of the valid values. For more information, see “Event recording and reporting filter criteria descriptions” on page 302 and “Event recording and reporting sort field descriptions” on page 305.

GLO7084E Wildcard prefix not allowed for specified filter criteria.

Explanation: A filter criteria was specified with a wildcard prefix indicator ("*") in the value, but the specified filter criteria does not allow wildcard prefixing.

User response: Specify the value in the filter criteria again, and remove the wildcard prefix indicator.

SMF Reporting Utility messages

These messages are related to issues with the SMF Reporting Utility.

GLO7603W Tape mount requires APF-authorization, tape input skipped.

Explanation: One or more specified SMF data sets reside on tape. To request tapes to be mounted, you must run the GLOSMFRP utility with APF authorization.

User response: Ensure that the load library that you run the GLOSMFRP utility from is APF-authorized.

GLO7610E OPEN failed for CNTLIN file.

Explanation: OPEN processing was unsuccessful for the CNTLIN file.

User response: Ensure that the CNTLIN DD statement is present in the JCL and that it points to a valid data set. For more information, see “Configuring the SMF Reporting Utility JCL” on page 307. Look in the job log for data management (IEC*xxx*) messages that contain more information about the error. If the problem persists, contact IBM Software Support.

GLO7612E Control statement parm exceeds maximum value allowed.

Explanation: The value that is specified for the control statement parameter was exceeded.

User response: Verify the syntax and specify a valid value. For more information, see “Configuring the SMF Reporting Utility JCL” on page 307.

GLO7613E Error(s) encountered during control statement processing.

Explanation: Errors were encountered while processing the SMF Reporting Utility control statements.

User response: Review the control statements and the prior error messages for more information. Verify that the syntax is correct. For more information, see “Configuring the SMF Reporting Utility JCL” on page 307.

GLO7614E Error occurred during allocation of temporary work file.

Explanation: An error occurred during the allocation of the temporary work file that is used during SMF Reporting Utility processing. One or more messages accompany this message and provide diagnostic information about the error.

User response: Contact IBM Software Support.

GLO7615E OPEN failed for TEMPFILE for output processing.

Explanation: OPEN processing was unsuccessful for the temporary work file that is used during SMF Reporting Utility output processing.

User response: Look in the job log for data management (IEC*xxx*) messages that contain more information about the error. If the problem persists, contact IBM Software Support.

GLO7616E Error occurred during allocation of SMF data set.

Explanation: An error occurred during the allocation of the SMF data set that is specified by the SMF_DATASET control statement. Messages GLO7618E and GLO7649E accompany this message and indicate the SMF data set name that is in error and provide diagnostic information about the error.

User response: Verify that the specified data set name exists and is cataloged.

GLO7617E OPEN failed for SMF data set -

Explanation: OPEN processing was unsuccessful for the specified SMF data set. Message GLO7618E accompanies this message and indicates the SMF data set name that is in error.

User response: Look in the job log for data management (IEC*xxx*) messages that contain more information about the error. If the problem persists, contact IBM Software Support.

GLO7619E No SMF records were found in any input file.

Explanation: No SMF records were found in the

specified SMF data sets to be searched for Advanced Allocation Management records.

User response: Ensure that the data sets that are specified by the SMF_DATASET control statement contain valid SMF data.

GLO7620E No SMF space abend records were found in any input SMF_DATASET files.

Explanation: No SMF space abend records were found in the SMF data sets to be searched for Advanced Allocation Management records.

User response: Verify that the appropriate steps for the SMF Reporting Utility were performed. Ensure that the SMF_DATASET files contained SMF 30 subtype 4 records during the time the space abends occurred.

GLO7621E Control statement parm non-numeric.

Explanation: The value that is specified for the control statement parameter is not numeric.

User response: Specify a numeric value. For more information, see "Configuring the SMF Reporting Utility JCL" on page 307.

GLO7622W Detail report not written because HISTOUT file was not defined.

Explanation: A HISTOUT file was not specified on the HISTOUT parameter.

User response: Specify where the abending jobs detail report should be written on the HISTOUT parameter. For more information, see "Required DD statements" on page 307.

GLO7624E SORT_BY field invalid.

Explanation: The column name that is specified for the SORT_BY parameter is not valid.

User response: Specify one of the following valid column names: SYS, JOBNAME, STEPNAME, PROGRAM, CMP, DATETIME, CPUTIME, or ELAPSEDTIME. For more information, see "Valid input control statements" on page 308.

GLO7625E Above control statement contains one or more errors.

Explanation: A control statement that is specified as input to the SMF Reporting Utility contains an error.

User response: Review the control statement and verify that the keyword, value, and syntax is valid. For more information, see "Valid input control statements" on page 308.

GLO7626E No data set names were specified for SMF_DATASET.

Explanation: A data set name was not specified for the SMF_DATASET parameter in the SMF Reporting Utility JCL. A minimum of one data set must be specified for reporting to take place.

User response: Specify one or more names of SMF data sets to be searched for Advanced Allocation Management Projected Savings Report records. A maximum of 32 data sets can be specified. For more information, see "Valid input control statements" on page 308.

GLO7627E CPU_TIME_COST_PER_HOUR is required, but was not specified.

Explanation: To generate the Projected Savings Report, the SMF Reporting Utility requires a CPU_TIME_COST_PER_HOUR control statement.

User response: Add a CPU_TIME_COST_PER_HOUR control statement to the SMF Reporting Utility input. For more information, see "Valid input control statements" on page 308.

GLO7628I Parenthesis unmatched before end-of-file on CNTLIN data set.

Explanation: While processing the CNTLIN data set, a required matching parenthesis was found to be missing.

User response: Review the input control statements, add the required parenthesis to the appropriate control statement, and run the SMF Reporting Utility again.

GLO7631E Maximum number of SMF_DATASET files (32) exceeded.

Explanation: The number of SMF data sets that are specified through the SMF_DATASET control statement is greater than the maximum number allowed.

User response: Modify the SMF_DATASET control statement to specify no more than 32 SMF data sets and run the SMF Reporting Utility again.

GLO7632E ELAPSED_TIME_LOST_COST_PER_HOUR required, but was not specified.

Explanation: The ELAPSED_TIME_LOST_COST_PER_HOUR parameter was not specified.

User response: Add an ELAPSED_TIME_LOST_COST_PER_HOUR statement that specifies a value no greater than 99999 and run the SMF Reporting Utility again. For more information, see "Valid input control statements" on page 308.

GLO7633E AVERAGE_ABEND_RECOVERY_TIME required, but was not specified.

Explanation: The AVERAGE_ABEND_RECOVERY_TIME parameter was not specified.

User response: Add an AVERAGE_ABEND_RECOVERY_TIME statement that specifies a value no greater than 99999 and run the SMF Reporting Utility again. For more information, see "Valid input control statements" on page 308.

GLO7634E ABEND_RECOVERY_TIME_COST required, but was not specified.

Explanation: The ABEND_RECOVERY_TIME_COST parameter was not specified.

User response: Add an ABEND_RECOVERY_TIME_COST statement that specifies a value no greater than 99999 and run the SMF Reporting Utility again. For more information, see "Valid input control statements" on page 308.

GLO7635E Error occurred in CONVTOD service.

Explanation: A non-zero return code was returned from the system time-of-day conversion routine. The routine is required for the SMF Reporting Utility to work properly.

User response: Contact IBM Software Support.

GLO7636E Error occurred in STCKCONV service.

Explanation: A non-zero return code was returned from the system store-clock conversion routine. The routine is required for the SMF Reporting Utility to work properly.

User response: Contact IBM Software Support.

GLO7637E Required delimiter "=" is missing in above control statement.

Explanation: While processing the control statements for the SMF Reporting Utility, a control statement was encountered that requires an equal sign (=), but it was found to be missing.

User response: Add the required delimiter to the appropriate control statement and run the SMF Reporting Utility again.

GLO7638E Data set specified by SMF_DATASET is not cataloged.

Explanation: One or more of the data sets that are specified by the SMF_DATASET parameter are not cataloged. Message GLO7618E accompanies this message and indicates the SMF data set names that were not found in the catalog.

User response: Check the data sets that are specified by the SMF_DATASET parameter and verify that they are cataloged.

GLO7639E OPEN failed for TEMPFILE file for input processing.

Explanation: OPEN processing was unsuccessful for the TEMPFILE file that is required for input processing.

User response: Look in the job log for data management (IEC*xxx*) messages that contain more information about the error. If the problem persists, contact IBM Software Support.

GLO7640E Unable to locate required ddname - SMFIN

Explanation: During SMF reporting processing, Advanced Allocation Management was not able to locate the SMF input data set ddname (SMFIN) in the TIOT. Message GLO7618E accompanies this message and indicates the SMF data set name that was being processed at the time of the error. This error is an internal error.

User response: Contact IBM Software Support.

GLO7641E Error occurred during SWAREQ JFCB retrieval for SMFIN, rc=*rc*,

Explanation: During event reporting processing, Advanced Allocation Management issued a SWAREQ request to retrieve the JFCB for the SMF input data set ddname (SMFIN). The SWAREQ request was unsuccessful and returned with return code *rc*. Message GLO7618E accompanies this message and indicates the SMF data set name that is being processed at the time of the error. This error is an internal error.

User response: Contact IBM Software Support.

GLO7644E Invalid DSORG (VSAM) for data set specified by SMF_DATASET -

Explanation: One or more data sets that are specified by the SMF_DATASET parameter do not have a valid DSORG (PS). Message GLO7618E accompanies this message and indicates the SMF data set name that has the incorrect DSORG.

User response: The SMF Reporting Utility works only with SMF data that was previously unloaded from the system SMF (MAN) data sets. Ensure that all SMF data sets that are specified by the SMF_DATASET parameter have a DSORG of PS.

GLO7649E DYNALLOC return code=*retcode*, error code=*errcode*, info code=*infocode*.

Explanation: During SMF Reporting Utility processing, a dynamic allocation request failed. This

message accompanies a previous message and provides diagnostic information about the error. *retcode* is the value that is returned in register 15 by the dynamic allocation service. *errcode* and *infocode* are the error and info code values that are returned in fields S99ERROR and S99INFO in the dynamic allocation request block.

User response: The *retcode*, *errcode*, and *infocode* values can be found in the *MVS Programming: Authorized Assembler Services Guide*, which can help determine the exact nature of the problem. Contact IBM Software Support, if needed.

GLO7650E Unable to load sort routine.

Explanation: During SMF reporting processing, the SORT_BY parameter was specified, which caused Advanced Allocation Management to attempt to load a module with the name of SORT, but the load was unsuccessful.

User response: If the module SORT does not reside in link list or LPALIB on your system, add a DD to the STEPLIB concatenation of the SMF Reporting Utility step and specify the name of the loadlib in which SORT resides. If the module SORT resides in link list or LPALIB on your system, or you add a DD to the STEPLIB concatenation of the SMF Reporting Utility step and the problems persists, contact IBM Software Support.

GLO7651E Unable to locate required ddname - TEMPFILE.

Explanation: During SMF Reporting Utility processing, the SORT_BY parameter was specified. While attempting to perform sort processing, Advanced Allocation Management was not able to not locate its temporary processing DD (TEMPFILE) in the TIOT. This error is an internal error.

User response: Contact IBM Software Support.

GLO7652E Error occurred during SWAREQ JFCB retrieval for TEMPFILE, rc=*rrrrrrrr*.

Explanation: During SMF Reporting Utility processing, the SORT_BY parameter was specified. While attempting to perform sort processing, Advanced Allocation Management issued a SWAREQ request to retrieve the JFCB for its temporary processing DD (TEMPFILE). The SWAREQ request was unsuccessful, returning with return code *rrrrrrrr*. This error is an internal error.

User response: Contact IBM Software Support.

GLO7653E ddname SYSIN unavailable; already allocated.

Explanation: During SMF Reporting Utility processing, the SORT_BY parameter was specified. To perform sort processing, Advanced Allocation Management must allocate the ddname SYSIN to communicate with the sort routine, but discovered that the ddname was already allocated.

User response: If the problem persists, contact IBM Software Support.

GLO7654E DDname SYSOUT unavailable; already allocated.

Explanation: During SMF Reporting Utility processing, the SORT_BY parameter was specified. To perform sort processing, Advanced Allocation Management must allocate the ddname SYSOUT to communicate with the sort routine, but discovered that the ddname was already allocated.

User response: If the problem persists, contact IBM Software Support.

GLO7655E Error occurred during allocation for SORTIN.

Explanation: During SMF Reporting Utility processing, the SORT_BY parameter was specified. To perform sort processing, the utility must allocate the ddname SORTIN to communicate with the sort routine, but the allocation was unsuccessful.

User response: Contact IBM Software Support.

GLO7656E Error occurred during allocation for SORTOUT.

Explanation: During SMF Reporting Utility processing, the SORT_BY parameter was specified. To perform sort processing, the utility must allocate the ddname SORTOUT to communicate with the sort routine, but the allocation was unsuccessful. Message GLO7649E accompanies this message and provides diagnostic information about the error.

User response: Contact IBM Software Support.

GLO7657E Error occurred during allocation for SYSOUT.

Explanation: During SMF Reporting Utility processing, the SORT_BY parameter was specified. To perform sort processing, the utility must allocate the ddname SYSOUT to communicate with the sort routine, but the allocation was unsuccessful. Message GLO7649E accompanies this message and provides diagnostic information about the error.

User response: Contact IBM Software Support.

GLO7658E Error occurred during allocation for SYSIN.

Explanation: During SMF Reporting Utility processing, the SORT_BY parameter was specified. To perform sort processing, the utility must allocate the

ddname SYSIN to communicate with the sort routine, but the allocation was unsuccessful. Message GLO7649E accompanies this message and provides diagnostic information about the error.

User response: Contact IBM Software Support.

GLO7659E OPEN failed for SYSIN file.

Explanation: During SMF Reporting Utility processing, the SORT_BY parameter was specified. To perform sort processing, the utility must open the ddname SYSIN to communicate with the sort routine, but the OPEN operation was unsuccessful.

User response: Look in the job log for data management (IEC*xxx*) messages that contain more information about the error. If the problem persists, contact IBM Software Support.

GLO7660E OPEN failed for SYSOUT file.

Explanation: During SMF Reporting Utility processing, the SORT_BY parameter was specified. To perform sort processing, the utility must open the ddname SYSOUT to communicate with the sort routine, but the OPEN operation was unsuccessful.

User response: Look in the job log for data management (IEC*xxx*) messages that contain more information about the error. If the problem persists, contact IBM Software Support.

GLO7661E Error occurred during sort processing; sort messages follow-

Explanation: During SMF Reporting Utility processing, the SORT_BY parameter was specified. The utility invoked the sort routine, but the sort was unsuccessful. Advanced Allocation Management copies the messages from the sort to the CNTLOUT file following this message.

User response: Evaluate the sort messages that follow and attempt to correct the problem. If you cannot correct the problem or if the problem persists, contact IBM Software Support.

GLO7690E Duplicate keyword specified above.

Explanation: A duplicate keyword was specified.

User response: Remove the duplicate keyword.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

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.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some 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 is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which 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.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the Unites States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Privacy policy considerations

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.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

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 http://www.ibm.com/privacy and the section titled "Cookies, Web Beacons, and Other Technologies" in IBM's Online Privacy Statement at http://www.ibm.com/privacy/details. Also, see the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Glossary

This glossary includes definitions reprinted from the *American National Dictionary for Information Processing Systems*, various ISO publications, and the *CCITT Eighth Plenary Assembly Red Book, Terms and Definitions*. Definitions are identified as follows:

- Definitions reprinted from the *American National Dictionary for Information Processing Systems*, copyright 1982 by the Computer Business Equipment Manufacturers Association (CBEMA), are identified by the symbol (A) following the definition. Copies can be purchased from the American Standards Institute, 1430 Broadway, New Your, New York 10018.
- Definitions reprinted from a published section of the International Organization for Standardization's (ISO) *Vocabulary—Information Processing* or from a published section of the ISO *Vocabulary—Office Machines* are identified by the symbol (I) following the definition. Because many ISO definitions are also reproduced in the *American National Dictionary for Information Processing Systems*, ISO definitions may also be identified by the symbol (A).
- Definitions reprinted from working documents, draft proposals, or draft international standards of ISO Technical Committee 97, Subcommittee 1 (Vocabulary) are identified by the symbol (T) following the definition, indicating that final agreement has not yet been reached among its participating members.
- Definitions are also reprinted from the *CCITT Eighth Plenary Assembly Red Book, Terms and Definitions* and working documents published by the International Telegraph and Telephone Union, Geneva, 1985.
- Definitions that are specific to IBM products are so labeled.

The following cross-references are used in this glossary:

Contrast with. This refers to a term that has an opposed or substantively different meaning.

Synonym for. This indicates that the term has the same meaning as a preferred term, which is defined in its proper place in the dictionary.

Synonym with. This is a backward reference from a defined term to all other terms that have the same meaning.

See. This refers the reader to multiple-word terms that have the same last word.

See also. This refers the reader to terms that have a related, but not synonymous, meaning.

abend The abnormal end of a task, job, or subsystem, due to an error condition that cannot be resolved.

B37 abend
A B37 abend indicates that an error was detected by the end-of-volume routine. A B37 abend can occur in the following situations: If the DASD volume to which the system has assigned one of the program's output data sets does not have enough available space to allow the necessary

secondary allocations to be made. Or, if the output data set has used all of the 16 extents of secondary allocations —but needs more space.

configuration
The process of describing to a system the devices, optional features, and program products that have been installed so that these features can be used.

customization
The process of describing optional changes to defaults of a software program that is already installed on the system and configured so that it can be used.

D37 abend
A D37 abend indicates that an error occurred when an output operation to a direct access device was requested. This error occurs when an output data set has used all of the primary space and no secondary space was requested.

DASD
Direct Access Storage Device. A storage device for which data access time is independent of the physical location of that data on the storage device.

data set
(1) A named set of records that are stored or processed as a unit. (2) The major unit of data storage and retrieval. A data set consists of a collection of data in one of several prescribed arrangements and described by control information to which the operating system has access.

E37 abend
An E37 ABEND indicates that an error occurred when an output operation was requested. The data set was on a direct access or magnetic tape device.

ESDS An entry sequenced data set (VSAM).

extent A continuous space on a DASD volume occupied by a data set or portion of a data set.

file Synonym for data set.

free space
Space reserved within the control intervals of a key-sequenced data set for inserting new records into the data set in key sequence —or for lengthening records that are already there; also, whole control intervals reserved in a control area for the same purpose.

install (1) To add a program, program option, or software program to the system in a manner such that it may be executed and will interact properly with all affected programs in the system. (2) To connect a piece of hardware to the processor.

intercept
A feature of a software program that provides for future expansion. For example, in a program, an intercept might call an external routine, or allow

for entry of a variable. An intercept is similar, in concept, to a program exit—but unlike a program exit (which already exists in the code), an intercept is placed into the program by the developer.

job control language (JCL)
Job Control Language. A control language that specifies a job and its requirements to the operating system.

job entry subsystem (JES)
Job Entry Subsystem. JES is a facility of the OS/390 and MVS operating systems that manages jobs (units of work) that the system executes. There are two versions of JES, JES2 and JES3.

MVS image
A single copy of the MVS operating system, executing in a real machine, or within a logical partition of an MVS configuration.

parameter
Information that the user supplies to a panel, command, or function.

partitioned data set (PDS)
Partitioned data set. A data set on direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

partitioned data set extended (PDSE)
A system-managed data set that contains an indexed directory and members that are similar to the directory and members of partitioned data sets. A PDSE can be used instead of a partitioned data set except that a PDSE may contain only program objects or only members other than program objects.

resource access control facility (RACF)
Resource Access Control Facility. A program providing security facilities to protect archive data and forms where specified.

relative record data set (RRDS)
Relative record data set. A type of VSAM data set whose records have fixed or variable lengths, and are accessed by relative record number.

System Management Facility (SMF)
A z/OS facility that collects and records a variety of system and job-related information. For example, statistics, accounting information, and performance data.

storage management subsystem (SMS)
Storage Management Subsystem. 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.

STOW macro (STOW processing)
The MVS STOW macro updates partitioned data set directories.

Virtual Storage Access Method (VSAM)
(1) An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability. VSAM is emulated in CICS to control access to files. 2) An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by relative-record number.

volume pooling
Technique for managing DASD (also known as volume or storage pooling – or volume grouping). Volume pooling consists of categorizing data by factors such as: performance, recovery, and application requirements, and then allocating data to pools of DASD volumes according to that classification. Data types for pooling might be database, interactive temporary data, large and multi-volume data sets, and system data. Pooling is normally associated with a data naming convention.

Index

Special characters

A

B

C

D

E

F

G

I

J

K

L

M

N

O

P

R

S

T

U

V

W

Product Number: 5697-P35

Printed in USA

SC23-9817-05