

CA MIA Tape Sharing for z/OS

CA MIA Programming Guide

Release 12.0



This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2014 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This document references the following CA Technologies products:

- CA 1® Tape Management (CA 1)
- CA Easytrieve® Report Generator (CA Easytrieve)
- CA MIA Tape Sharing (CA MIA)
- CA MIC Message Sharing (CA MIC)
- CA MII Data Sharing (CA MII)
- CA MIM™ Resource Sharing (CA MIM)
- CA OPS/MVS® Event Management and Automation (CA OPS/MVS)

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at <http://ca.com/docs>.

Documentation Changes

The following documentation updates have been made since the last release of this documentation:

Note: In PDF format, page references identify the first page of the topic in which a change was made. The actual change may appear on a later page.

- Added the [How You Exclude Devices from CA MIA Management](#) (see page 37) section.
- Added the [Specify Devices in the DEVEXCL Member](#) (see page 38) section.
- Added the [Dynamic Detection and Notification \(DDN\)](#) (see page 67) section.
- Added the [Contention Delay Detection](#) (see page 68) section.
- Added the [WAIT NOHOLD Detection](#) (see page 68) section.
- Updated the [Plan the MIMINIT Member](#) (see page 17) section.
- Updated the [Plan the MIMCMNDS Member](#) (see page 18) section.
- Updated the [Resynchronization: RESYNCH Command](#) (see page 39) section.
- Updated the [Resynchronization: HCD ACTIVATE](#) (see page 41) section.
- Updated the [Report Generation for CA MIA](#) (see page 148) section.

Contents

Chapter 1: Introduction	11
Overview	11
Automatic Cartridge Loading	12
GTAF	12
Device Integrity	12
Tape Sharing.....	13
TPCF.....	13
Device Selection	13
How You Capture Device Information	14
Chapter 2: Planning Initial Settings	15
Overview	15
General Statements	16
Facility-Specific Members	16
Plan the MIMINIT Member	17
Plan the MIMCMNDS Member	18
Plan the MIMSYNCH Member	20
Plan the MIMUNITS Member.....	20
Chapter 3: Advanced Topics	21
Tape Device Allocation and Allocation Recovery Considerations	21
How z/OS Device Allocation Works.....	22
How z/OS Device Allocation with CA MIA Works.....	24
How z/OS Device Allocation Recovery Works	25
How z/OS Device Allocation Recovery with CA MIA Works.....	27
Product Activation and Termination Considerations	30
How You Identify Devices to CA MIA	30
Identify a Class of Devices	31
Identify a List of Devices	32
Specify Devices in the MIMUNITS Member	33
Code the MIMUNITS Member	35
Direct MIMUNITS Entries to Certain Systems	36
How You Exclude Devices from CA MIA Management	37
Specify Devices in the DEVEXCL Member	38
How You Verify Whether Devices are Defined to CA MIA	38
Compatibility Restrictions	39

How You Dynamically Change Which Devices Are Managed	39
Resynchronization: RESYNCH Command	39
Resynchronization: HCD ACTIVATE	41
z/OS Hardware Configuration Definition (HCD) ACTIVATE Considerations	41
Overview of HCD ACTIVATE	42
HCD ACTIVATE	43
Configure Tape Devices Offline until CA MIA Synchronization	45
VARY Device Serialization	46
VARY Command Serialization	48
Overview of Managed Device VARY Processing	49
Delays in Tape Allocation, UNLOAD, SWAP, and Online Processing	50
Allocation Recovery and Online Processing	50
Define CA MIA in the Subsystem Name Table	50
How You Change the Status of Managed Devices (VARY)	51
VARY Devices OFFLINE and ONLINE	52
How You Make a Device Unavailable for Allocation	53
Force a Device Offline while Allocated to Another System	55
Wake Jobs Waiting for Devices	56
Wake (POST) Jobs Waiting for Devices	57
Activate the Automatic Cartridge Loader	57
Respond to ASSIGN Processing Requests	58
How You Influence Device Selection	59
Assign a Preference Value to a Device	59
Dedicate a Device to a System	60
Reserve a Device for a Job	61
How You Reserve a Device for a Single Job	61
Releasing a Job Reserved Device	61
Reserve a Device for a Group of Jobs	62
How You Force a Job to Use a Reserved Device	62
How You Limit Access to Devices Through Job Reserve	63
How You Limit Access to Devices Through the OVERGENNED Status	65
Using Exit Routines to Influence Device Selection	67
Dynamic Detection and Notification (DDN)	67
Contention Delay Detection	68
WAIT/NOHOLD Detection	68
TPCF Device Processing	69
Factors That Influence TPCF Preferencing	69
How the EDL Limits TPCF Preferencing	69
How z/OS and Other Vendor Products Eliminate Eligibility of Devices	70
How z/OS and Other Vendor Products Preference Devices	70
z/OS ACL Preferencing	70
Unexpected Results in Device Allocation	71

How You Control Allocation Recovery Actions of TPCF.....	72
How You Control Recovery Processing Using SETOPTION AUTOREPLY	72
How You Override the z/OS Default Value for Automatic Replies	73
How TPCF Responds When a Job Enters Allocation Recovery	73
How You Respond when CANCEL Is the Only Option.....	74
How TPCF Removes Devices from the Offline Device List	74
How You Respond When a Job is Waiting for a Device.....	76
How You Determine Where to Send Message MIM2046	77
How You Use Exit Routines to Control Allocation Recovery	77
z/OS Dynamic Device Reconfiguration (SWAP) Processing.....	77
SWAP Processing.....	78
Performance Considerations.....	80
Setting INTERVAL and CYCLE Parameters	80
How You Automate Replies to Allocation Recovery Prompts (AUTOREPLY)	81
Reply To SWAP WTORs	81
Separate CA MIM into Multiple Address Spaces.....	82
Sysplex Considerations.....	82
z/VM Considerations	82
Shared Tape Support.....	83
Guests Attached by MULTUser Option.....	85
How You Define Multiple Addresses per Device.....	86
Part-time Guest Considerations.....	87
The Autopath Feature.....	87

Chapter 4: Troubleshooting 91

How You Obtain Information for Diagnosing Problems.....	91
Tape Device Allocation Problems.....	92
Overview of GTALOCAL Enqueue Processing.....	92
Diagnosing Allocation Issues	94
Delay Detection and Notification	97
Contention Detection.....	98
Wait/NoHold Detection	99
Tape Device Allocation Recovery Problems.....	99
TPCF Does Not Reply WAIT to the z/OS IEF238D Message.....	99
z/OS Issues Message IEF238D for a Job that Should Have Allocated An Online Device	100
An OFFLINE Device That Should Not Be Used Appears In The OFFLINE Device List.....	100
TPCF Does Not Respond to z/OS IEF238D Message.....	100
Message MIM2042 Does Not List All of the Offline Devices the Job Can Use	101
NOTAVAILABLE or Externally DEDICATED Devices Are Appearing In The Offline Device List	101
z/OS Issues Message IEF700I	102
Tape Device Preferencing Problems	102

Ineligible Device Allocated	102
Less-preferred Device Allocated	103
Tape Device Dual Allocation Problems.....	103
How You Activate Tracing	107
Activate the TRACE Feature	107
Activate Printing for a Specific Facility	107
Activate the Tracing Function for a Specific Facility.....	108
Reverse Tracing Settings for a Specific Facility	108
Reverse Printing Settings for a Specific Facility.....	108
How You Obtain Dumps	108
Obtain a Dump of Facility Activity.....	109
Obtain a System Dump	109
How to Restart CA MIA in an ACTIVE MIAplex	109
Contact Technical Support	117

Chapter 5: User Exits 119

How You Manage User Exits	119
Activate Exit Routines	120
View Exit Routine Information	120
MIMINIXT (Common Exit Interface).....	120
Rules for Coding Exit Routines	121
Notes on Using Exit Routines	121
Sample User Exits	122
Exit Routine Programming Considerations	122
TPCEDLXT Exit	122
TPCRECXT Exit	125
TPCSRMXT Exit	131

Chapter 6: Utilities and Other Interfaces 135

The Application Program Interface	135
Prepare the Interface	136
The MIMAPI1 Module	137
How You Retrieve Device Information.....	145
How You Start the Interface.....	146
The Retrieval Panel	147
The Display Panel	148
Exit from the Interface	148
Report Generation for CA MIA	148
How You Plan Reports.....	149
Sample Tape Usage Statistics Report (TP).....	151

Appendix A: Integration with CA OPS/MVS	155
Overview	155
Enable CA OPS/MVS Event Notification	156
API Rules.....	156
CA MIA VARY Delay Detected Event	156
CA MIA VARY REQUEUE Notification Event	158
CA MIA VARY Delay Abort Complete Event	160
Index	163

Chapter 1: Introduction

This section contains the following topics:

[Overview](#) (see page 11)

[Automatic Cartridge Loading](#) (see page 12)

[GTAF](#) (see page 12)

[TPCF](#) (see page 13)

Overview

CA MIA is the component of CA MIM that provides integrity when you are sharing tape devices among systems (or CPU images). This component also enhances z/OS device processing for tape devices. Use of this product requires a permanent channel path from each operating system to each tape drive that will be shared by those operating systems.

CA MIA improves device processing at your site by:

- Preventing jobs on different systems from concurrently allocating the same tape device and thereby compromising data set integrity. By protecting device integrity, CA MIA enables you to share tape devices among systems or images.
- Influencing device selection during the z/OS allocation process.
- Automating responses to allocation recovery messages issued by z/OS whenever a job cannot allocate a suitable online device.
- Capturing information about device activities through an application program interface.
- Improving operations for assignable tape devices by providing an easy way to activate the ACL feature on these devices.
- Providing an SMF report showing information about tape device usage using CA Easytrieve.
- Sharing remote tape devices between images.

Important! The term “assignable tape devices” refers to 3480, 3490, 3590, and any other compatible tape devices.

Note: If you are running z/OS systems as guests under z/VM, then you have multiple options for sharing a device among the guest systems using a single tape drive path. The available options are:

- [The Autopath feature](#) (see page 87) of CA MIA
- [The Shared Tape Support](#) (see page 83) feature of z/VM

For more information on the Autopath feature and Shared Tape Support, see z/VM Considerations in the chapter “Advanced Topics.”

CA MIA is composed of two facilities:

- The Global Tape Allocation Facility (GTAF)
- The Tape Preferencing and Control Facility (TPCF)

Automatic Cartridge Loading

Both GTAF and TPCF let you determine whether the automatic cartridge loader (ACL) feature is present and whether it is active. You obtain this information by issuing a display command.

GTAF

The GTAF prevents:

- Jobs on different systems from simultaneously allocating the same tape.
- Jobs from allocating devices already in use on another system.

Device Integrity

GTAF provides integrity for devices during the allocation process in two ways:

- By interfacing with the locking mechanism z/OS uses to locally serialize device allocation, GTAF prevents jobs on different systems from simultaneously allocating the same device.
- By communicating allocation information among systems, GTAF enables the local z/OS operating system to eliminate both locally and externally allocated devices when selecting a device for allocation. This prevents jobs from allocating devices that are already in use on another system.

z/OS provides device serialization on each system. GTAF then propagates that local serialization across all managed systems in the complex. As jobs on each system enter tape allocation, GTAF intercepts the request and forces the job to wait. GTAF then checks to verify that no job on any system is currently allocating from the same group of devices. When the requested device groups are free, the job is allowed to proceed through the normal allocation process.

Whenever a device is allocated or deallocated, GTAF writes a transaction representing the allocation or deallocation to the CA MIM control file. At regular intervals, GTAF on each system checks the control file for new information about device allocations and deallocations.

When GTAF reads a transaction about an allocation on another system, GTAF changes the local status of that device to “allocated.” This prevents jobs on the local system from allocating that device.

Tape Sharing

CA MIA provides methods to improve operations for assignable tape devices.

By intercepting system assignment requests for assignable tape devices, GTAF lets you share these tape devices among a group of systems. You can tell GTAF to convert all system assignment requests to multiple-system assignment requests so that any system providing a designated password can use those devices. This ensures that tape devices are shared among the systems you choose.

TPCF

The TPCF lets you influence device selection during the z/OS allocation process. TPCF also responds automatically to the messages z/OS issues when a job cannot allocate a suitable online device.

Device Selection

TPCF provides you with two ways to influence device selection during the z/OS allocation process:

- You can use the VARY command to change the status of any device being managed by CA MIA. By assigning special types of device status, you can influence device selection as follows:
 - Prevent a device from being allocated
 - Make a device unavailable for allocation on one or more systems
 - Make a device available for allocation on only one system
 - Reserve a device for a certain job or group of similarly named jobs
 - Assign device preference values, which causes TPCF to choose the most preferred device for an allocation whenever possible
- You can use optional exit routines to customize allocation at your site, based on any number of device allocation criteria.

TPCF can respond automatically to messages issued by z/OS when no device is available for an allocation request. By automating responses or providing an exit routine during this point in allocation (known as allocation recovery), TPCF helps alleviate manual intervention, reducing the chance of delays in z/OS allocation.

How You Capture Device Information

You can use the application program interface (API) to capture information about the status of your devices, including whether the device is currently allocated, the name of the volume mounted on a device, and the TPCF preferencing status for the device.

By capturing this information in a log or on a screen, you can perform a number of useful tasks, such as:

- Trace significant allocation events
- Measure device usage for system tuning
- Record statistics in a historical log

You can call the API from an external program, or you can use the sample display routine to view the device information at a TSO session.

Chapter 2: Planning Initial Settings

This section contains the following topics:

[Overview](#) (see page 15)

[General Statements](#) (see page 16)

[Facility-Specific Members](#) (see page 16)

[Plan the MIMINIT Member](#) (see page 17)

[Plan the MIMCMNDS Member](#) (see page 18)

[Plan the MIMSYNCH Member](#) (see page 20)

[Plan the MIMUNITS Member](#) (see page 20)

More information:

[Advanced Topics](#) (see page 21)

Overview

The MIMPARMS data set contains parameter values that CA MIM uses as input during product initialization to define the characteristics of the MIMplex. You specify the MIMPARMS data set on the //MIMPARMS DD statement in the JCL procedure used to start CA MIM. A sample MIMPARMS data set is installed with the CA MIM product into data set CAI.CBTDPARM.

You need to set initial values for statements and commands in the following CA MIM MIMPARMS parmlib members:

- MIMINIT
- MIMCMNDS
- MIMSYNCH
- MIMUNITS

Note: For detailed planning information, see the following:

- [Advanced Topics](#) (see page 21)
- *CA MIM Statement and Command Reference Guide*
- *CA MIM Programming Guide*

General Statements

The following general statements can be used in any of the CA MIM parmlib members:

- IFSYS
- ENDIF
- INCLUDE
- LOG
- NOLOG

Note: For more information, see the *CA MIM Statement and Command Reference Guide*.

Facility-Specific Members

Some parameter library members are used only when specific facilities have been activated. The MIMUNITS member is specific to GTAF and TPCF. This member:

- Defines which devices GTAF/TPCF will manage
- Assigns global names to devices
- Defines Autopath characteristics of devices

This member name is optionally pointed to by the MIMINIT `DEVLIST=mimname` statement in the MIMINIT member of the MIMPARMS data set. Using this member is optional.

Plan the MIMINIT Member

The MIMIINT member of the CA MIM parameter data set contains statements that specify initialization values for CA MIM. The initialization members specific to CA MIA do the following tasks:

- Determine which CA MIA facilities are activated at initialization. Use the following parameters:
 - MIMINIT GTAF
 - MIMINIT TPCF
- Determine which devices CA MIA will manage, assign global names, and define the Autopath participation. Use the following parameters:
 - MIMINIT DEVCLASS
 - MIMINIT DEVLIST
 - MIMINIT DEVEXCL
- Determine the type of ASSIGN placed on assignable tape devices. Use the following parameter:
 - GTAINIT ASSIGN
- Determine the order in which CA MIA allocation recovery exit routines are placed in the z/OS IEF_ALLC_OFFLN and IEF_SPEC_WAIT exit points. Use the following parameter:
 - TPCINIT AUTOREPLYPOS

Note: For more information, see the *CA MIM Statement and Command Reference Guide*.

More information:

[Assign a Preference Value to a Device](#) (see page 59)

[Identify a List of Devices](#) (see page 32)

[How z/OS Device Allocation Recovery with CA MIA Works](#) (see page 27)

Plan the MIMCMNDS Member

The MIMCMNDS member of the CA MIM parameter data set contains commands that specify operating values for CA MIM.

The commands specific to CA MIA do the following tasks:

- Control Delay Detection and Notification. Use the following parameter:
 - SETOPTION GTAF DDN
- Control the Analyze command defaults. Use the following parameter:
 - SETOPTION GTAF ANALYZE
- Control Autopath processing. Use the following parameter:
 - SETOPTION GTAF AUTOPATH
- Set format values for device displays. Use the following parameters:
 - SETOPTION GTAF GLOBALDISPLAY
 - SETOPTION TPCF LOCALDISPLAY
- Control RESYNCH processing in response to an HCD ACTIVATE or RESYNCH command. Use the following parameters:
 - SETOPTION GTAF RESYNCH
 - SETOPTION TPCF RESYNCH
- Set the values for collecting statistical records. Use the following parameters:
 - SETOPTION GTAF STATCOLLECT
 - SETOPTION GTAF STATCYCLE
 - SETOPTION GTAF STATINTERVAL
 - SETOPTION TPCF STATCOLLECT
 - SETOPTION TPCF STATCYCLE
 - SETOPTION TPCF STATINTERVAL
- Control the automatic response to allocation recovery messages and the content of offline device lists. Use the following parameter:
 - SETOPTION TPCF AUTOREPLY
- Set the default scope for the VARY commands. Use the following parameter:
 - SETOPTION GTAF VARYSCOPE
- Set the FORCE default for job reserve processing. Use the following parameter:
 - SETOPTION TPCF JOBFORCE

- Control whether the OVERGENNED devices are marked ineligible for jobs in allocation. Use the following parameter:
 - SETOPTION TPCF OVGINELIG
- Control ACL status messages. Use the following parameter:
 - SETOPTION TPCF MIM2032
- Control the updating of the USERDATA device field and issuing of USERDATA status messages. Use the following parameter:
 - SETOPTION TPCF USERDATA
- Control allocation timing functions. Use the following parameter:
 - SETOPTION GTAF LOCKTUNING
- Obtain diagnostic information. Use the following parameters:
 - SETOPTION GTAF MIM216X
 - SETOPTION GTAF SETPRINT
 - SETOPTION GTAF RESETPRINT
 - SETOPTION GTAF SETTRACE
 - SETOPTION GTAF RESETTRACE
 - SETOPTION TPCF SETPRINT
 - SETOPTION TPCF RESETPRINT
 - SETOPTION TPCF SETTRACE
 - SETOPTION TPCF RESETTRACE
 - SETOPTION TPCF MIM2044DIAG

Note: For more information, see the *CA MIM Statement and Command Reference Guide*.

More information:

[z/VM Considerations](#) (see page 82)

[How You Dynamically Change Which Devices Are Managed](#) (see page 39)

[How You Force a Job to Use a Reserved Device](#) (see page 62)

[How You Control Allocation Recovery Actions of TPCF](#) (see page 72)

[How You Limit Access to Devices Through the OVERGENNED Status](#) (see page 65)

[Activate the Automatic Cartridge Loader](#) (see page 57)

Plan the MIMSYNCH Member

The MIMSYNCH member of the CA MIM parameter data set contains CA MIM, z/OS, and z/OS subsystem commands you wish to execute when CA MIM synchronizes with the other systems in the MIMplex.

This member name is pointed to in the MIMPROC member by the PROC SYNCH=MIMSYNCH statement. Commands found in this member execute after MIMINIT statements and MIMCMNDS commands have executed, and only after all CA MIM address spaces in the complex have synchronized.

We recommend that VARY commands be placed in this member when you are running CA MIA.

You can use VARY commands to alter the online/offline status of a device. You can also use VARY commands to assign a special status to a device to influence device selection. See the section *Changing the Status of Managed Devices and Influencing Device Selection* in the chapter “Advanced Topics” in this guide.

Plan the MIMUNITS Member

The MIMUNITS member of the CA MIM parameter data set contains a list of device addresses you would like GTAF to manage. The MIMINIT DEVLIST statement in the MIMINIT member of the CA MIM parameter data set optionally points to this member name. In addition to defining devices to be managed by CA MIM, the MIMUNITS member can be used to assign GLOBAL names to devices and to assign Autopath eligibility to devices.

More information:

[How You Identify Devices to CA MIA](#) (see page 30)

Chapter 3: Advanced Topics

This chapter provides an in-depth analysis of the CA MIA functional interfaces to z/OS allocation, allocation recovery, SWAP processing, and dynamic hardware configuration. It also discusses performance, sysplex, and z/VM considerations.

This section contains the following topics:

[Tape Device Allocation and Allocation Recovery Considerations](#) (see page 21)

[Product Activation and Termination Considerations](#) (see page 30)

[How You Identify Devices to CA MIA](#) (see page 30)

[How You Dynamically Change Which Devices Are Managed](#) (see page 39)

[z/OS Hardware Configuration Definition \(HCD\) ACTIVATE Considerations](#) (see page 41)

[Configure Tape Devices Offline until CA MIA Synchronization](#) (see page 45)

[VARY Device Serialization](#) (see page 46)

[VARY Command Serialization](#) (see page 48)

[Define CA MIA in the Subsystem Name Table](#) (see page 50)

[How You Change the Status of Managed Devices \(VARY\)](#) (see page 51)

[How You Influence Device Selection](#) (see page 59)

[Dynamic Detection and Notification \(DDN\)](#) (see page 67)

[TPCF Device Processing](#) (see page 69)

[How You Control Allocation Recovery Actions of TPCF](#) (see page 72)

[z/OS Dynamic Device Reconfiguration \(SWAP\) Processing](#) (see page 77)

[Performance Considerations](#) (see page 80)

[Sysplex Considerations](#) (see page 82)

[z/VM Considerations](#) (see page 82)

Tape Device Allocation and Allocation Recovery Considerations

This section gives you a fundamental understanding of the z/OS device allocation and allocation recovery process.

How z/OS Device Allocation Works

Allocation is the process in which:

1. z/OS evaluates the device requirements of a job.
2. z/OS selects a device that best meets those requirements.

In normal allocation processing, z/OS selects only online devices that are not allocated currently and that are suitable for that job.

The z/OS allocation mechanism creates a list of devices that the requesting job may be able to use. This list is known as the eligible device list (EDL) and contains the UCB address of every suitable device. An EDL is created for each DD.

The EDL is made up of one or more device groups. For tape allocations, each device group contains a single device. z/OS chooses a device to allocate from the device groups in the EDL.

z/OS serializes the allocation process by allowing only one job to control a device group at a time. This process prevents two jobs on the same system from concurrently choosing the same device.

As each DD for a job goes through allocation, z/OS requests the group locks for all the device groups in the EDL for that DD. The job must get control of all the locks for all the groups in the EDL before z/OS allows it to proceed in allocation. If another job already controls that device group, then z/OS makes the requesting job wait for the device group to be released.

Once all the device group locks are obtained for a DD, if a job cannot allocate a suitable online device, then the job enters the z/OS allocation recovery process.

The following steps give an overview of the z/OS allocation process:

1. z/OS allocation assembles the EDL for a DD for the job, based on the device groups z/OS creates during the system generation (SYSGEN) process.
2. z/OS or vendor products may mark devices in the EDL ineligible for allocation for a variety of reasons. For example, robotic software may mark devices ineligible based on whether the device is inside or outside the robot.
3. z/OS requests the device group locks for all device groups in the EDL. When all the locks are obtained, the job is allowed to proceed through allocation.
4. z/OS examines the EDL and eliminates ineligible, allocated, inaccessible, and offline devices from possible allocation. If there are no devices that can be allocated, then the job enters allocation recovery. If there is one device, then the job allocates it. If there are multiple devices, then the job goes to Step 5.

5. z/OS or vendor products may preference devices in the EDL over other devices for a variety of reasons. For example, z/OS may preference a device based on its ACL status.
6. z/OS chooses a device for allocation from the remaining devices based on the preferencing values set for the devices by z/OS, vendor products, or both.

How z/OS Device Allocation with CA MIA Works

The two CA MIA facilities, GTAF and TPCF, interact with the z/OS allocation mechanism in different ways.

The function of GTAF in the allocation process is to prevent concurrent allocations of the same tape drive on multiple systems from occurring. GTAF extends the device serialization from one system to all systems in the CA MIA complex. To accomplish this goal, GTAF performs the following actions:

- GTAF controls cross-system access to the device group locks that z/OS uses to serialize device allocation.
- GTAF communicates allocation information among systems after a device has been selected for a job.

One of the functions of TPCF in the allocation process is to influence device selection. To do this, TPCF intercepts the EDL created by the z/OS allocation mechanism and eliminates or preferences devices according to the criteria you establish.

Note that CA MIA facilities interact with the z/OS allocation mechanism only when a managed device is on the EDL for a job. Therefore, GTAF only serializes allocation for managed devices, and TPCF intercepts the EDL only when a managed device is on the list. In addition, TPCF acts only on managed devices from these lists. TPCF does not act on devices that you have not placed under CA MIA management.

Note: To determine if a device is managed, see *Verifying Whether Devices are Defined to CA MIA* in this chapter.

The following steps provide an overview of the role of CA MIA in the z/OS allocation process:

1. z/OS allocation creates the EDL for a DD for the job, based on the device groups that z/OS creates during SYSGEN.
2. TPCF intercepts the EDL and marks devices ineligible if:
 - The elimination logic of the TPCEDLXT exit routine specifies that the device should be eliminated.
 - The device is reserved for another job.
 - The device is not reserved for this job and other devices are reserved FORCE=YES for this job.
 - The device is OVERGENNED and SETOPTION TPCF OVGINELIG is set to YES or JOBRESV.

If all eligible devices are eliminated from the EDL by TPCF, then TPCF returns the original EDL to z/OS allocation.

z/OS or other vendor products may also eliminate devices from allocation at this time.

3. z/OS requests the device group locks for all device groups in the EDL. GTAF makes the job wait until it determines that the locks are not in use by jobs on other systems. When the locks are available, GTAF releases the job to continue through z/OS allocation. GTAF tells the other systems that the locks are in use on this system.
4. z/OS examines the EDL and eliminates ineligible, inaccessible, allocated, and offline devices from possible allocation. Because TPCF may have varied additional devices offline (due to the preferencing status that you have assigned to these devices), z/OS may eliminate additional devices from possible allocation. These devices include OVERGENNED, NOTAVAILABLE, and EXTERNALLY DEDICATED devices.

If there are no devices that can be allocated, the job enters allocation recovery. If there is one device, then the job allocates it and then goes to Step 7. If there are multiple devices, the job goes to Step 5.

5. TPCF preferences the remaining devices based on the preferencing criteria specified. TPCF preferences devices that:
 - Are reserved for this job
 - Are dedicated to this system
 - Are selected by the TPCSRMXT exit
 - Have a preference value set

z/OS or other vendor products may also preference devices at this time.

6. z/OS chooses a device for allocation from the remaining devices based on the preferencing for the devices.
7. z/OS marks the device as allocated and the job releases the device group locks. GTAF tells the other systems which device was allocated and that the group locks are no longer in use by this system.

How z/OS Device Allocation Recovery Works

Allocation recovery is the part of allocation that a job enters when it cannot allocate a suitable online device. During allocation recovery, z/OS tries again to select a device that meets the device requirements of the job. However, z/OS considers the suitability of both currently allocated and offline devices during this part of the allocation process.

z/OS creates two types of lists from the EDL during allocation recovery:

- A list of offline or inaccessible devices. This list is known as the offline device list.
- A list of currently allocated devices for which the job can wait. These devices are called wait-eligible devices.

After creating these lists, z/OS issues messages that tell you the name of the job in allocation recovery and what options you have in getting this job out of allocation recovery. z/OS also may issue additional messages, depending on whether there are any offline devices on the offline device list and on your response to the z/OS IEF238D allocation recovery message.

Remember that z/OS serializes allocation through a locking structure for device groups. While an allocation recovery message is outstanding, z/OS allows no other jobs to select devices from the device groups on the EDL of the current job. z/OS also suspends SWAP processing while a job is in allocation recovery.

While the job is in allocation recovery, no other jobs wanting any of the same devices can proceed in allocation. When GTAF is serializing devices among systems, no jobs wanting the same devices can proceed in allocation on external systems. Therefore, a prompt response to any z/OS allocation recovery message is essential to avoid delays in allocation locally and on external systems.

The following steps provide an overview of the z/OS allocation recovery process:

1. z/OS allocation creates the offline device list and a list of wait-eligible devices for this job, based on the devices in the EDL.
2. z/OS passes the allocation recovery information to any recovery exits loaded at the IEF_SPEC_WAIT or IEF_ALLC_OFFLN exit points.

If there are no exits, then z/OS issues message IEF238D, which tells you the name of the job in allocation recovery and what options you must choose from to get this job out of allocation recovery. If there are devices on the offline device list, then z/OS issues message IEF877E before issuing the IEF238D message to show you the available offline devices. z/OS waits for you to reply to message IEF238D. z/OS suspends all SWAP processing and all jobs in allocation that need to use the device groups on the EDL of the current job. The job holds the device group locks.

If exits are loaded, then z/OS takes whatever recovery action is specified by the exits. These actions can include replying automatically to the IEF238D/IEF4433D messages, issuing the IEF238D/IEF433D WTORs, or other actions.

3. z/OS processes your response (or the response of the exit) to the IEF238D message in one of these ways:
 - If you replied CANCEL, then z/OS cancels the job.
 - If you replied with the name of an offline device, then z/OS brings this device online, if possible, and sends the current job back through the allocation process.
 - If you replied WAIT, then z/OS makes the job wait for a wait-eligible device to become available. z/OS also issues message IEF433D, described in Step 4.
 - If you replied CANCEL or entered the name of an offline device, then z/OS allocation and SWAP processing resumes and Step 4 is omitted. The job releases the device group locks.

4. z/OS issues message IEF433D whenever you reply WAIT to the z/OS IEF238D message. The IEF433D message tells you that z/OS needs additional information on what to do about other resources this job is holding.

z/OS processes your reply to the IEF433D message as follows:

- If you reply HOLD, then z/OS keeps the device group locks this job is holding. As a result, other jobs that need these device groups must wait until the current job allocates a device.
- If you reply NOHOLD, then z/OS releases the device group locks this job is holding. This lets other jobs use the device groups while the current job is waiting for a device.

How z/OS Device Allocation Recovery with CA MIA Works

The CA MIA facilities play different roles in the allocation recovery process. GTAF propagates the effect of allocation recovery to all systems, while TPCF influences device selection and automatically responds to z/OS allocation recovery messages.

z/OS suspends SWAP processing and certain device group requests while an allocation recovery message is outstanding, because the job in allocation recovery holds the device group locks until the recovery message is replied to. Because GTAF propagates this effect to other systems, you must respond promptly to all allocation recovery messages on all systems, or tell TPCF to reply automatically.

TPCF uses the values you have specified on the SETOPTION command to determine how to respond to z/OS allocation recovery messages. You should also realize that CA MIA facilities only affect allocation recovery when a managed device is in the EDL for a job.

To influence device selection, TPCF intercepts the z/OS offline device list and removes devices from it according to your preferencing criteria. TPCF obtains this criteria from the elimination logic you have defined in the TPCREEXT exit routine and from device status changes you have made through the VARY command.

Note: Either TPCF or the exit may remove the DEVICE NAME option on the z/OS IEF238D message when TPCF makes this option unavailable. For example, if TPCF removes all devices from the offline device list, then the DEVICE NAME option on message IEF238D is removed.

The following steps give you an overview of the role of CA MIA in the allocation recovery processing:

1. z/OS creates the offline device list and a list of wait-eligible devices from the EDL of that job.
2. z/OS passes the allocation recovery information to any recovery exits loaded at the IEF_SPEC_WAIT or IEF_ALL_OFFLN exit points. TPCF acts as an exit at both exit points. TPCF examines the offline device list and removes these devices from the list:
 - Devices that are overgennered
 - Devices reserved for a different job
 - Devices removed by the elimination logic in the TPCEDLXT exit or devices removed by the TPCREEXT exit

In the exit parameter list, z/OS lists the possible ways to get this job out of allocation recovery. TPCF examines these options, which are usually DEVICE NAME, WAIT, or CANCEL.

If WAIT is not an option and this step removes all devices on the offline device list, then TPCF tells z/OS to cancel the job. TPCF skips all remaining steps in this diagram.

TPCF responds to the allocation recovery according to the value you specified on the IEF238D operand on the SETOPTION AUTOREPLY command, or to the value entered for the optional TPCREEXT exit routine, which would override SETOPTION values:

- If you specified CANCEL, then TPCF removes not-available and externally dedicated devices from the offline device list. TPCF then takes one of these actions:
 - When there are no devices left on the offline device list, TPCF replies CANCEL. This tells z/OS to cancel the job, and TPCF skips all remaining steps in this diagram.
 - When at least one device remains, TPCF asks the operator to reply.

- If you specified WAIT, then TPCF does one of the following:
 - When WAIT is an option on the z/OS message, TPCF replies WAIT so that the job waits for an online device to be deallocated. TPCF then goes directly to Step 4.
 - When WAIT is not an option on the z/OS message, TPCF removes unavailable and externally dedicated devices from the offline device list unless doing so would remove all devices. If doing so would remove all devices, then TPCF lists externally dedicated or not available devices based on the SETOPTION TPCF AUTOREPLY=(NOWAIT=(EXTDEDDISP=,NOTAVLDISP=)) settings. If there are devices listed in the offline device list, then TPCF asks the operator to reply. If there are no offline devices, then TPCF replies CANCEL.
 - If you specified COND, then TPCF does one of the following:
 - When WAIT is an option on the z/OS message and TPCF removes all devices from the offline device list, TPCF replies WAIT. This makes the job wait for an online device to be deallocated. TPCF then goes directly to Step 4. If there are offline devices, then TPCF asks the operator to reply.
 - When WAIT is not an option on the z/OS message, or if TPCF does not remove all devices from the offline device list, TPCF asks the operator to reply. Externally dedicated or not available devices may be listed in the offline device list if there are no other offline devices based on the SETOPTION TPCF AUTOREPLY=(NOWAIT=(EXTDEDDISP=,NOTAVLDISP=)) settings.
 - If you specified MANUAL, then TPCF removes unavailable and externally dedicated devices from the offline device list (if possible). Externally dedicated or not available devices may be listed in the offline device list if there are no other offline devices based on the SETOPTION TPCF AUTOREPLY=(NOWAIT=(EXTDEDDISP=,NOTAVLDISP=)) settings. TPCF then asks the operator to reply.
3. TPCF issues message MIM2060 whenever it needs to ask the operator to reply. Message MIM2060 shows you the options you have in responding to the allocation recovery after TPCF has removed any additional devices from the offline candidate list. If there are offline devices to choose from, then TPCF also issues message MIM2042 to show you the names of these devices.

4. If TPCF replied WAIT for the allocation recovery, then TPCF also replies with a IEF443D option according to the value you specified on the IEF433D operand on the SETOPTION AUTOREPLY command, or to the values entered for the optional TPCREXT exit routine, which would override SETOPTION values:
 - If you specified HOLD, then TPCF replies HOLD. This tells z/OS to hold the locks to all of the device groups on the EDL of this job. As a result, other jobs needing these device groups must wait until the current job allocates a device.
 - If you specified NOHOLD, then TPCF replies NOHOLD. This tells z/OS to release the device group locks this job is holding so that other jobs can use this device group while the current job is waiting for a device.
 - If you specified MANUAL, then TPCF asks the operator for assistance.
5. When the TPCF exit point has processed the allocation recovery, the job continues through the z/OS allocation recovery process.

More information:

[How You Verify Whether Devices are Defined to CA MIA](#) (see page 38)

Product Activation and Termination Considerations

For information about starting and stopping CA MIA, see the chapter “Advanced Topics” in the *CA MIM Programming Guide*.

How You Identify Devices to CA MIA

You can place tape devices under CA MIA management. CA MIA supports all types of tape devices and four-character device names.

The device identification method you choose depends on whether you need to place only certain devices under CA MIA management (for example, you are running CA MIA on a test complex), whether you need to specifically assign global names to devices, or whether you plan to use the Autopath feature.

A global name is a unique name that all systems can use to refer to the same device. When you do not specifically assign a global name, CA MIA uses the local name of the device as its global name. The local name is obtained from the unit control block (UCB) of the device.

If the local name of a device is not the same on all systems or you have assigned the same local name to different devices on different systems, then you must specifically assign global names (through the MIMUNITS member specified on the MIMINIT DEVLIST statement) to the devices to indicate which addresses point to the same device. Otherwise, dual allocations may result.

Identify a Class of Devices

You can identify a class of devices for CA MIA management by specifying MIMINIT DEVCLASS=value.

To identify a class of devices

Specify the following command:

```
MIMINIT DEVCLASS=value
```

***value* can be one of the following:**

TAPE

This places all tape devices under CA MIA management.

NONE

Indicates that you are not placing a class of devices under CA MIA management using the MIMINIT statement. This is the default.

Example: Identify device class

To place all tape devices under CA MIA management, specify the following statement in the initialization member:

```
MIMINIT DEVCLASS=TAPE
```

Identify a List of Devices

You can identify a list of devices for CA MIA management.

To identify a list of devices

Specify the following command:

```
MIMINIT DEVLIST=value
```

value

Specifies the device list. This value can be one of the following:

MIMUNITS or *membername*

Identifies the member of the CA MIM parameter data set that contains the list of tape devices to be controlled and their associated global names. Each line of the device control member names a device address or a range of device addresses that CA MIA controls. The default is MIMUNITS.

NONE

Indicates that you are not placing a list of devices under CA MIA management.

Example: Identify device list

To specify a list of devices to be controlled by CA MIA, specify the following statement in the initialization member:

```
MIMINIT DEVLIST=MIMUNITS
```

Notes:

- You must place at least one device per system under CA MIA control otherwise CA MIM terminates during initialization. There is no maximum number of devices that CA MIA can manage.
- If you specify MIMINIT DEVLIST=NONE, then you must specify DEVCLASS=TAPE. Otherwise, CA MIA will not manage any devices and will terminate at initialization.
- If you specify MIMINIT DEVCLASS=NONE, then you must specify DEVLIST=*membername* otherwise, CA MIA will not manage any devices and will terminate at initialization.
- If are using the Autopath feature, then you must define MIMINIT DEVLIST=MIMUNITS.
- You may specify both DEVCLASS=TAPE and DEVCLIST=MIMUNITS. If a device does not have an entry in MIMUNITS, the device's global name defaults to the local name and the device will not be eligible for Autopath. If a device has an entry in MIMUNITS, the global name and Autopath eligibility are determined by the MIMIUNITS entry.

Specify Devices in the MIMUNITS Member

The MIMUNITS member specifies one or more devices that you want CA MIA to manage. The following is an example of a MIMUNITS member:

```
3C0,T3C0,7C0
0E8*
0E9*,T39*
```

Each line of the MIMUNITS member specifies one or more devices that CA MIA will manage. Using the preceding example, the following table describes the order of attributes in a line in the MIMUNITS member:

Position 1	Position 2	Position 3
3C0,	T3C0,	7C0
z/OS Device Name	Global Name of the Device	Autopath Eligibility of the Device

Position 1-z/OS Device Name

Represents the required z/OS device name. This is also known as the local name in some CA MIA commands and displays. It is *always* a three- or four-digit hexadecimal number. If three digits are specified, then a leading zero is assumed.

Position 2-Global Name of the Device

Represents the global name of the device. This value may be alphanumeric, as shown, or it may be hexadecimal. Members of the MIMplex use the global name to identify the drive in commands and displays. If you omit this information, *then the global name is made equal to the local name*. It is possible for a device to have different local names on different systems, but the global name must remain the same across all systems. If your MIMUNITS member is shared by different systems, then you can use IFSYS and ENDSYS statements to define different local names in each system.

Note: The global device name cannot be equal to the local name of a different device on any system in the MIMplex.

Position 3-Autopath Eligibility of the Device

Represents optional information about the drive to the Autopath feature. This value may be either hexadecimal or the three-letter string MIM. *If omitted, then the drive is not attached or detached by Autopath.*

This value should be specified only for z/OS systems that are running as guests under z/VM. The Autopath feature only works under z/VM. "MIM" *must* be specified when CA MIA is controlling the drive on the host z/VM system. The z/VM real address of the device, expressed in hexadecimal, is used when CA MIA is not controlling the drive on the host z/VM system. In the previous example, the device known as 7C0 to z/VM is attached to z/OS at virtual address 3C0. Autopath uses CA MIM commands to attach or detach the drive to the z/OS guest when "MIM" is specified, and uses CP commands to attach or detach the drive when a hexadecimal value is specified.

Note: Separate each attribute of the MIMUNITS member with a comma.

Examples

The following are examples of valid entries in the MIMUNITS member:

3C0

Defines z/OS device 03C0 and implies global name is 3C0

3C0,3C0

Has the same effect as the example above

3C0,5C0

Defines z/OS device 03C0 as global name 5C0

3C0,3C0,3C0

Same as the first and second examples, but also specified Autopath real address 3C0

03C0,T3C0,MIM

Defines z/OS device 03C0 as global name T3C0 and Autopath control through CA MIA

03C0,,7C0

Defines z/OS device 03C0 as global name 03C0 and Autopath real address 7C0

03C*, 03C*, 7C*

Defines z/OS devices 03C0-03CF as global names 03C0-03CF and Autopath real addresses 7C0-7CF

Note: You can specify a group of drives having similar addresses on a single line. Use an asterisk in the last character of each attribute to indicate a range of up to 16 devices. For example, 03C* would indicate that all drives from 03C0 through 03CF are included. If an asterisk is used in the first position, then it must also be used in the second and third positions, if they are present, for example, 3C*, T3C*, 7C*.

More information:

[z/VM Considerations](#) (see page 82)

Code the MIMUNITS Member

Follow the summarized coding rules below when you are coding the MIMUNITS member:

- Specify the local name of the device using the first three or four non-blank positions of the entry. Note that CA MIA ignores leading blanks.
- Use a comma as a delimiter immediately following the local name if you are assigning a global name to the device. Then, specify the global name of the device immediately following the comma.
- Use a comma as a delimiter immediately following the global name, if you are supplying Autopath information.
- Use two commas after the local name if you are not specifying a global name but are supplying Autopath information.
- Place an asterisk (*) in column 1 to indicate that a line is a comment and not an entry. You also can create comment lines by specifying the /* character string in the first two non-blank positions of a line. The entire line is treated as a comment.
- Place a slash and then an asterisk (/*) after an entry if you want to include a comment after the entry. The /* character string distinguishes the entry from the comment.
- Enclose entries in IFSYS and ENDIF statements to direct those entries to specific systems. This is applicable only when you are sharing the MIMUNITS member among systems.

The global device name cannot be equal to the local name of a different device on any system in the MIMplex.

More information:

[Direct MIMUNITS Entries to Certain Systems](#) (see page 36)

Direct MIMUNITS Entries to Certain Systems

You can share the MIMUNITS member among systems, even when you are using different local names on different systems, or when a device has the same local name as a different device on a different system. To share the MIMUNITS member under these circumstances, you need to direct entries to the appropriate systems.

To direct an entry, place an IFSYS statement before the entry and an ENDIF statement after the entry. On the IFSYS statement, specify the system name or system alias of the system (or group of systems) to which that entry applies. This tells CA MIA to use this entry only for the system specified.

For example, suppose that a device has the local name 01A0 on systems A and B and the local name 02A0 on systems C and D. To identify and assign global name G1A0 to that device, specify the following statements in the MIMUNITS member:

```
IFSYS A,B
    01A0,G1A0
ENDIF
IFSYS C,D
    02A0,G1A0
ENDIF
```

Notes:

- The letter “G” is used only for the global name examples and has no special meaning.
- For more information on IFSYS and ENDIF statements, and for information on assigning names and aliases to systems, see the *CA MIM Programming Guide*.

How You Exclude Devices from CA MIA Management

You can exclude devices from CA MIA management by coding a list of devices in the DEVEXCL CA MIA parmlib member.

To exclude specific devices during CA MIA startup, specify the following statement in your MIMINIT member:

```
MIMINIT DEVEXCL=membername
```

membername

Specifies a CA MIM parmlib member that contains a list of devices to exclude from CA MIA management. Code NONE if you do not want to exclude any devices from CA MIA> management.

Notes:

- The DEVEXCL=*membername* parameter is intended to be used with DEVCLASS=TAPE. This parameter combination causes CA MIA to discover and manage all manageable tape devices EXCEPT devices in the DEVEXCL= member.
- Using DEVEXCL=*membername* with the DEVLIST=*membername* parameter causes CA MIA to manage all devices in the DEVLIST= member EXCEPT the devices in the DEVEXCL= member.
- The same DEVEXCL=*membername* parameter was added to the CA MIA RESYNCH command. You can resynchronize CA MIA to exclude devices from being managed.
- The device exclusion list takes precedence over the MIMUNITS member. If MIMUNITS contains devices you have specified in the DEVEXCL member, they are excluded from CA MIA management.

Specify Devices in the DEVEXCL Member

The DEVEXCL member excludes devices from CA MIA management.

Example: DEVEXCL

```
*****  
*           Sample DEVEXCL Member           *  
*****  
8000      /*Device 8000  
810*     /*Devices 8100 through 810F inclusive  
9000-A000 /*Devices 9000 through A000 inclusive
```

Each line in the DEVEXCL member specifies a single device or range of devices that CA MIA will NOT manage. The syntax of the DEVEXCL member is as follows:

- One device or range of devices per statement (line).
- Valid device numbers are hex 0000-FFFF.
- Specify a range with the '-' character. For example, 1000-2000 specifies devices 1000 through 2000, inclusive.
- Specify a suffix of 0 through F with the '*' character. For example, 800* would specify devices 8000-800F, inclusive.
- Direct your entries to specific systems by enclosing those entries in IFSYS/ENDIF statements. These statements are only applicable if you share the DEVEXCL member with other systems.
- Place an asterisk (*) in column 1 to indicate that this line is a comment.
- Place a slash (/) and then an asterisk (*) after an entry if you want to include a comment after the entry.

How You Verify Whether Devices are Defined to CA MIA

You should verify the entries in the MIMUNITS member when you start CA MIM for the first time and any time you change the contents of this member. You can check the entries in these ways:

- By examining the MIM2004 messages issued on each system during the initialization process. This message lists the local and global name for each managed device on the local system.
- By using the DISPLAY LOCALUNITS command. This command lets you see the local name, global name, and status information for managed devices.
- By using the DISPLAY GLOBALUNITS command. This command lets you see the global name and status information for managed devices.

Compatibility Restrictions

You should be aware of the following compatibility issues between CA MIA and z/OS operating system functions:

- CA MIA cannot manage devices that are being managed by JES3. You can run JES3, but you cannot manage tape devices with JES3 and CA MIA. If you want CA MIA to manage devices that are currently under JES3, then you must remove the devices from JES3 management.
- CA MIA cannot manage devices that are defined to z/OS as “automatically switchable” (through HCD or the z/OS VARY AS=ON command). If you want CA MIA to manage devices currently defined as automatically switchable, then you must issue z/OS VARY AS=OFF commands for those devices and update the HCD definition.
- There is a potential for cross-system deadlocks to occur on device group locks if a user has specified (through HCD or the z/OS CP command) mismatching search orders for z/OS generic device groups containing CA MIA managed devices. We recommend using identical z/OS generic device group search ordering on all systems in the CA MIA complex.

How You Dynamically Change Which Devices Are Managed

You can dynamically change which devices CA MIA manages. This process is called resynchronization. CA MIA resynchronizes automatically when a Hardware Configuration Definition (HCD) ACTIVATE is performed or on demand when the RESYNCH command is issued.

Resynchronization: RESYNCH Command

The DEVLIST, DEVCLASS, DEVEXCL and COMMANDS parameters on the RESYNCH command, the SETOPTION RESYNCH command, and the MIMINIT statement control the actions of CA MIA during device resynchronization.

If the DEVLIST, DEVCLASS, DEVEXCL and COMMANDS parameters are not specified on the RESYNCH command, then CA MIA uses the DEVLIST, DEVCLASS, and COMMANDS settings on the SETOPTION RESYNCH command.

For example, to issue a RESYNCH command that dynamically updates the managed device list for CA MIA and places all tape devices under CA MIA control, issue the following command:

```
RESYNCH DEVCLASS=TAPE
```

In this example, CA MIA uses TAPE for the DEVCLASS option, which temporarily overrides the value set on the SETOPTION RESYNCH command for DEVCLASS. The values for the COMMANDS and DEVLIST options remain the same.

Notes:

- When a task is in the tape allocation process and a resynchronization occurs that removes all the devices requested by the task from CA MIA management, CA MIA discontinues the cross-system serialization of the allocation. However, CA MIA may continue its involvement in some parts of the allocation until the allocation completes. For example, if the task should go into allocation recovery, then CA MIA manages the recovery until the allocation recovery messages are responded to.
- The RESYNCH command is a local command. That is, it only takes effect on the system where it is issued. If the devices managed by CA MIA are to be changed on more than one system, then the RESYNCH command must be issued on each system that requires the change.
- The SETOPTION RESYNCH SAMEDEVS parameter does not apply to resynchronizations performed due to the RESYNCH command. SAMEDEVS is only valid for automatic resynchronizations through an HCD ACTIVATE.
- Any RESYNCH option, except the SAMEDEVS option, can be overridden dynamically when using the RESYNCH command.
- The RESYNCH parameters are available on both the SETOPTION TPCF and SETOPTION GTAF commands. Setting the RESYNCH parameters with one of these SETOPTION commands automatically sets the same values for the other SETOPTION command.
- It is possible to RESYNCH a device out of CA MIA control and have the device still exist on the system. These are the devices that CA MIA will schedule OFFLINE.

If a device is RESYNCHed out of CA MIA control due to an HCD ACTIVATE configuration change, the device is being deleted by z/OS. To be deleted, the device must already be OFFLINE, and it will not exist after the ACTIVATE completes; the CA MIA OFFLINE processing is not relevant in this case.

Note: For more information, see the *CA MIM Statement and Command Reference Guide*.

Resynchronization: HCD ACTIVATE

When an HCD ACTIVATE is performed, CA MIA automatically performs resynchronization that is based on the values on the DEVCLASS, DEVLIST, DEVEXCL, COMMANDS, and SAMEDEVS parameters on the SETOPTION RESYNCH command. If no values are specified on the SETOPTION RESYNCH command, then the values on the MIMINIT DEVCLASS, DEVLIST, and COMMANDS parameters are used.

Notes:

- When a task is in the tape allocation process and a resynchronization occurs that removes all the devices requested by the task from CA MIA management, CA MIA discontinues the cross-system serialization of the allocation. However, CA MIA can continue its involvement in some parts of the allocation until the allocation completes. For example, if the task goes into allocation recovery, then CA MIA manages the recovery until the allocation recovery messages are responded to.
- If SETOPTION RESYNCH SAMEDEVS=NO, then during resynchronization CA MIA uses the DEVCLASS and DEVLIST options to determine what devices to manage. CA MIA recognizes any devices added by the HCD or the z/OS ACTIVATE command. NO is the default setting.
- If SETOPTION RESYNCH SAMEDEVS=YES, then regardless of what occurs in the z/OS I/O reconfiguration, CA MIA manages the same devices it was managing prior to reconfiguration. The DEVCLASS and DEVLIST options are ignored.
- The RESYNCH parameters are available on both the SETOPTION TPCF and SETOPTION GTAF commands. Setting the RESYNCH parameters with one of these SETOPTION commands automatically sets the same values for the other SETOPTION command.
- Use SETOPTION RESYNCH DEVEXCL=*membername* to specify an exclude list member. When CA MIA enters the resynchronization process as a result of a z/OS ACTIVATE, the DEVEXCL member that is specified on the SETOPTION RESYNCH command is read and processed. Before issuing an MVS ACTIVATE command, issue the CA MIM DISPLAY OPTIONS command to verify your current DEVEXCL member value.

Note: For more information, see the *CA MIM Statement and Command Reference Guide*.

z/OS Hardware Configuration Definition (HCD) ACTIVATE Considerations

I/O configurations are used to define hardware resources available to an operating system, as well as the connections between these resources. A hardware configuration provides both physical and logical information about the resources provided.

Overview of HCD ACTIVATE

The tape devices managed by CA MIA are a subset of these hardware resources. I/O configurations must be defined to both the operating system software as well as the channel subsystem (hardware). Both the hardware and software I/O configuration processes are consolidated in a single interactive end-user interface called the Hardware Configuration Definition (HCD) component of z/OS.

The output from HCD is an I/O Definition File (IODF), which contains I/O configuration data. An IODF can define multiple hardware and software configurations to the operating system. When an IODF is ACTIVATED, HCD actually defines the I/O configuration to the channel subsystem, the operating system, or both. With the HCD ACTIVATE function or the z/OS ACTIVATE operator command, changes can be made to the current configuration without the need to initial program load (IPL) the software or power-on reset (POR) the hardware. Making configuration changes while the system is running is known as dynamic configuration or dynamic reconfiguration.

How You Change Hardware and Software Definitions

When an I/O configuration is dynamically changed, the definitions to both hardware and software *or* to software only, can be changed. A software-only change modifies only software control structures, such as the unit control blocks (UCBs) and the eligible device table (EDT). A hardware and software change modifies both hardware and software control structures. In most cases, simultaneous dynamic configuration changes to both hardware and software configuration definitions are made.

A dynamic configuration change causes z/OS to activate a new EDT definition. Under normal operating circumstances, z/OS uses one eligible device table (EDT) to process allocation requests. As a result of dynamically changing the software I/O configuration definition of a system, z/OS must use two EDTs to process the change: a primary and secondary EDT.

The primary EDT processes all current and new allocation requests. z/OS usually runs with only a primary EDT, until a dynamic configuration change is initiated. z/OS then activates a new primary EDT for the new I/O configuration, making the old primary EDT the secondary EDT.

All new allocation requests, from the ACTIVATION point forward, use the new primary EDT. The secondary EDT receives no new allocation requests. z/OS deletes the secondary EDT when the allocation requests issued before the dynamic configuration change (all requests currently using the old EDT) complete.

Notes:

- Allocation does not stop because an ACTIVATE of a new IODF is in progress. New allocations use the new primary EDT built by the ACTIVATE; allocation requests initiated before the ACTIVATE of the new IODF use the secondary EDT.
- The secondary EDT cannot be deleted until all allocation requests using it have completed. Even though an IOS500I ACTIVATE RESULTS message is issued indicating that the ACTIVATE COMPLETED SUCCESSFULLY, until the IOS501I ACTIVATE CLEANUP COMPLETE message is issued, IOS holds an exclusive enqueue for SYSZIOS/Dynamic.

A subsequent ACTIVATE, initiated on the same system before the IOS501I ACTIVATE CLEANUP COMPLETE message is issued, may fail because residual allocation requests are still bound to the secondary EDT.

HCD ACTIVATE

The CA MIA GTAF facility determines what devices to manage based on the MIMINIT DEVCLASS and DEVLIST initialization parameters.

When a new I/O definition (IODF) is activated, which, in turn constructs a new EDT, it can define additional devices or remove existing devices from the systems and the control of CA MIA.

The CA MIA TPCF facility influences device selection during the z/OS allocation process. TPCF performs device preferencing based on user specified criteria enacted upon the list of devices eligible for a specific allocation request. z/OS creates this list of eligible devices (EDL) from the EDT.

Therefore, both facilities of CA MIA rely on the availability of the EDT and must detect and react appropriately to any change in the EDT.

To achieve this, CA MIA constantly evaluates the location of the current EDT by interrogating parameter lists at key CA MIA intercept points in the z/OS allocation process. In addition, CA MIA performs additional EDT verification as part of every control file cycle. When CA MIA detects a reconfiguration, it automatically resynchronizes in response.

CA MIA holds a bind or lock on the storage occupied by the EDT. Each control file cycle performed by CA MIA releases and re-achieves this bind. This process is also repeated as part of CA MIA processing in response to a z/OS ACTIVATE command or the CA MIA RESYNCH command.

It is normal for IOS Configuration displays to show CA MIA appearing to hold a continual bind on the EDT at all times. The bind that CA MIA maintains does not delay an ACTIVATE from completing.

As discussed in previously, during a dynamic configuration change, z/OS activates a new EDT definition.

CA MIA, like z/OS, processes all current and new allocation requests using the new primary EDT. Both z/OS and CA MIA use the secondary EDT to process allocation and allocation recovery requests issued before the dynamic configuration change completed. z/OS deletes the secondary EDT when the allocation requests, issued before the dynamic configuration, complete.

The following is an example of a D IOS,CONFIG(EDT) display showing binds on both the primary and secondary EDT:

```
D IOS, CONFIG(EDT)
IOS506I 11.56.09 I/O CONFIG DATA 514
ELIGIBLE DEVICE TABLE LATCH COUNTS
1 OUTSTANDING BINDS ON PRIMARY EDT
ASID = 0023 JOBNAME = MIMGR
1 OUTSTANDING BINDS ON SECONDARY EDT
ASID = 0028 JOBNAME=TAPE1
```

In this CA MIA address space example, MIMGR holds a bind on the primary EDT. Job name TAPE1, which started before the ACTIVATE initiated, went into recovery and has not yet completed.

TAPE1 holds a bind on the secondary EDT, which was the primary EDT when this job went through allocation. As soon as TAPE1 completes, the secondary EDT is deleted and an IOS501I ACTIVATE CLEANUP COMPLETE is issued.

More information:

[Resynchronization: HCD ACTIVATE](#) (see page 41)

CA MIA Serialization of HCD ACTIVATE

CA MIA serializes HCD ACTIVATE across all of the active systems in an MIAplex. When CA MIA detects that an HCD ACTIVATE is underway on the local system, it raises an exclusive ENQ for SYSZIOS/DYNAMIC on all active, external systems in the MIAplex. CA MIA does so to ensure that potential changes to the list of devices managed by CA MIA, affected by the ACTIVATE of a new IODF, only occur on one system in the MIAplex at a time.

Until the IOS501I ACTIVATE CLEANUP COMPLETE is issued for an ACTIVATE being performed on an external system, IOS holds an exclusive ENQ for SYSZIOS/DYNAMIC. As a result, CA MIA will not be able to serialize a local ACTIVATE with the rest of the MIAplex, until an IOS501I ACTIVATE CLEANUP COMPLETE is issued for the ACTIVATE in progress on the external system and IOS drops its exclusive ENQ for SYSZIOS/DYNAMIC there.

CA MIA issues repeated MIM2161W SERIALIZATION OF SYSZIOS/DYNAMIC REQUESTED messages on the local system and will delay a local HCD ACTIVATE until it can serialize the local ACTIVATE with the rest of the MIAplex by obtaining a SYSZIOS/DYNAMIC ENQ EXCL on all external systems.

Specify Same z/OS Preference Order of Generic Device Groups on EDL

If different z/OS generic device group search orders (or preference values) have been specified for generic device groups containing CA MIA managed devices on systems in the CA MIM complex, then the potential exists for a cross-system deadlock on generic device groups. Values for generic device group preference are specified in the Hardware Configuration Definition.

Specifying the same z/OS preference order on all the systems defined to the MIMplex prevents this type of lockout. Generic device groups are acquired one at a time, but must all be acquired prior to proceeding into allocation.

For example, imagine that there is a two-system MIMplex, comprised of SYSA and SYSB. SYSA and SYSB preference the generic device groups in reverse order (SYSA preferences in the order 3480, 3490; and SYSB preferences the 3490 generic first followed by the 3480 generic). If an allocation request occurs simultaneously on both systems, then SYSA first acquires generic device group 3480 per its preference order. SYSB acquires 3490 per its preference order. Each system then attempts to acquire the generic device group already acquired by the other system, resulting in a deadlock.

Specifying the same z/OS preferencing order of generic device groups on the EDL prevents this type of lockout from occurring.

Configure Tape Devices Offline until CA MIA Synchronization

When sharing devices between multiple systems, there is an integrity exposure while devices are online to more than one system and CA MIA is not running. To avoid this, configure tape devices to be offline at IPL time, start CA MIA early in the IPL process to ensure that the product initializes and is fully functional when allocation begins on a system, and bring the tape devices online with VARY commands in the MIMSYNCH member.

To prevent devices from coming online automatically during IPL, we strongly recommend that you specify OFFLINE=YES on the IODEVICE macro used as input to the IOCP or MVSCP for devices to be shared. This tells z/OS to not allow devices to come online during the IPL.

If you are using the Hardware Configuration Dialog, then we strongly recommend that you specify YES for the OFFLINE parameter for the same reason.

The Hardware Configuration Dialog Define Device Parameters/Features panel provides the ability to configure devices as initially offline. The following is an example of this panel:

```

----- Define Device Parameters / Features -----
                                                    Row 1 of 3
Command ==> _____ Scroll ==> PAGE

Specify or revise the values below.

Configuration ID . : CONFIG01           OS/390 systems
Device number   . : 074A               Number of devices : 1
Device type    . : Dummy

Parameter/
Feature          Value  P Req. Description
OFFLINE          Yes    Device online or offline
DYNAMIC          Yes    Device supports dyn config
LOCANY           Yes    UCB can reside in 31 bit

*****Bottom of data*****

```

To have CA MIA automatically bring devices online following system synchronization, you can place VARY ONLINE commands in the MIMSYNCH member of the MIMPARMS data set.

You can use the z/OS VARY ONLINE command or the CA MIA VARY ONLINE command, prefaced by z/OS MODIFY or the CA MIM command prefix character.

Note: For more information, see the *CA MIM Programming Guide*.

To prevent needless multiple executions of the VARY ONLINE command on any system, specify a scope of LOCAL on any VARY command that is to be executed on multiple systems. For example, specify a scope of LOCAL on VARY commands issued from a MIMSYNCH member shared by multiple systems.

VARY Device Serialization

Whenever a tape device is shared among multiple systems, tape data corruption can occur if online processing for the device is occurring on one system while a job on another system is using the device. For example, Automatic Volume Recognition (AVR) during online processing may rewind a tape that is in use on another system. AVR is done as part of z/OS online processing. If a tape is found on a device during online processing, then AVR rewinds the tape and reads the label to find the VOLSER for the tape. If a VARY ONLINE is issued for a device on one system while the device is in use by a job on another system, then AVR during online processing may rewind the tape that is in use by the job on the other system, thus causing tape data corruption.

CA MIA VARY device serialization prevents the exposure outlined above. Device OFFLINE requests are managed by CA MIA as well to ensure the order of ONLINE and OFFLINE requests so that the resulting device states from such requests are set to the desired condition.

The CA MIA VARY device serialization provides granular device level serialization at online processing time. When CA MIA intercepts online processing locally for a device (VARY command, programmatic IEEVARYD call), it only allows online processing to occur if the device is not allocated or is not in the process of being allocated by a job or a SWAP on an external system. Conversely, while online processing is occurring locally for a device, CA MIA on external systems prevents jobs and SWAPs from allocating or going into the process of allocating the device. The device serialization is extended to be in effect throughout the entire online process. As mentioned, CA MIA controls VARY OFFLINE processing as well. This maintains the respective order of online and offline requests on the local system. Offline processing does not require the device level serialization, as does online processing; therefore, offline requests on the local system are not affected by any activity on an external system. However, an offline request might be forced to wait behind an outstanding local online request.

CA MIA cross-system VARY device serialization is performed through cross-system device group lock serialization. This aligns CA MIA VARY device serialization with tape allocation and SWAP serialization methods that also use cross-system device group lock serialization. By aligning device allocation, device swap, and device VARY to use the granular device level serialization technique, the performance and operational impacts of the VARY device serialization are minimized.

VARY Command Serialization

CA MIA VARY Command Serialization is a software technique designed to maintain tape data integrity during VARY ONLINE Command processing in multi-system tape sharing environments.

CA MIA VARY Command Serialization maintains tape data integrity by:

- Intercepting VARY ONLINE and OFFLINE commands
- Intercepting programmatic calls to IEEVARYD
- Allowing VARY commands and IEEVARYD calls for managed devices to execute when tape data integrity will not be compromised.

CA MIA ONLY allows VARY ONLINE processing for a managed device to occur if the device is:

- Not allocated or in the process of being allocated
- Not the target of a DDR SWAP operation
- Not the target of an UNLOAD command

If VARY ONLINE processing is in progress for a managed device on the local system, CA MIA prevents:

- The device from being allocated on external systems
- The device from being swapped on external systems
- The device from being unloaded on external systems

CA MIA preserves the order of VARY ONLINE and OFFLINE requests to ensure that the device state resulting from a sequence of commands is correct. VARY OFFLINE requests do not require serialization.

CA MIA VARY device serialization software employs z/OS device group locking to maintain tape data integrity on all active systems in a MIMplex. CA MIA Tape Device Allocation, UNLOAD and SWAP serialization also use this cross-system device group locking technique. CA MIA global alignment of VARY, Allocation, UNLOAD and SWAP processing to use this granular, device level serialization minimizes the performance and operational impacts of VARY device serialization.

When a sequence of VARY commands is issued for a device, CA MIA serialization begins when the first VARY ONLINE command is processed. Serialization is maintained until all intercepted VARY ONLINE/OFFLINE commands for the device have been processed.

The operational characteristics associated with the CA MIA enhanced VARY device serialization are discussed in the following sections.

Overview of Managed Device VARY Processing

CA MIA intercepts requests to VARY a CA MIA managed device ONLINE or OFFLINE. If a VARY request originated from an intercepted z/OS VARY command, then an acknowledgment is directed back to the origin console. The VARY device request is then sent to the CA MIA address space for processing. Multiple requests for the same device are maintained in first in, first out order so that the proper online and offline processing sequence is maintained.

CA MIA can optionally delete redundant VARY commands when a sequence of duplicate VARY commands is intercepted. By default, this feature is disabled. The SETOPTION VARYDEDUP command is used to enable the deletion of duplicate VARY commands.

CA MIA attaches up to eight concurrent VARY service subtasks to perform VARY processing. Additional incoming VARY requests remain queued by CA MIA until one of the eight VARY subtasks completes and is detached.

When CA MIA intercepts a VARY device request, CA MIA determines if there are VARY requests for the device already being processed. If there are no active VARY requests for the device, CA MIA attaches a service subtask to process the request. If there are active VARY requests for the device, CA MIA queues the incoming request to the existing subtask processing VARY requests for the device. A given subtask is only responsible for servicing VARY requests for a single device.

When a VARY service subtask is excessively delayed in obtaining the respective device group lock, the service task times out. Timed out VARY requests are re-queued. Timeout processing ensures that a VARY request that cannot obtain device group locks will not excessively delay other requests.

CA MIA VARY Device Service subtasks call the IBM IEEVARYD service routine to process VARY device requests. The results of VARY device requests are logged in SYSLOG by the IEEVARYD service routine regardless of the command origin.

Delays in Tape Allocation, UNLOAD, SWAP, and Online Processing

VARY ONLINE processing competes with tape device allocations, UNLOADs, and SWAPs for device group locks for CA MIA managed devices. These processes on the local system delay VARY ONLINE processing on external systems. For example, assume a job in tape device allocation on a local system holds the device group locks for a set of devices. VARY ONLINE commands issued on external systems for these devices must wait until device group locks are released. Similarly, a local VARY ONLINE delays external systems tape allocation until VARY processing finishes for the device.

The CA MIA DIAGNOSE command provides information about the device group lock usage for each system. Use this command to display information about delays in tape device allocation, SWAP, UNLOAD, and VARY ONLINE processing.

VARY ONLINE cross-system lock requests time out if they are not satisfied within the interval specified by SETOPTION VARYDELAY. The default VARY Delay timeout value is 30 seconds. When the timeout occurs, the request is re-queued, allowing the next VARY ONLINE request in the queue to enter cross-system lock processing.

Allocation Recovery and Online Processing

When a job goes into allocation recovery and an allocation recovery message is outstanding waiting for a reply (DEVICE NAME, WAIT, or CANCEL), the job holds device group locks for the devices that it can allocate. When an allocation recovery message is outstanding on the local system, tape allocations on external systems that require the same locks are delayed until the allocation recovery message is replied to. Jobs in allocation recovery on one system also delay any VARY ONLINE commands on external systems for devices for which the locks are held by the allocation recovery. The VARY ONLINE commands on the external systems are delayed until the local allocation recovery message is replied to. If allocation recovery messages are not replied to promptly, then cross-system delays in VARY ONLINE processing as well as in tape allocation processing can result.

Define CA MIA in the Subsystem Name Table

The order in which tape subsystems process subsystem interface (SSI) events is important, especially in tape allocation environments where multiple tape subsystems, such as CA MIA and Sun STK's SMC and HSC, need to process the same tape device oriented events.

In an environment in which CA MIA is to coexist with the robotic tape drive software of other vendors, CA MIA should process tape-related SSI events last. Processing each event last allows CA MIA to detect and react appropriately to actions taken by other tape subsystems that have also processed the event.

Proper placement of the CA MIA definition in the IEFSSNxx member of SYS1.PARMLIB ensures that SSI tape device-oriented events are processed in the proper order by each subsystem.

To establish the order in which tape device-oriented events are processed

1. In the MIMINIT member of the MIMPARMS data set, specify MIMINIT SUBNAME=MIMA for your CA MIA started task.
2. In SYS1.PARMLIB(IEFSSNxx), using subsystem name format, CA and Sun STK recommend that you define your tape management subsystems in the following order:
 - a. TMS (CA 1)
 - b. SMC0 (Sun STK-SMC)
 - c. SLS0 (Sun STK-HSC)
 - d. MIMA (CA MIA)
3. Within SMC, make sure that MIACOMPAT(ON) is specified. Issue the following command on each system to verify that the correct value is in place:

```
F SMC,ALLOCDEF LIST
```
4. Restart CA MIA, Sun STK-SMC, and Sun STK-HSC.
5. Issue the following z/OS command on every MIAplex system to verify that your tape management subsystems are defined in the proper order:

```
DISPLAY SSI
```

Note: For more information, see the chapter “Advance Topics” in the *CA MIM Programming Guide*.

How You Change the Status of Managed Devices (VARY)

You can use the z/OS or CA MIA VARY command to vary devices online or offline through the MIMSYNCH member of the CA MIM parameter data set or from a console. If a special CA MIA status (such as NOT AVAILABLE, AVAILABLE, OVERGENNED, NOT OVERGENNED, JOB RESERVED, or EXTERNALLY DEDICATED) is to be given to a device, you must use a CA MIA VARY command because the z/OS VARY command does not support these statuses. The same is true when using a scope value on the VARY command such as GLOBAL, LOCAL, system ID, or EXTERNAL. Use the command prefix or z/OS MODIFY command when issuing CA MIA VARY commands from MIMSYNCH or a console.

VARY Devices OFFLINE and ONLINE

The CA MIA VARY command enables you to change the status of devices that are being managed by the CA MIA on one or more systems in your MIAPlex.

Examples:

1. To VARY device 01A0 offline to all systems, issue the following command:

```
VARY 01A0 OFFLINE GLOBAL
```

2. To VARY this device offline to all systems except the local system, issue the following command:

```
VARY 01A0 OFFLINE EXTERNAL
```

3. To VARY devices 01A0 and 01A1 online on the local system, issue the following command:

```
VARY (01A0,01A1) ONLINE LOCAL
```

4. To VARY a range of devices (including devices 01A0, 01A1, 01A2, 01A3, and 01A4) online on a specific system, issue the following command:

```
VARY (01A0-01A4) ONLINE systemid
```

The *GLOBAL*, *EXTERNAL*, *LOCAL* and *systemid* values determine the scope of the VARY command. If you specify the *ONLINE* or *OFFLINE* parameters without specifying a scope value, GTAF uses the value set for the *VARYSCOPE* parameter on the *SETOPTION* command. The default for *VARYSCOPE* is *LOCAL*.

The *systemid* variable represents the system name or alias you have assigned to the system through the *DEFSYS* statement or the one- or two-digit index number of the system.

Note: The devices specified on a VARY RANGE command are presented to the operating system for processing as individual VARY device requests in *ASCENDING* device number order, regardless of the order in which device number ranges are specified on the VARY RANGE command.

PURGE Pending VARY Requests

The PURGE parameter of the CA MIA VARY command forces pending VARY requests for a device or range of devices to be discarded. Use the DIAGNOSE VARY command to identify pending VARY requests. The PURGE parameter is useful for eliminating redundant VARY requests that can cause allocation delays.

Note: The PURGE parameter only supports a scope of LOCAL.

Examples:

1. To purge pending VARY requests for a single device on the local system, issue the following command:

```
VARY 01A0,PURGE
```

2. To purge pending VARY requests for devices 01A0 and 01A1 on the local system, issue the following command:

```
VARY (01A0,01A1),PURGE
```

3. To purge pending VARY requests for a range of devices(including devices 01A0, 01A1, 01A2, 01A3, and 01A4) on the local system, issue the following command:

```
VARY (01A0-01A4),PURGE
```

How You Make a Device Unavailable for Allocation

You can use the NOTAVAILABLE or the OVERGENNED parameter on the CA MIA VARY command to make a device unavailable for allocation. When you do this, TPCF varies the device offline.

NOTAVAILABLE Parameter

When the NOTAVAILABLE parameter is used, the device that was varied offline and NOTAVL can only be selected for allocation if the AUTOREPLY and IEF238D settings generate message MIM2060 during allocation recovery, allowing the operator to reply with that device, or if you issue the VARY AVAILABLE command for that device. TPCF intercepts and suppresses all VARY ONLINE commands for NOTAVAILABLE devices so that these devices are not varied online accidentally.

Note: For information about how these devices are handled during allocation recovery, see Using SETOPTION AUTOREPLY to Control Recovery Processing in this chapter.

Examples:

- To make device 01A0 unavailable for allocation on the local system, issue this command:

```
VARY 01A0 NOTAVAILABLE LOCAL
```
- To make this device unavailable for allocation on all systems, issue this command:

```
VARY 01A0 NOTAVAILABLE GLOBAL
```
- To make the device unavailable on all systems except the local system, issue this command:

```
VARY 01A0 NOTAVAILABLE EXTERNAL
```
- To make this device unavailable only on system SYS1, issue this command:

```
VARY 01A0 NOTAVAILABLE SYS1
```
- To make this device available and online to the local system, issue this command:

```
VARY 01A0 AVAILABLE LOCAL
```

OVERGENNED Parameter

There are two main differences between the OVERGENNED status and the NOTAVAILABLE status. The first difference lies in the way the devices are treated in allocation recovery. OVERGENNED devices are always eliminated from the offline device list in allocation recovery and cannot be selected by replying to an allocation recovery message. OVERGENNED devices can be made available for allocation again only by issuing the VARY NOTOVG command. NOTAVAILABLE devices are eliminated conditionally from the offline device list in allocation recovery based on the setting of the SETOPTION TPCF AUTOREPLY=(NOWAIT...) options. NOTAVAILABLE devices can be made available for allocation through replying with them to allocation recovery messages or issuing the VARY AVL command.

The second difference lies in the eligibility of the device. OVERGENNED devices may be optionally marked ineligible in the EDL based on the SETOPTION TPCF OVGINELIG parameter settings. NOTAVAILABLE devices are not marked ineligible in the EDL.

The status of OVERGENNED is usually given to devices that are to be unavailable for allocation for long periods of time. For example, the status of OVG should be used for addresses that are defined as tape device addresses but do not have tape devices attached. The status of NOTAVAILABLE is usually given to devices that are to be temporarily unavailable for allocation. For example, the status of NOTAVL should be used for devices that are temporarily unavailable due to maintenance.

By using the SETOPTION TPCF OVGINELIG={NO|YES|JOBRESV} command, you can dynamically control whether OVG devices are marked ineligible in the EDL. Marking OVG devices ineligible in the EDL makes them unavailable for allocation early in the allocation process. This affects the way a job behaves in the device allocation process and can eliminate jobs going into allocation recovery and being canceled during allocation recovery. Note that OVG devices are always removed from the offline device list in allocation recovery regardless of the setting of the OVGINELIG parameter.

Examples: OVERGENNED Parameter

- To make device 01A0 OVERGENNED on the local system, issue the following command:
VARY 01A0 OVG LOCAL
- To make this device OVERGENNED on all systems, issue the following command:
VARY 01A0 OVG GLOBAL
- To make this device available again on all systems, issue the following command:
VARY 01A0 NOTOVG GLOBAL

More information:

[How You Influence Device Selection](#) (see page 59)

Force a Device Offline while Allocated to Another System

When a device is externally allocated, CA MIA plugs the device on the local system. The plug makes the device appear allocated locally so it cannot be allocated by a local system. While a device appears allocated, any attempt to VARY the device offline (VARY OFFLINE, VARY NOTAVL, VARY OVG, and so on) puts the device in a pending offline status. In other words, the device does not immediately go offline. To force a device offline that is in use on another system, use the VARY NOTAVAILABLE FORCE command. If the device is allocated externally, then it is taken offline locally and given a status of not available. If the device is online but not allocated, then it is varied offline and is also given a status of not available.

To force device to be unavailable on the local system

Issue the following command:

```
VARY devicename NOTAVAILABLE FORCE
```

devicename

Specifies the name of the device you want to make unavailable

You can use the VARY AVAILABLE command to bring this device back online.

Example:

Suppose you have an HCD ACTIVATE scenario in which the devices allocated on systems external to the activating system are to be modified or deleted on the local system by the ACTIVATE command. When you issue the ACTIVATE command, the system ensures that the devices to be deleted or modified are offline and deallocated. If the devices are allocated on an external system, then ACTIVATE will fail, because the devices will be plugged by CA MIA on the local system. Use of the VARY NOTAVAILABLE FORCE command allows devices to be taken offline even if they are allocated externally, thus allowing ACTIVATE to complete.

Note: This command should only be used in situations where externally allocated devices must be taken offline immediately. Typically, VARY NOTAVL without the FORCE should be used.

Wake Jobs Waiting for Devices

The VARY AVAILABLE command releases a not-available device by VARYing that device online and making it available for allocation. In addition, an operator may use the command to wake jobs waiting for a device on the z/OS allocation queues.

For example, two jobs have gone through allocation recovery and an operator has responded WAIT to the IEF238D messages and NOHOLD to the IEF433D messages. If the operator issues a VARY AVAILABLE command to any device that could satisfy both job requests, then the command wakes both jobs and sends them through allocation again.

You can specify the LOCAL, EXTERNAL, GLOBAL, or *systemid* parameter to indicate the scope of the status change.

Note: For more information, see the *CA MIM Statement and Command Reference Guide*.

Wake (POST) Jobs Waiting for Devices

A VARY ONLINE command for a device will wake up (POST) all local jobs waiting for the device on the z/OS allocation queues. If a VARY ONLINE command is deferred due to the external allocation of the target device, local jobs will not be posted until the external allocation completes and the local VARY ONLINE executes successfully. If a device has an MIA status of NOTAVAILABLE, use the MIA VARY AVAILABLE command to update the device's status to AVAILABLE and drive online processing.

Example:

Assume two jobs go through allocation recovery. An operator responds WAIT/NOHOLD on behalf of both jobs. The operator then issues a VARY ONLINE command to a device that is not externally allocated and is present on Eligible Device List (EDL) for both jobs. Both jobs will be POSTed out of their WAITs and will re-enter allocation.

Activate the Automatic Cartridge Loader

ACL processing refers to the special processing z/OS performs for assignable devices in which the automatic cartridge loader (ACL) feature is installed. When a job needs one of these devices, z/OS determines which devices can satisfy the mount request by looking at the type of request and applying preference logic to devices.

Use the VARY ACTIVE command to change the ACL status of an assignable device to active on all systems. When the ACL status is active, the device is preferred for non-specific, non-temporary requests, and not preferred for specific, temporary requests.

Use the VARY NOTACTIVE command to change the ACL status to inactive so the device is preferred for specific, temporary requests and not preferred for non-specific, non-temporary requests.

For example, to change the ACL status for the device 01A0, issue this command:

```
VARY 01A0 ACTIVE
```

Note: Issuing a V 01A0 ACT command for an inactive ACL only sets the status to active until z/OS determines that it should once again be inactive. For example, when an ACL tape input hopper becomes empty, it is automatically marked inactive by z/OS.

Use the MIM2032 parameter on the SETOPTION command to determine whether TPCF monitors and reports changes in the active status on the local system for an assignable device with the ACL feature activated. The active or inactive ACL status of a device affects how z/OS preferences assignable devices. Initiated by the MIM2032 message to vary the ACL status of a device to active or inactive, use of an automated rule can influence z/OS device selection, to or away from a device, based upon the type of allocation request.

More information:

[z/OS ACL Preferencing](#) (see page 70)

Respond to ASSIGN Processing Requests

Note: Any references in the following section to “assignable” tape devices include the following: 3480, 3490, 3590, and any other type of compatible device.

When you bring an assignable tape device online, z/OS issues an ASSIGN request for that device. The ASSIGN request indicates which system or group of systems can use that device. By default, z/OS assigns the device exclusively to the current system. You also can allow multiple systems to assign the device by issuing the z/OS command VARY ONLINE,SHR. This enables any system that provides the correct password to use the device.

You can tell GTAF to intercept ASSIGN requests for GTAF managed devices by using the GTAINIT ASSIGN statement.

Note: For more information, see the *CA MIM Statement and Command Reference Guide*.

When you specify MULTISYSTEM, you must use the same value for ASSIGN on all systems that will be sharing the devices. For example, if you want systems A and B to share a group of devices (by providing the password “NON-JES”), you must specify the following for both systems:

```
GTAINIT ASSIGN=(MULTISYSTEM, 'NON-JES')
```

Use of different assign passwords in a multiple MIMplex environment is beneficial when both MIMplexes have access to the same devices. Devices brought online to a system in one MIMplex can only be brought online to other systems in the same MIMplex that supply the correct password. In other words, a device can only be online to one MIMplex at a time. Using the same assign password in two separate MIMplexes causes dual allocations if a device is brought online to both MIMplexes.

Important! All assignable devices must be offline to all systems before you start CA MIM if you are changing the value of the ASSIGN parameter since the last time GTAF was started.

How You Influence Device Selection

You can influence device selection in a number of different ways. The following sections provide you with information on the different methods you can use to influence device selection:

- Assigning a preference value to a device
- Dedicating a device to a system
- Reserving a device for a job or group of jobs
- Releasing a job reserved device
- Forcing a job to use a reserved device
- Limiting access to devices through job reserve
- Limiting access to devices through the OVERGENNED status
- Using exit routines to influence device selection

Assign a Preference Value to a Device

You can tell TPCF to assign a preference value to a device using the PREFERENCE parameter on the VARY command. The *higher* the value you specify on this parameter, the more preferred the device. You can specify a value from 1 to 255, or you can specify NONE.

For example, to make local jobs prefer device 01A0 to other devices in the EDL, assign a high preference value to this device by issuing this command:

```
VARY 01A0 PREFERENCE=100
```

Devices with preference values are preferred over devices without preference values; therefore this device would be preferred whenever possible. However, if you then assigned the value 150 to device 01A2, then 01A2 would be preferred over device 01A0 if both of these devices were available.

You can assign the same preference value to several devices, creating a preference group. You also can remove preference values by specifying PREF=NONE on the VARY command.

Note: Preferencing is only effective for devices that are eligible for the current allocation. That is, preferencing placed on devices that have been eliminated from allocation eligibility by z/OS or vendor products (robotic software, CA MIA job reserve, and so on) is ineffective since these devices cannot be used for the allocation. Similarly, preferencing placed on offline and allocated devices is ineffective since these devices cannot be used for the allocation. Also, preference values are used only as 'tie breakers' when all other types of preferencing for devices are equal. For these reasons, device preferencing may not produce the results you expected.

Note: For information on diagnosing allocations that do not follow your preferencing specifications, see the chapter “Troubleshooting.”

More information:

[Troubleshooting](#) (see page 91)

Dedicate a Device to a System

TPCF lets you dedicate a device to a system so only jobs on that system can use the device. To dedicate a device, specify the DEDICATED parameter on the VARY command. For example, to dedicate device 01A0 to the local system, issue the following command:

```
VARY 01A0 DEDICATED
```

When you dedicate a device to the local system, TPCF varies the device offline on the external systems and allows the device to be allocated only by local jobs. In addition, the dedicated device is preferenced over other devices so that they are more likely to be used first for local jobs. However, if a job on another system cannot allocate a device, then TPCF may allow this device to be placed in the offline device list.

TPCF also intercepts all VARY ONLINE commands from the external systems and suppresses them for dedicated devices so that these devices are not varied online accidentally to other systems.

You must use the VARY NOTDEDICATED command to release a dedicated device. NOTDEDICATED status causes the device to be varied online on external systems.

Reserve a Device for a Job

To reserve a tape device for a certain job, specify the JOB parameter on the VARY command. For example, to reserve device 01A0 for the job named PAYROLL on all systems, issue the following command:

```
VARY 01A0 JOB=PAYROLL GLOBAL
```

Note: If a device is reserved for a job on the local system only, then the device can only be used by *that* job on *that* system. No job can use the device on any other system, including the job for which the device is reserved locally. The device must be reserved for the job on all systems for the job to be allowed to allocate the device on any system.

Note: Tape swap is not affected by job reserve processing.

How You Reserve a Device for a Single Job

When you reserve a tape device, TPCF prevents any other job from allocating that device and tries to use that device whenever the reserving job needs a device. If the reserved device is not available, then TPCF lets the job select any available device that is not already reserved for another job. If the reserving job cannot allocate this device (or any other device) and enters the allocation recovery process, then TPCF lets the job select any offline device that is not already reserved for another job.

Releasing a Job Reserved Device

Use the NOJOB parameter on the VARY command to release a reserved device, making that device available for allocation by any job. To do this, issue the following command:

```
VARY 01A0 NOJOB GLOBAL
```

More information:

[Troubleshooting](#) (see page 91)

Reserve a Device for a Group of Jobs

By including the wildcard character in the job name, you can reserve a device for a group of similarly named jobs. For example, you can reserve a device for any job with a name that starts with the characters PAY by issuing this command:

```
VARY 01A0 JOB=PAY*
```

The * wildcard character tells TPCF to accept any job with a name that matches the preceding character string. The wildcard character must be the last character in the job name and cannot be the only character.

Note: The VARY JOB command does not allow for multiple job masks for the same device. You must use the TPCEDLXT for this action.

How You Force a Job to Use a Reserved Device

You can use the FORCE parameter on the VARY JOB command to ensure that a job only allocates devices reserved for the job.

To set up the default option for the FORCE parameter, issue the SETOPTION TPCF JOBFORCE command. This command allows you to specify a default value if no value is specified for the VARY JOB command. You can specify YES, which forces the job to use only devices reserved for it, or NO, which serves to prefer these devices to the job, but does not limit the job to only these devices.

To limit the job PAYROLL to using only device 01A0, issue this command:

```
VARY 01A0 JOB=(PAYROLL, FORCE=YES)
```

Important! You should know the following about job reserve processing:

- Unless you are using the FORCE=YES parameter, a job may not allocate the reserved device. Using FORCE=NO lets jobs use devices not reserved for any jobs if a reserved device is unavailable.
- When using the FORCE parameter, be sure that the correct number of devices is reserved for the job. For example, if one job step in the specified job requires three tape devices allocated at one time, then three devices must be reserved for the job. If three devices are not reserved, then the job ends with the JCL error IEF700I.
- All types of processing are ignored if the job reserve processing, the TPCEDLXT exit routine processing, the OVG status (if SETOPTION TPCF OVGINELIG=JOBRESERV or YES) processing, or all three, eliminate all eligible devices for the current allocation. If this situation occurs, then TPCF restores all devices marked ineligible by TPCF back to eligibility.
- You should not reply WAIT,HOLD to the MIM2060 message when using job reserve. This results in message IEF700I, causing the job to abend with a JCL error.

How You Limit Access to Devices Through Job Reserve

In addition to reserving certain devices for particular jobs, you can use the job reserve facility to limit access to a group of devices. You can limit access to remote devices or non-tape devices defined on the system as tape devices, thus eliminating the need to cancel local jobs that incorrectly allocate remote or non-tape devices.

You typically limit access to these devices by defining esoteric device groups to z/OS. The esoteric device groups you create are subsets of the generic device groups created by z/OS. You can then specify the esoteric group name on the UNIT parameter in your JCL to allocate one of these devices. However, there are two potential problems with this method:

- Because remote and non-tape devices are also defined in a generic device group, any JCL request that specifies the generic group on the UNIT parameter can access these devices.
- If no UNIT parameter is specified in the JCL, then z/OS uses the generic device specification contained in the catalog entry.

You can overcome these z/OS limitations with the job reserve facility. To do this, you reserve all of the devices in the group to a non-existent (or dummy) job name. This approach takes advantage of the fact that job reserve processing eliminates *no* devices in the EDL if the reserve specifications eliminate *all* of the devices in the EDL.

For instance, a job that specified a generic UNIT parameter would have the reserved devices eliminated from its EDL, allowing it to only allocate local devices. If a job specified a UNIT parameter that only included the reserved devices, then device elimination would be bypassed since all of the devices in the EDL would be eliminated by the reserve specification. This allows the job to allocate from this reserved set of devices.

Important! TPCF ignores the reserve only if *all* of the devices in the EDL are eliminated.

Example:

This example illustrates the procedure you can use to limit access to remote tape devices through the job reserve facility. Assume that device groups 0580-058F and 0680-068F belong to the same generic device group. This example assumes that any request for an input tape that does not specify a UNIT, or that specifies a generic unit, is allocating a local device. If greater control over this process is required, then it is necessary to use the TPCEDLXT exit routine.

1. This step illustrates how you would create two esoteric device groups for devices 580-58F and 680-68F, defined to the system as remote devices and local devices. Specify the following z/OS SYSGEN statements:

```
UNITNAME NAME=REMOTE,UNIT=(0580,F) /* REMOTE TAPE DRIVES
UNITNAME NAME=LOCAL,UNIT=(0680,F) /* LOCAL TAPE DRIVES
```

2. This step illustrates how to limit allocation of devices located at the remote data center. Issue the CA MIA VARY command to specify a non-existent job name for the device group 0580-058F:

```
VARY 0580-058F, JOB=NONEXIST
```

3. This step illustrates an output DD statement you specify in a job to request a local device allocation:

```
//OUTPUT DD DSN=LOCAL.DATASET,UNIT=LOCAL,DISP=(NEW,CATLG)
```

Because UNIT=LOCAL is specified and z/OS does not need to search its catalog for a volume number on an output request, the only candidates for this job are devices from the 0680-068F group. However, at the end of this process, the data set is cataloged, and the generic device group for the data set is stored in the z/OS catalog.

4. This step illustrates how the job reserve facility handles the override process that occurs on an input tape request for which no volume serial number is provided, and no UNIT is coded in the JCL. You can run a job named LOCLREAD with the following DD statement to request a tape device for input:

```
//INPUT DD DSN=LOCAL.DATASET,DISP=(OLD,KEEP)
```

When z/OS searches the catalog for the volume where this data set is stored, it uses the generic device group stored in the catalog entry of the data set.

Devices in both groups 0580-058F and 0680-068F are eligible candidates for this request because they are all defined to the system in the same generic device group. However, because the devices in groups 0580-058F are reserved for job name NONEXIST, TPCF eliminates devices in this group and allows a device from the 0680-068F groups to be allocated.

5. This step illustrates how the job reserve facility handles the situation that occurs when allocation of a remote tape device is really desired. Run an output job named RMOTWRIT to create a tape data set:

```
//OUTPUT DD DSN=REMOTE.DATASET,UNIT=REMOTE,DISP=(NEW,CATLG)
```

Because UNIT=REMOTE is specified on this statement and z/OS does not need to search its catalog for a volume number on an output request, the only candidates for this request are devices from the 0580-058F group.

TPCF typically eliminates devices in this group because they are reserved by job name NONEXIST. However, the job reserve feature protects you from the problem of eliminating all eligible devices. In these cases, TPCF ignores the job reserve and allows the devices in groups 0580-058F to be eligible candidates for this allocation request.

More information:

[User Exits](#) (see page 119)

How You Limit Access to Devices Through the OVERGENNED Status

The SETOPTION TPCF OVGINELIG={NO|YES|JOBRESV} setting determines whether OVERGENNED devices are marked ineligible in the EDL. Marking OVERGENNED devices as ineligible in the EDL eliminates them from allocation early in the allocation process and affects the way device selection is performed for a job. Marking OVERGENNED devices ineligible in the EDL can prevent jobs from going into allocation recovery and prevent jobs from being canceled in allocation recovery.

Here are some examples of using SETOPTION TPCF OVGINELIG. In all the following examples, TAPE1 has an EDL consisting of devices 580-58F.

Examples:

1. This example assumes the following are true:
 - Devices 580-586 are job reserved for job TAPE2.
 - Devices 587-58F are OVG.
 - SET TPCF OVGINELIG=NO

When TAPE1 goes into allocation, devices 580-586 are eliminated because they are reserved for another job. Since devices 587-58F are OVG, they are offline. TAPE1 goes into allocation recovery with an offline device list of 587-58F. CA MIA eliminates 587-58F from the offline device list because they are OVG. There are no devices left eligible for allocation and the job is canceled in allocation recovery by TPCF after the message MIM2044 ALL DEVICES HAVE BEEN ELIMINATED is issued.

2. This example assumes the same values as the preceding example, except SET TPCF OVGINELIG=YES

This time, when TAPE1 goes into allocation, devices 580-586 are eliminated because they are reserved for another job *and* devices 587-58F are eliminated because they are OVG and OVGINELIG=YES. CA MIA has eliminated all devices during the initial allocation process so CA MIA restores 580-586 to the job for use. If a device is available in the 580-586 range, then the job allocates the device. If not, then the job goes into allocation recovery with 580-586 in the offline device list or wait eligible list. In the previous example, this job was canceled. In this example, the job is not canceled but is allowed to use a device reserved for another job.

In this case, SET TPCF OVGINELIG=JOBRESV would have the same effect as SET TPCF OVGINELIG=YES since job reserve is in effect.

3. This example assumes the following are true:

- Devices 580-58F are OVG
- SET TPCF OVGINELIG=NO

When TAPE1 goes into allocation, all devices are offline because they are OVG. TAPE1 goes into allocation recovery with an offline device list of 580-58F. CA MIA eliminates 580-58F because they are OVG. There are no devices eligible for this allocation. TPCF cancels the job in allocation recovery after the MIM2044 message is issued.

4. This example assumes the same values as the preceding example, except SET TPCF OVGINELIG=YES

When TAPE1 goes into allocation, all devices are eliminated because OVGINELIG=YES. The job ends with a START COMMAND DEVICE ALLOCATION ERROR from the operating system.

In examples 3 and 4 the job ends. In example 3, however, TAPE1 had to go through the entire allocation recovery process before being canceled. In example 4, TAPE1 ended before entering allocation recovery, eliminating the unnecessary overhead of allocation recovery.

If SET TPCF OVGINELIG=JOBRESV was specified, then the effect would be the same as example 3, since no job reserve is in effect.

Using Exit Routines to Influence Device Selection

TPCF provides three exit routines that you can use to influence device selection during allocation and allocation recovery:

TPCEDLXT

TPCF uses elimination logic to mark devices ineligible in the EDL early in the z/OS allocation process. Therefore, you can use this routine to unconditionally prevent devices from being candidates for allocation.

TPCREXT

Customizes the TPCF processing of allocation recovery messages. You can use this routine to address site-specific process needs that are not completely served by the CA MIA standard recovery options.

TPCSRMXT

Customizes the TPCF device preferencing to address site-specific needs not completely served by the CA MIA standard preferencing options.

More information:

[User Exits](#) (see page 119)

Dynamic Detection and Notification (DDN)

Diagnosing delays in tape device allocation is challenging, especially when multiple systems are involved. The Delay Detection and Notification (DDN) feature simplifies tape device allocation delay detection and provides real-time diagnostics to z/OS system operators when a problem occurs. This information allows operators to expedite resolution of tape device allocation delays involving CA MIA managed devices.

Example:

To enable DDN detection, issue the following command:

```
MIM SETOPTION DDN=ON
```

Contention Delay Detection

When DDN contention detection is turned on, users can be automatically notified of potential tape device delays due to a task owning tape device allocation locks for an extended period of time.

To enable contention delay warnings to be issued on a reoccurring five-minute interval, issue the following command:

```
MIM SETOPTION DDN(ON, CONTENTION=5)
```

For more information, see the DDN option in the *CA MIM Statements and Commands Reference Guide*.

Notes:

- Place the DDN SETOPTION in your MIACMNS member to enable the feature during CA MIA initialization. The default is OFF and has a range of 1 to 60 minutes.
- If DDN is turned off, then no allocation delay analysis is performed.
- To diagnose an issue presented by DDN further, use the ANALYZE TAPEDELAY command.

WAIT/NOHOLD Detection

When the CA MIM DDN Wait/NoHold detection feature is turned on, a warning is issued when two or more tasks monopolize a set of device allocation locks due to excessive Wait/NoHold responses.

To enable DDN Wait/NoHold detection to report a loop exceeding 10 iterations, issue the following command:

```
MIM SETOPTION DDN(WAITNOHOLD=10)
```

For more information, see the DDN option in the *CA MIM Statements and Commands Reference Guide*.

Notes:

- To enable the feature during CA MIA initialization, place the DDN SETOPTION in your MIACMNS member. The default is OFF and has a range of 10 to 9999 responses.
- DDN recognizes when work is progressing on the system and only DDN notifies operators when a tape device monopolization condition persists.
- If DDN is turned off, then no allocation delay analysis is performed.
- To diagnose an issue presented by DDN further, use the ANALYZE TAPEDELAY command.

TPCF Device Processing

It is not always apparent why one device is chosen over another device during allocation processing. You may find that a device you would rather not use is being used before a more preferred device in spite of the criteria you have defined to TPCF. If this is the case at your site, then you should read this section to find out more about TPCF device processing and how z/OS allocation processing may reduce the effectiveness of TPCF under certain circumstances. Also read this section if jobs are entering allocation recovery unexpectedly.

This section provides you with detailed information about the following topics:

- Factors that influence or limit the ability of TPCF to preference devices
- Why jobs enter allocation recovery when online devices are available
- What you can do to prevent jobs from entering allocation recovery
- What you can do if TPCF elimination or preferencing are not producing the results you expect

Factors That Influence TPCF Preferencing

TPCF does not replace the z/OS allocation process; TPCF interacts with z/OS allocation. The types of criteria TPCF uses to influence device selection (logic in the TPCEDLXT and TPCSRMXT exit routines and device status changes you make through the VARY command) are applied between steps in the z/OS allocation process.

TPCF is not the only program that influences device selection through preferencing. z/OS and other vendor software use preferencing. Final device selection is a combination of the preferencing applied by z/OS, TPCF, and other vendor software. Factors that influence or limit TPCF preferencing include:

- The contents of the z/OS eligible device list (EDL)
- The way z/OS allocation and other vendor software eliminate devices from eligibility for allocation
- The way z/OS allocation and other vendor software preference devices in the EDL

How the EDL Limits TPCF Preferencing

The first critical factor in TPCF preferencing is the EDL for a job. TPCF does not create its own list of devices during allocation. Instead, TPCF applies your device selection criteria to the list of devices created by z/OS allocation. TPCF has no influence on the creation of the EDL.

How z/OS and Other Vendor Products Eliminate Eligibility of Devices

Devices can be eliminated from eligibility for allocation for a variety of reasons. Some common examples are:

- z/OS eliminates devices that are offline, inaccessible, or allocated.
- TPCF eliminates devices reserved for other jobs, devices marked OVERGENNED according to the SETOPTION TPCF OVGINELIG parameter, and devices specified by the TPCEDLXT exit.
- Robotic software eliminates devices based on the existence of the device inside or outside the robot.

Any preferencing applied to devices eliminated from allocation is ineffective because these devices cannot be chosen for allocation.

How z/OS and Other Vendor Products Preference Devices

This section describes how z/OS and other vendor software can influence or limit the ability of TPCF to preference.

Devices can be preferenced for a variety of reasons. Some common examples are:

- z/OS generic preferencing (preference values set for 3480, 3490, and other devices in HCD panels)
- z/OS ACL preferencing (for details, see the following sections)
- Robotic software preferencing based on the existence of a device inside or outside the robot or client customization

Any preferencing performed by z/OS or vendor software combines with and possibly overrides TPCF preferencing. Final preferencing by z/OS results from a combination of the elimination and preferencing performed by z/OS and other vendor products.

z/OS ACL Preferencing

The following sections discuss z/OS ACL preferencing considerations.

More information:

[Troubleshooting](#) (see page 91)

How z/OS Processes Non-Specific, Non-Temporary Requests

Non-specific, non-temporary mount requests are those in which DISP=(NEW,KEEP), DISP=(NEW,CATLG), or DISP=(NEW,PASS) is specified in the JCL of the job.

If the job does not need a specific volume and it uses a non-temporary data set, then z/OS prefers to allocate assignable devices in this order:

1. Assignable devices on which the ACL feature is installed and active
2. Assignable devices on which the ACL feature is installed, but inactive
3. Assignable devices on which the ACL feature is not installed

How z/OS Processes Specific, Temporary Requests

If a job requests a specific volume, or the input data set is temporary (that is, the data set will not be retained at the end of the job), then z/OS prefers to allocate assignable devices in this order:

1. Assignable devices on which the ACL feature is not installed
2. Assignable devices on which the ACL feature is installed, but inactive
3. Assignable devices on which the ACL feature is installed and active

As a result of ACL processing, TPCF preference logic may be overridden.

Unexpected Results in Device Allocation

Sometimes device allocation for a job does not produce expected results. It may be that a job goes into allocation recovery when it appears that online devices are available. Or a job may select an unexpected device for allocation or not select an expected device. Most of the time, the unexpected results are due to the elimination or preferencing applied to the devices. The final selection of devices of z/OS is a combination of all the elimination and preference logic performed by z/OS and other vendor software.

For example, robotic software, TPCF (job reserve, TPCEDLXT exit, and so on), or both may eliminate all available online devices, sending a job into allocation recovery, when it appears online devices are available. When a job gets unexpected results in device allocation, examine the elimination and preferencing affecting the devices in the EDL of the job. Are the devices inside or outside a robot? Are they ACL devices? Are any devices reserved?

TPCF provides tracing through the DEVSEL24, DEVSEL78, and RECOVERY parameters of the SETOPTION TPCF SETTRACE and SETPRINT commands that assist in diagnosing device selection and allocation recovery problems.

How You Control Allocation Recovery Actions of TPCF

Allocation recovery is the part of the z/OS allocation process that a job enters when it cannot allocate a suitable online device. During allocation recovery, z/OS assembles a list of allocated devices and a list of offline devices:

- *Wait-eligible device list*-allocated devices for which a job can wait.
- *Offline device list*-offline devices that can be brought online for this job.

To control allocation recovery actions of TPCF

Either issue the SETOPTION AUTOREPLY command, use a CA MIA exit routine, or both.

How You Control Recovery Processing Using SETOPTION AUTOREPLY

If you specify the SETOPTION AUTOREPLY=ON command, then TPCF intercepts the *wait-eligible device list* and the *offline device list*. You then can tell TPCF how to respond to the z/OS IEF238D message that has been issued. By having TPCF respond to these messages, an operator may not have to respond to them whenever a job cannot select an online device. Specify the command SETOPTION AUTOREPLY=OFF to stop TPCF from intercepting the device lists and responding to allocation recovery messages.

The IEF238D message provides a list of ways to handle the current allocation recovery situation. Depending on the circumstances, z/OS may allow you to cancel the job, bring an offline device online, or make the job wait for a wait-eligible device to become available. If you can use an offline device, then z/OS also issues message IEF247I, which lists the offline devices that are suitable for the job.

We recommend that you specify the following values for IEF238D and IEF433D on the SETOPTION TPCF AUTOREPLY command:

```
AUTOREPLY=(IEF238D=CONDITIONAL, IEF433D=NOHOLD)
```

Important! If you reply WAIT to message IEF238D, then you must also reply HOLD or NOHOLD to message IEF433D. Therefore, the SETOPTION AUTOREPLY command must be set to MANUAL for both messages IEF238D and IEF433D, or for neither. They cannot be mixed. The one exception to this is that the value for the IEF238D parameter can be set to CANCEL, regardless of what the IEF433D parameter is set to. This requirement is also true for the TPCREXT user exit routine.

How You Override the z/OS Default Value for Automatic Replies

IBM implemented a standard interface to the allocation recovery process. One aspect of this implementation limits the number of times TPCF can automatically reply to a particular job step. The z/OS default for this value is 5. However, you can override this z/OS default value using either of the following SETOPTION commands:

```
SETOPTION AUTOREPLY SPECMAXNWAIT  
SETOPTION AUTOREPLY OFFLNMAXNWAIT
```

As long as CA MIA is active and managing your devices, the new value specified for these exit routines stays in effect. You can specify a value from 1 to 255, or you can specify UNLIMITED, so that TPCF never stops replying to recovery messages for a job step.

Note: The TPCF OFFLNMAXNWAIT and SPECMAXNWAIT parameters take effect (override the z/OS settings in SYS1.PARMLIB) regardless of the TPCF AUTOREPLY=ON/OFF setting. This gives you the option of an unlimited number of replies to allocation recovery messages when TPCF is replying (AUTOREPLY=ON) and when TPCF is not replying (AUTOREPLY=OFF). If a specific number of replies are preferred, then you may set TPCF OFFLNMAXNWAIT and SPECMAXNWAIT to a number from 1 to 255. This also takes effect regardless of the TPCF AUTOREPLY=ON/OFF setting.

The TPCF OFFLNMAXNWAIT and SPECMAXNWAIT parameters do not take effect if MIMINIT TPCF=OFF.

Note: For more information, see the *CA MIM Statement and Command Reference Guide*.

How TPCF Responds When a Job Enters Allocation Recovery

The value you specify for the SETOPTION AUTOREPLY IEF238D command tells TPCF how to respond to the z/OS IEF238D message.

Note: For more information, see the *CA MIM Statement and Command Reference Guide*.

If AUTOREPLY=ON and TPCF requests operator intervention, then TPCF reissues the z/OS IEF238D as a MIM2060 message. If TPCF responds automatically to the z/OS IEF238D message, then TPCF issues a multi-line WTO containing the IEF238D and MIM2046 messages.

Note: If you specify SETOPTION AUTOREPLY=OFF, the z/OS IEF238D message appears instead of the MIM2060 message and TPCF will *not* automatically reply to any messages, nor will it remove devices from the offline device list.

How You Respond when CANCEL Is the Only Option

Under the following circumstances, CANCEL is the only option available to the operator:

- WAIT is not an option on the IEF238D message
- There are no offline devices to satisfy the request, or all offline devices have been eliminated for one of the following reasons:
 - Job reserve
 - z/OS or other vendor software
 - TPCEDLXT exit routine
 - Overgennded status
 - Externally DEDICATED status
 - NOT AVAILABLE status

When CANCEL is the only option available to the operator, you can have TPCF automatically respond CANCEL by specifying SETOPTION AUTOREPLY=(ON,NOWAIT=(CANCEL=YES)). This is the default setting.

Note: If CANCEL=NO is specified, then the MIM2060 WTOR is issued with CANCEL as the only option. The operator is forced to reply CANCEL. Until the operator replies CANCEL, the job in allocation recovery holds the tape device allocation resources that are serialized across the MIMplex. Tape allocations on all systems in the MIMplex experience delays until the operator replies CANCEL.

We recommend that you specify CANCEL=YES unless you have a specific reason for not doing so.

How TPCF Removes Devices from the Offline Device List

TPCF removes devices from the offline device list as follows:

1. TPCF always removes overgennded devices from the offline device list. Therefore, all devices that are defined in the IOGEN or HCD, but do not physically exist, should be assigned OVERGENNED status, using the VARY OVERGENNED command. Additionally, devices reserved for other jobs, devices eliminated by the TPCEDLXT, and any other devices marked ineligible (for example, by robotic software) do not appear in the offline device list.

2. TPCF removes NOTAVAILABLE and externally DEDICATED devices from the offline device list according to the following rules:
 - When WAIT is an option on the z/OS IEF238D message, TPCF always removes the devices.
 - When WAIT is not an option, TPCF tries to list other offline devices first.
 - If all offline devices are NOTAVAILABLE, externally DEDICATED, or both, then TPCF lists the devices as determined by the values specified on the SETOPTION AUTOREPLY=(NOWAIT=(EXTDEDDISP=YES/NO,NOTAVLDISP=YES/NO)) options.
 - If SETOPTION AUTOREPLY=(NOWAIT=(EXTDEDDISP=YES,...)) is specified, then TPCF displays any externally DEDICATED devices.
 - If SETOPTION AUTOREPLY=(NOWAIT=(EXTDEDDISP=NO,...)) is specified, or there are no externally DEDICATED devices, then TPCF displays any NOTAVAILABLE devices if SETOPTION AUTOREPLY=(NOWAIT=(...,NOTAVLDISP=YES)) is specified.
 - If there are no NOTAVLDISP devices, then no devices are displayed (that is, TPCF eliminates all offline devices).
3. After TPCF has removed devices from the offline device list, TPCF responds to the IEF238D message according to the value specified on the SETOPTION AUTOREPLY IEF238D= parameter.
 - IEF238D=CANCEL

If there are no devices left in the offline device list, TPCF replies CANCEL or asks the operator to reply CANCEL based on the SETOPTION AUTOREPLY NOWAIT=(CANCEL=) parameter. If at least one device remains in the list, then TPCF asks the operator to reply.
 - IEF238D=WAIT

If WAIT is an option, then TPCF replies WAIT. If WAIT is not an option and devices exist in the offline device list, then TPCF asks the operator to reply. Otherwise, TPCF replies CANCEL or asks the operator to reply CANCEL based on the SETOPTION AUTOREPLY NOWAIT=(CANCEL=) parameter.
 - IEF238D=CONDITIONAL

Any time there are devices in the offline device list, TPCF asks the operator to reply. If there are no offline devices and WAIT is an option, then TPCF replies WAIT. If there are no offline devices and WAIT is not an option, then TPCF replies CANCEL or asks the operator to reply CANCEL based on the SETOPTION AUTOREPLY NOWAIT=(CANCEL=) parameter.
 - IEF238D=MANUAL

TPCF asks the operator to reply.

Whenever TPCF asks the operator to reply, it issues the MIM2060 message with the reply options. When you need to reply and you can use an offline device, TPCF reissues the z/OS IEF2471 message as message MIM2042. The MIM2042 message lists the offline devices you can use.

Note that we recommend using IEF238D=CONDITIONAL. When using CA MIA, all devices available for allocation on a system should be varied online to the system. All other tape devices should be varied OVERGENNED, NOTAVAILABLE or be externally dedicated, as appropriate. In other words, there should not be any offline devices in the offline device list. Then, every time WAIT is an option, CA MIA will reply WAIT. If CA MIA does not reply WAIT when WAIT is an option, then there are devices in the offline device list. This should be an indication to operators that the devices in the list should either be brought online or varied OVERGENNED, NOTAVL, and so on as necessary. Specifying IEF238D=WAIT will ensure that WAIT is replied every time WAIT is an option. But IEF238D=WAIT will reply WAIT even when there are offline devices in the offline device list. This could hide the fact that there may be offline devices that should be online or varied with another status.

How You Respond When a Job is Waiting for a Device

z/OS issues message IEF433D whenever you (or TPCF) tell z/OS to make a job wait for a device to become available. You can tell TPCF how to respond to the z/OS IEF433D message by specifying one of these values for the SETOPTION AUTOREPLY=(IEF433D=*value*) command:

- If you specify IEF433D=HOLD, then waiting jobs receive a suitable device before other jobs. However, z/OS still allows jobs using dynamic allocation to allocate a device. When HOLD is replied, the job continues to hold device group locks, preventing other jobs requesting the locks from proceeding in allocation.
Important! Do not specify IEF433D=HOLD if you are reserving devices for jobs through the VARY command, if you have defined elimination logic in the TPCEDLXT exit routine, or if you are running with tape robotic software. Replying HOLD under these circumstances can cause jobs to fail during allocation. We recommend that you do not specify IEF433D=HOLD, even when you are not reserving devices or using an exit routine, since allocation processing may be slowed down. A reply of HOLD results in the job holding allocation resources that may be needed by other jobs.
- If you specify IEF433D=NOHOLD, then jobs are placed on a wait queue until a device of the correct type deallocates or a VARY AVAILABLE command is issued for an eligible device. When NOHOLD is replied, the device group locks are released.
- If you specify IEF433D=MANUAL, then you can make a choice between HOLD and NOHOLD whenever the message is issued. If you specify IEF433D=MANUAL, then you must also specify IEF238D=MANUAL.

How You Determine Where to Send Message MIM2046

If you specify `AUTOREPLY=ON` on the `SETOPTION` command, then the MIM2046 message indicates how an operator or TPCF replied to z/OS messages IEF238D and IEF433D. Use the MIM2046 parameter on the `SETOPTION` command to indicate whether TPCF should send message MIM2046 to the console of the operator *and* to the system log, or the system log only:

- For the console of the operator and the system log, specify the following:

```
SETOPTION AUTOREPLY=(MIM2046=CONSOLE)
```

- For the system log only, specify the following:

```
SETOPTION AUTOREPLY=(MIM2046=HARDCOPY)
```

How You Use Exit Routines to Control Allocation Recovery

Another way to control how TPCF responds to allocation recovery messages is to modify the `TPCRECXT` exit routine. For example, you can use the `TPCRECXT` exit routine to specify that you want to reply `WAIT` to a specific job. Also, you can use the `TPCRECXT` exit routine to modify the offline device list.

You can use the `TPCEDLXT` and the `TPCSRMTXT` exit routines to modify the eligible device list (EDL) and device preferencing.

More information:

[User Exits](#) (see page 119)

z/OS Dynamic Device Reconfiguration (SWAP) Processing

When a device fails, operators can enter the `SWAP` command to perform dynamic device reconfiguration (DDR) or the operating system can automatically initiate DDR in response to a detected device failure. DDR allows the operator to move or swap a de-mountable volume from one device to another with compatible attributes.

When the system detects a device failure or when an operator issues the `SWAP` command from a console, DDR processing under the `MASTER` address space issues a series of subsystem calls and unit allocation requests as the tape `SWAP` is processed. DDR uses the unit allocation calls to allocate, through UCB bit settings, potential TO devices for the `SWAP` and to determine if the potential TO devices can be used by `SWAP` processing. The `SYSIEFSD/DDRTPUR EXCL` enqueue is obtained during the allocation of the potential TO devices to serialize the `SWAP` device allocations with z/OS common allocations, other `SWAPs` that may be occurring, and `HCD ACTIVATE`.

While the MASTER address space is holding the SYSIEFSD/DDRTPUR EXCL during SWAP processing, no new allocation requests, no new SWAP requests, and no new HCD ACTIVATES can start. Conversely, the MASTER address space is unable to obtain EXCLUSIVE control of SYSIEFSD/DDRTPUR to process a new SWAP until all old allocations, SWAPs, and HCD ACTIVATES have completed, therefore, current allocations, SWAPs, and ACTIVATES can prevent SWAP processing from initiating.

In summary, SWAP is serialized with other device activity because SWAP must wait for all existing allocations, SWAPs, and ACTIVATES to complete, and no new allocations, SWAPs, or ACTIVATES can occur until SWAP completes.

SWAP Processing

CA MIA serializes the device allocations performed by SWAP across all the systems in the MIMplex to prevent tasks on external systems from simultaneously allocating the same device as a local SWAP.

CA MIA serializes SWAP processing across all active systems in the MIMplex by obtaining the device group lock for the device requested on each DDR unit allocation and then plugging the device allocated by SWAP on the external systems.

When CA MIA intercepts a DDR unit allocation request for a potential TO device during a tape SWAP, CA MIA requests the device group lock for the device before allowing the unit allocation request to continue. CA MIA then serializes the device group lock request with the other systems in the MIMplex as it would device lock requests for normal allocations.

CA MIA does not allow the unit allocation to continue until the device group lock becomes available to the local system; that is, no other systems in the MIMplex are currently using the device group lock. Once the lock is available to the local system, CA MIA allows the unit allocation request to continue to allocate the device. When the allocation is complete, CA MIA releases the device group lock for the device, notifies the other systems in the MIMplex that the lock is available, and plugs the allocated device on the other systems.

With device group lock serialization, only those allocations and SWAPs on external systems that are requesting the same device group locks as the local SWAP are delayed by the SWAP. Allocations and SWAPs on the external systems that do not involve the device group requested by the local SWAP continue. Conversely, only those allocations and SWAPs on external systems that are requesting the same device group locks as the local SWAP delay the local SWAP. If no allocations or SWAPs on external systems hold the device group lock requested by the local SWAP, the SWAP continues.

With device group lock serialization, local SWAPs do not delay HCD ACTIVATES on external systems.

Note: Since z/OS carries out local serialization of the SWAP through the enqueue SYSIEFSD/DDRTPUR EXCL, allocations, SWAPs, and HCD ACTIVATES on the system local to the SWAP is delayed by the SWAP. Any local allocations, SWAPs, and HCD ACTIVATES will delay the SWAP. This is true, regardless of the device groups associated with the local allocations and SWAPs.

More information:

[How z/OS Device Allocation Works](#) (see page 22)

[How z/OS Device Allocation with CA MIA Works](#) (see page 24)

SWAP-related Messages

The MIM2207I highlighted message is issued when CA MIA intercepts the z/OS SSI call indicating the start of a tape SWAP. This message provides the name of the job and the FROM device involved in the SWAP. The MIM2207I message is issued early in the SWAP process, before CA MIA begins serialization of the SWAP and before any z/OS SWAP messages are issued. The MIM2207I message is informational in nature and is deleted automatically by CA MIA as part of the SWAP progresses. The MIM2207I message remains highlighted until CA MIA issues the MIM2208I or the MIM2209I message or the swapping task ends. The MIM2207I message can be very helpful in detecting the presence of a SWAP when tape allocation delays arise due to a delay in SWAP processing. Because the MIM2207I message is issued early in the SWAP process, before the z/OS messages, the MIM2207I message may be the only external indication that a SWAP is occurring.

If the MIM2207I message remains outstanding for an extended period of time, then it may indicate a delay in the ability of CA MIA to serialize the SWAP or outstanding SWAP WTORS are not being responded to in a timely manner. Because delays in SWAP processing can lead to delays in tape allocation, the cause of the delay in SWAP processing should be investigated.

The MIM2208I highlighted message is issued when CA MIA intercepts the z/OS SSI call indicating that DDR is ready to switch the UCBs of the FROM and TO devices. At this point in the SWAP, DDR has chosen a final TO device. Therefore, the MIM2208I message includes the name of the TO device as well as the FROM device and job name. CA MIA has completed serialization of the SWAP and the SWAP is close to completion. The MIM2208I message is informational in nature and is deleted when the MIM2209I message is issued or when another MIM2207I message is issued in the case of a recycled SWAP or when the swapping task ends. A SWAP recycles if DDR finds a problem with the final TO device chosen or the tape after the UCBs have been switched.

The MIM2209I non-highlighted message is issued when CA MIA intercepts the z/OS SSI call indicating that DDR is in the final stages of the SWAP and the SWAP is completing. At this point, the involvement of CA MIA with the SWAP ends.

SWAP During HCD ACTIVATE Processing

During an HCD ACTIVATE, the IOS address space holds a shared enqueue on SYSIEFSD/DDRTPUR. As discussed in prior sections, z/OS DDR processing uses an exclusive enqueue on SYSIEFSD/DDRTPUR to serialize SWAP allocations with allocations, other SWAPs, and HCD ACTIVATE. Therefore, when an ACTIVATE is occurring on a system, SWAPs on that system are delayed until the ACTIVATE completes and releases the SYSIEFSD/DDRTPUR enqueue. Conversely, when a SWAP is occurring, ACTIVATES are delayed on that system until SWAP processing releases the SYSIEFSD/DDRTPUR enqueue.

If a SWAP starts and requests the SYSIEFSD/DDRTPUR exclusive enqueue while an ACTIVATE in progress holds the enqueue shared, then the SWAP waits behind the ACTIVATE due to the exclusive SYSIEFSD/DDRTPUR enqueue. Also, any new allocations and SWAPs on the local system wait behind the exclusive enqueue for the SWAP. In this situation, all local SWAPs and allocations are delayed until the ACTIVATE completes.

CA MIA serializes SWAPs by requesting the device group locks for the devices indicated on DDR unit allocation calls during the SWAP. Since CA MIA uses locks rather than an exclusive SYSIEFSD/DDRTPUR enqueue to serialize the SWAPs on external systems in the MIMplex, only allocations and SWAPs needing the same device group locks are delayed on the external systems. No other allocations and SWAPs are delayed and HCD ACTIVATES are not delayed on the external systems. Also, because CA MIA does not need to obtain a blocking SYSIEFSD/DDRTPUR enqueue on the external systems, HCD ACTIVATES on external systems will not delay a SWAP occurring locally. A local SWAP is delayed only by allocations or SWAPs on external systems that need the same device group lock as the local SWAP. The CA MIA device lock serialization provides a very granular level of cross-system SWAP serialization.

Performance Considerations

This section discusses some performance considerations for CA MIA.

Setting INTERVAL and CYCLE Parameters

By design, during periods of little system activity, CA MIA waits for the expiration of a timer before accessing the control file. This timer is determined by values defined on the SETOPTION MIM statement in the MIMCMNDS member of the CA MIM parmlib. The parameters, INTERVAL and CYCLE, are supplied with default values of 1. Unless you have tuned CA MIM as described in the *CA MIM Programming Guide*, we recommend that these values remain set to 1. This ensures that CA MIA accesses the CA MIM control file to process important transactions that may be waiting on the control file for it to process, even on a system with relatively little tape activity.

How You Automate Replies to Allocation Recovery Prompts (AUTOREPLY)

z/OS serializes allocation using device group locks. When a job goes into allocation recovery, it generates message IEF238D, which lists the options available in recovery for this current allocation request.

A subsequent WTOR, message IEF433D, is issued requesting operator intervention to determine whether the subset of the device group locks requested by this job should be held until the remainder of the locks become available. The operator can reply HOLD or NOHOLD to this message. While either of these WTORS is outstanding, allocation of devices represented by these device group locks on the local system stops. That is, the job in allocation recovery holds device group locks and other jobs wanting the locks must wait until the WTORS for the job in recovery are replied to.

When CA MIA is started, it extends the local z/OS serialization to all systems in the MIMplex. Allocation stops on all systems for all jobs requesting even a subset of the device group locks held by the task for which the allocation recovery prompt was issued.

As soon as the WTOR has been responded to, allocation resumes on all systems. For this reason, we recommend that outstanding Allocation Recovery prompt replies should be automated. The AUTOREPLY function of TPCF performs this function, and can virtually eliminate the need for manual operator intervention in allocation recovery.

More information:

[How z/OS Device Allocation Works](#) (see page 22)

Reply To SWAP WTORS

As part of normal z/OS SWAP processing, WTORS are issued by DDR that request operator permission to proceed with a SWAP. Until these WTORS are responded to, DDR holds resources that prevent new jobs from entering z/OS allocation. CA MIA extends this z/OS serialization across all systems in the complex. For this reason, outstanding SWAP WTORS should be responded to as soon as they are detected to prevent delays in allocation.

Separate CA MIM into Multiple Address Spaces

CA MIM components are often run together in the same address space. Under normal operating conditions, CA MIA adds minimal processing overhead to a shared address space and the performance impact is negligible. However, if the CA MIA TRACE facility is activated at the request of CA MIA support to provide documentation for a reported problem, then the processing overhead and performance impact of the trace can have significant impact. For this reason, it may be advantageous to separate the CA MIM product into multiple address spaces.

It is common, when running all components of CA MIM, to run CA MII in its own address space, and CA MIA combined with CA MIC in a separate address space. Sample CA MIM parmlib members are provided in the CAI.CBTDPARM data set.

Note: For more information, see the members MIACMNDS, MIAINIT, MIAPROC, and MIASYNCH in the CAI.CBTDPARM data set sample parmlib.

Note: Using the coupling facility, XCF, or CTC communication methods for CA MIM provide the best response times for managing enqueue, tape, and DASD resources. These methods are recommended for any CA MIM address space running CA MII, because enqueue processing requires the best possible response time. In most cases, however, tape processing does not require the faster response times that enqueue processing does. If you have CA MIA in its own address space, then you may find the shared DASD communication method provides satisfactory response time for tape allocation, freeing CTC, XCF, or coupling facility resources for applications where response time is critical.

Sysplex Considerations

CA MIA currently does not have any compatibility restrictions when running on systems participating in a sysplex or parallel sysplex.

z/VM Considerations

When CA MIA is running on multiple guest operating systems on a single processor, every guest must have a path to each tape drive it shares. This means that a real tape drive should be attached or dedicated to all guest machines that share the tape drive.

Shared Tape Support

Prior to the availability of shared tape support, each real device address could be attached to only one virtual machine at a time, requiring the definition of an independent real address for each virtual machine that would share the tape drive. You do this by defining paths to multiple channel interfaces on a single control unit as if there were multiple control units, each with a single channel interface.

The Shared Tape Support feature of z/VM alleviates this restriction. With Shared Tape Support, a real tape device can be attached to an unlimited number of guest systems using the same address. This is accomplished using the new MULTIUSER option on the CP ATTACH command. The MULTIUSER option implies the NOASSIGN option. Guests to which a device is attached as MULTIUSER must provide their own serialization for the device.

For example, if a device is defined to a z/VM host system (VMH) with address 3A0, it can be attached to one z/VM guest system (ZVM1) and two z/OS guest systems (ZOS1 and ZOS2) using the following CP ATTACH commands:

```
CP ATTACH 3A0 TO ZVM1 AS 880 MULTIUSER
CP ATTACH 3A0 TO ZOS1 AS 880 MULTIUSER
CP ATTACH 3A0 TO ZOS2 AS 880 MULTIUSER
```

Each guest system knows the device as address 880. CA MIA running on each guest system serializes access to the device. CA MIA runs on guest systems ZOS1 and ZOS2. CA MIA for z/VM runs on guest system ZVM1. The device must be defined to CA MIA on each system using the same global name. Global name G880 is used in this example. For CA MIA for z/VM running on VM1, the device must be defined with global name G880 in the UNITS MIM file. For CA MIA running on ZOS1 and ZOS2, the device must be defined with global name G880 in the MIMUNITS member.

Note: The Shared Tape Support enhancement is not designed to share a device between CMS users or between CMS users and guest systems. The use of the MULTIuser ATTACH option assumes that the user to whom the device is attached serializes access to a device. CMS users do not have the means to serialize access to a device with other CMS users or with guest systems. There is no serialization of the device if it is attached to multiple CMS users or to a CMS user and a guest as MULTIuser. Data corruption can occur as the CMS users and guests access the device at the same time.

CA MIA for z/VM, in conjunction with CA MIA, provides serialization of devices between CMS users and z/OS guests while using the Shared Tape Support enhancement to minimize the number of paths required per device. Sharing devices between CMS users and an unlimited number of guests using CA MIA requires only two paths per device:

- The one path attaches the device to all of the guests as MULTUser.
- The other path attaches the device to CMS users.

For example, to share device 3A0 between CMS users on VMH and guests ZVM1, ZOS1, and ZOS2 in the complex defined above, the following additional setup is required:

- CA MIA for z/VM must be running on the host z/VM system, VMH.
- You must define a second address and path for device 3A0. In this example, the second address is 4A0.
- You must define the second address, 4A0, to CA MIA for z/VM on VMH with the same global name as used for CA MIA running on the guest systems, G880. You do this by customizing the UNITS MIM file for VMH.

In this configuration, CA MIA for z/VM on VMH does not manage device address 3A0 and is not aware of the CP ATTACHes of 3A0 to the guests. However, CA MIA on the guest systems manages device address 880, which is the address the guests use for access to the device through z/VM address 3A0. Any time a guest allocates device 880, CA MIA on the guest informs CA MIA for z/VM on VMH that device with global name G880 is in use by a guest.

If a CMS user on VMH wants to attach the device, then a CA MIA for z/VM ATTACH command (MI ATTACH) is used to attach device address 4A0/G880 to the user. CA MIA for z/VM on VMH recognizes that the ATTACH is for the device with global name G880. If G880 is already in use on a guest (through address 3A0), then CA MIA on the guest informs CA MIA for z/VM on VMH that the device is busy. CA MIA for z/VM on VMH does not allow the attach to proceed. If the device is not busy on a guest, then CA MIA for z/VM on VMH attaches the device and informs CA MIA on the guests that the device is not available for use by the guests.

Note: The preceding configuration provides for sharing a device between CMS users on a host z/VM and an unlimited number of guests using only two paths to the device. If configuration constraints prohibit defining more than one path per device, then the Autopath solution uses the CA MIA for z/VM Autopath for z/VM feature in conjunction with the CA MIA Autopath feature to allow for sharing between CMS users and guest systems using only one path per device. With the Autopath feature, a device is automatically moved between guest and host systems using attach and detach commands based on need for the device. While the Autopath feature has the advantage of needing only one path, the disadvantage is that the device is moved between host and guest system by detaching and attaching the device. With the Shared Tape Support configuration, the device remains attached to all guests all the time. There is no need to actually attach and detach the device. The disadvantage of shared tape support is that two paths are required. The Autopath feature is discussed in greater detail later in this section.

More information:

[How You Define Multiple Addresses per Device](#) (see page 86)

Guests Attached by MULTluser Option

You must specify the GTAINIT ASSIGN=NOASSIGN parameter if any device managed by CA MIA is attached to the guest using the MULTluser option.

Devices attached to guests using the MULTluser option do not currently support MULTISYSTEM ASSIGN. Control Program (CP) has been enhanced to support only ASSIGN and UNASSIGN processing types for MULTluser attached devices. If a MULTluser attached device receives MULTISYSTEM ASSIGN information, then an I/O error is returned.

Example:

If a z/OS guest of z/VM sends a MULTISYSTEM ASSIGN command to a MULTluser attached device, then the guest receives an I/O error for the device from the z/VM host. If the guest specifies the GTAINIT ASSIGN=MULTISYSTEM parameter, then CA MIA changes specifications for managed devices from ASSIGN to MULTISYSTEM ASSIGN.

Devices are assigned at VARY ONLINE time; therefore, when a VARY ONLINE command is issued, any MULTluser attached device managed by CA MIA receives an I/O error from the z/VM host, and the device is not brought online. Similarly, if the GTAINIT ASSIGN=ASIS parameter is specified and a VARY ONLINE,SHR command is issued for a MULTluser attached, managed device (appending the SHR option requests MULTISYSTEM ASSIGN), then an I/O error is received from the z/VM host, and the device is not brought online.

To eliminate I/O errors, specify the GTAINIT ASSIGN=NOASSIGN parameter.

Note: MULTISYSTEM ASSIGNS (VARY ONLINE,SHR) that are run against non-managed devices or against devices on guests where CA MIA is not running receive I/O errors if the devices are attached to the guest as MULTluser. IBM Shared Tape Support imposes this MULTISYSTEM ASSIGN restriction, not CA MIA.

Important! The value of the GTAINIT ASSIGN parameter must be the same on all systems in the CA MIA complex for ASSIGN processing to work correctly. If you are changing the GTAINIT ASSIGN= parameter, then perform a GLOBAL SHUTDOWN. CA MIA will shut down on all systems in your complex, including systems that are not guests under z/VM. All tape devices managed by CA MIA should be offline on all systems with no tape activity to the devices, to clear the ASSIGNS from all of the devices. The MIMINIT member is updated to specify the GTAINIT ASSIGN=NOASSIGN parameter before restarting CA MIA on any system. Failure to follow this procedure can result in the inability to use tape devices on one or more systems in the complex.

If you are running in a CA MIA complex that includes both CA MIA and CA MIA for z/VM, then you should already be using the GTAINIT ASSIGN=NOASSIGN parameter on all systems, since the NOASSIGN parameter is required in a mixed CA MIA and CA MIA for z/VM environment.

How You Define Multiple Addresses per Device

Prior to the implementation of shared tape support, each device address could only be attached to one virtual machine at a time, requiring the definition of an independent real address for each virtual machine that shares the tape device. As discussed above, this restriction is eliminated with Shared Tape Support and can be avoided using the Autopath feature. However, you still need to define multiple addresses per device if you are not using the Autopath feature and if any of the following are true:

- You are running at a level of z/VM that does not support Shared Tape Support.
- You are running at a level of z/VM that does support Shared Tape Support but you do not plan to use the Shared Tape Support feature.
- You are using CA MIA and the Shared Tape Support feature to share devices between CMS users on a z/VM host system and guest systems using the two-path configuration described in the previous section.

To define multiple addresses per device, define paths to multiple channel interfaces on a single control unit as if there were multiple control units, each with a single channel interface.

Unsupported device definitions should be used so z/VM does not set its own path group ID on the drives. For example, if supported devices are defined on two paths, then both paths will have the same path group ID. If one of the z/OS systems goes offline, then both paths are disbanded. If the drives are defined as unsupported, then each z/OS guest defines its own path group ID. This way, the path to one z/OS guest is not affected by a reset of the other path.

Note that the drives defined as unsupported are not usable in z/VM environments, either by CMS users or by CP functions such as SPTAPE or MONITOR. You can define one path as a supported tape drive, such as 3480, and another path as unsupported to handle this type of situation.

IBM hardware permits up to eight paths to a single tape drive. In general, when you plan to run multiple z/OS guests under z/VM, only one of the addresses you define to z/VM for a device should be supported. All extra paths should be defined as unsupported and only be used by z/OS or other operating system guests, such as z/VM guest systems. For information on the configuration statements required for defining tape devices, see the documentation provided by your hardware vendor.

Part-time Guest Considerations

The previous sections describe how to attach to multiple guest systems when the guests are all full-time production systems. You may also want to support guests that only operate occasionally or only need the use of tape drives part-time, such as during a backup procedure. If CA MIA for z/VM is managing the devices on the z/VM host system, then the SYSTEM option on the MI ATTACH command can be very useful in this situation.

If you attach using the SYSTEM option, then two rules must be observed:

- The guest can be a z/OS system or a z/VM system, but it must be running CA MIA, and share the same control file as all other systems that share the drive.
- The virtual address you use to attach the drive must be related through the UNITS MIM file or MIMUNITS member at the guest level to the same global name by which the drive is known to all other systems. This means that the global name of the drive on the guest system must be the same as the global name known to all other systems.

If either rule is violated, then dual allocation of the drive can occur. The second requirement is the same as the requirement placed on drives that are permanently attached to the guest systems. It is mentioned here because the manual use of the ATTACH command is more prone to error than other attach methods, such as a DEDICATE statement in the z/VM directory.

When you use the SYSTEM option, the guest system participates as a peer with all other systems that share the same drive. If the guest allocates the drive, then it blocks allocation on all other systems. This implies that before the guest can allocate, the drive must be free on all other systems.

The key feature of the SYSTEM option is that it prevents the drive from looking allocated due to the ATTACH command at the z/VM host level. If the SYSTEM option were not used in this situation, then neither the guest system nor any other system would be able to use the drive, because the attach to the guest would count as an allocation of the device.

The Autopath Feature

The Autopath feature allows you to share a device between multiple first level z/OS guest systems of the same z/VM host using only one path for the device. When used in conjunction with Autopath for CA MIA for z/VM, the device can also be shared with z/VM users on the z/VM host and first level z/VM guests using only one path.

Note: For information on the Autopath for z/VM feature, see the *CA MIA for z/VM Programming Guide*.

How Autopath Works

Autopath reacts to allocation recovery messages that occur due to a tape drive resource shortage in a z/OS guest. The stages of the Autopath process are:

1. A tape request enters allocation recovery on z/OS due to a shortage of tape drives.
2. Autopath examines the offline device list for Autopath-eligible devices.
3. Autopath selects the first Autopath-eligible device that does not already have an outstanding Autopath request and is not already attached to the local system.
4. Autopath attempts to attach the device. If the device is successfully attached to the local system, Autopath varies the drive ONLINE.

Note: When Autopath on one system detaches a drive for use by another system, it first VARYs the drive OFFLINE, if it is online.

Autopath Guidelines

The following is a list of guidelines for implementing the Autopath feature:

- If the CA MIA component is running on the host z/VM system, Autopath sends commands from the z/OS GUEST systems, running CA MIA with Autopath enabled, to CA MIA running on the z/VM host, requesting CA MIA managed devices be attached or detached to and from the requesting z/OS guest system.
- If you use native z/VM CP commands (meaning drives are not managed by CA MIA for z/VM on the host), it is not necessary to use the CA MIA for z/VM component to take advantage of the Autopath feature. Autopath on the z/OS guest uses CP commands to attach and detach drives.
- If you use CP commands for some drives, the CA MIA component for some drives, and leave other drives outside of the control of Autopath, you can use the Autopath feature-Autopath supports this configuration.
- If you have a system with drives connected to both native z/OS systems and to z/VM guest z/OS systems, Autopath can be used by the guest systems.
- If you have only native z/OS systems, you cannot use the Autopath feature.

Two things affect the way Autopath operates:

- **Information in the MIMUNITS member**

This information defines whether Autopath is to be used for each drive, and whether to use CP commands or a host resident release of CA MIA.

Note: If you are using both Autopath for z/OS and Autopath for z/VM, then devices must be defined as Autopath managed in both the MIMUNITS member for z/OS and the UNITS MIM file for z/VM.

- **Optional Autopath settings**

All of these settings are accessible through the SETOPTION GTAF AUTOPATH= command. In addition, all of these settings can be individually changed for each z/OS system.

More information:

[How You Identify Devices to CA MIA](#) (see page 30)

Autopath Assumptions

You can make the following assumptions about the Autopath feature:

- It is designed to improve average availability of tape drives over a long period of time.
- It should help provide the most resources to the busiest systems.
- In the short run, there is no guarantee that any specific job will benefit from Autopath. This feature is designed make more devices available on busier systems, not to move specific devices for specific jobs.

Note: We recommend that you not allow allocation recovery messages to remain outstanding while waiting for a device to be attached. This delays allocation and there is no guarantee that a device will be available to be attached.

- In some cases, certain system conditions can prevent Autopath from obtaining needed tape drive resources. For example, Autopath may be unable to act when two or more systems frequently invoke allocation recovery.

Note: For more information, see the *CA MIM Statement and Command Reference Guide*.

Chapter 4: Troubleshooting

This chapter provides commands and procedures for diagnosing and resolving allocation problems and hangs.

This section contains the following topics:

[How You Obtain Information for Diagnosing Problems](#) (see page 91)

[Tape Device Allocation Problems](#) (see page 92)

[Delay Detection and Notification](#) (see page 97)

[Tape Device Allocation Recovery Problems](#) (see page 99)

[Tape Device Preferencing Problems](#) (see page 102)

[Tape Device Dual Allocation Problems](#) (see page 103)

[How You Activate Tracing](#) (see page 107)

[How You Obtain Dumps](#) (see page 108)

[How to Restart CA MIA in an ACTIVE MIAplex](#) (see page 109)

[Contact Technical Support](#) (see page 117)

How You Obtain Information for Diagnosing Problems

You can use the CA MIA DIAGNOSE command to assist in diagnosing the cause of allocation delays. The DIAGNOSE command returns message MIM2150, which displays one of the following:

- When you issue a DIAGNOSE JOBSTATUS command, you receive information about jobs currently in allocation and their lock status, DELAYED, GIVEN, RELEASED, WAITING, or WAITING FOR DEVICES.
- When you issue a DIAGNOSE SYSTEMS command, you receive information about devices locked on each system in the complex.
- When you issue a DIAGNOSE EDT command, you receive information about the Eligible Device Table (EDT).
- When you issue a DIAGNOSE GROUPS command, you receive information about z/OS device groups.
- When you issue a DIAGNOSE VARY command, you receive information regarding the status of active and pending VARY device requests.
- When you issue a DIAGNOSE ALL command, you receive the information from the SYSTEMS, JOBSTATUS, and VARY displays.

Tape Device Allocation Problems

When you experience an allocation problem, use the information in this section to diagnose and resolve the problem. If you cannot resolve the problem yourself, then contact CA Technical Support for assistance. Most device allocation problems you may encounter fall into one of the following categories:

- A job allocates a different device than you expected.
- TPCF responds to z/OS allocation recovery messages in a different way than you expected.
- A dual allocation occurs.
- Allocation stops, hangs, or is delayed on all systems.

Overview of GTALOCAL Enqueue Processing

During initialization, CA MIA issues an exclusive enqueue with a QNAME of GTALOCAL and an RNAME of UP. It holds this resource for the entire duration that the address space is active. This enqueue resource prevents multiple versions of CA MIA from running concurrently on the same system. If the enqueue resource cannot be obtained during initialization processing, the CA MIA address space terminates.

The GTALOCAL enqueue resource is also used as a form of resource management by some of the CA MIA system intercepts. Specifically, those system intercepts that execute in a user's address space, and that make a request of the CA MIA address space and wait for a response, make use of a GTALOCAL enqueue resource during the time they are suspended. The intercept issues a shared request for an enqueue resource that is always held exclusively by the CA MIA address space. The intercept waits for a response to its request from the CA MIA address space or for shared control of the GTALOCAL enqueue resource. Should some event terminate the CA MIA address space that prevents normal post-back and clean-up processing, such as an operator FORCE command, the intercepts are redispached from their wait condition and discover that they have received control for the GTALOCAL enqueue resource. This prevents the intercepts from remaining in an enabled wait, and permits them to continue processing with the knowledge that the CA MIA address space has terminated.

The GTALOCAL enqueue resource RNAME is used to identify the process involved in the intercept wait condition. The current RNAMEs in use are:

- `Waiting_For_HCD_ACTIVATE_Completion`
- `Waiting_For_EDT_Synchronization`
- `Waiting_For_TAPE_SWAP_Completion`
- `Waiting_For_Device_Group_Locks`

Normally, the amount of time an intercept waits for a request to be serviced by the CA MIA address space is minimal, thus the resource management enqueue conflict between the intercept executing in a user's address space and the CA MIA address space rarely surface. However, there are circumstances that can cause the resource management GTALOCAL enqueue conflict to become visible. For example, a job in allocation recovery processing holds device group locks during the time that the MIM2060 (or IEF238D) WTOR is outstanding. The CA MIA address spaces on all systems maintain global knowledge of the device group locks. Should any job on an external system request any of the device group locks held by the job in allocation recovery, the requesting job is required to wait until such time that the desired device group locks are available. While any job requesting device group locks is waiting for a response to continue from the CA MIA address space, that job remains in enqueue conflict. The conflict shows the CA MIA address space having exclusive ownership of the enqueue resource QNAME, GTALOCAL, and RNAME, `Waiting_For_Device_Group_Locks` while jobs waiting for shared ownership of the named enqueue resource are, in reality, waiting for the CA MIA address space to grant them control of the requested device group locks. Once the allocation recovery event is satisfied and ownership of the device group locks can be granted by the CA MIA address space, the waiting jobs are re-dispatched and will drop their request for the resource management enqueue, thus, eliminating the enqueue resource conflict.

CA MIA also makes use of enqueue resources to control intersecting processes between intercepts in a user's address space and the CA MIA address space. Again, the GTALOCAL enqueue resource RNAME is used to identify the process involved in the intercept wait condition.

Diagnosing Allocation Issues

Allocation delays identified through GTALOCAL enqueue conflicts can be diagnosed by using the following procedures.

1. Issue the following command to identify the system causing allocation delay.

```
F MIM,DIAGNOSE SYSTEMS
```

Is there a system listed as holding locks?

YES

Go to Step 2.

NO

Have MIM2162W and MIM2165W or MIM2207I messages been issued?

YES

Issue the following z/OS command to determine why the resource is being held:

```
D GRS,RES=(<resource name>,*)
```

Where <resource name> is either SYSIEFSD or SYSZIOS as is shown in the MIM216x message.

If the resource in the message is SYSIEFSD DDRTPUR. Issue the following z/OS command to check for outstanding SWAP WTORS:

```
D R,L,CN=(ALL)
```

Is there an outstanding SWAP WTOR?

YES

Respond to the WTOR and follow swap to completion.

NO

If you cannot determine why the job holding the SYSIEFSD DDRTPUR resource is causing the allocation delay, go to Step 6.

NO

Problem circumstances may have changed. If you come to this point repeatedly and still believe there is a problem, contact CA Technical Support.

Return to Step 1.

2. On the system holding the locks identified in Step 1 issue the following CA MIA command:

```
F MIM,DIAGNOSE JOB
```

Has any job been given ownership of device group locks?

YES

Go to Step 3.

NO

Is the CA MIA task waiting for device group locks?

YES

Go to Step 3.

NO

- a. Issue the following CA MIA command:

```
SYSDUMP JOBNAME=(ALLOCAS,IOSAS,GRS) TITLE="ALLOCATION HANG"
SYSTEM=ALL
```

- b. Issue the following CA MIA command on this system:

```
... SHUTDOWN LOCAL
```

- c. Restart CA MIA on this system.

If the problem is not resolved, call CA Technical Support and ask for help with CA MIA.

3. Issue the following z/OS command:

```
D R,L,CN=(ALL)
```

Are there any outstanding SWAP messages or any outstanding allocation recovery messages?

YES

Reply to the outstanding WTOR.

NO

Go to Step 4.

4. Issue the following CA MIA command:

```
F MIM,DIAGNOSE VARY
```

Are there any active VARY requests that show a Wait-RSN of "VARY Dev?"

YES

- a. Check to see if the devices involved in outstanding VARY processing are functional. If the problem is not resolved, issue the following command:

```
F MIM,SYSDUMP JOBNAME=(ALLOCAS,IOSAS,GRS) TITLE="VARY DELAY"
```

- b. Issue a the following command to CA MIA:

SHUTDOWN LOCAL

If the problem is not resolved, contact CA Technical Support and ask for help with CA MIA.

NO

Go to Step 5.

- 5. Do the following:
 - a. Look in the system log for IOS003A INTERVENTION REQUIRED messages for tape devices. "Ready" any tape device mentioned in these messages. Allocation hangs can occur when jobs perform AVR processing on devices that are experiencing I/O problems.
 - b. Review the system log for I/O error messages for tape devices. If any have been issued for devices that do not appear to be allocated, VARY those devices OVERGENNED. Allocation hangs can occur when jobs perform AVR processing on devices that are experiencing I/O problems.
 - c. If I/O errors are not the cause of the allocation delay, go to Step 6.

- 6. Issue the following z/OS command to obtain related z/OS allocation enqueue information:

D GRS,RES=(SYSIEFSD,*)

Retain the information in the resulting display for diagnostic purposes.

Go to Step 7.

- 7. Get an SVC dump of problem jobs identified in Step 1 or Step 2 by issuing the following command:

F MIM,SYSDUMP JOB=(jobname,ALLOCAS,ISOAS,GRS),TITLE='ALLOCATION DELAY',SYSTEM=ALL

Note: When the DIAGNOSE display shows a job ID of MASTER as being 'delayed for' or 'given' a single device group lock, this most likely represents a job that is in the middle of SWAP processing. Look for the MIM2207I message to determine the actual job associated with the SWAP. Then, *issue the preceding dump* command for two job names - the job name for the MASTER address space and the swapping job name (JOB>(*MASTER*,jobname,ALLOCAS,IOSAS,GRS)).

Can this job be canceled?

YES

Cancel the job.

NO

Issue a SHUTDOWN LOCAL command to CA MIA. If the problem is not resolved, call CA Technical Support and ask for CA MIA.

Go to Step 8.

8. Gather any dumps taken, SYSLOGs for the problem time period, along with traces and job logs of any canceled jobs, and contact CA Technical Support during normal business hours to receive an incident number for this problem.

Important: If the problem is recurring, then adequate documentation is essential in solving the problem. Technical Support needs GTAF trace information before they can analyze the problem. Follow these steps:

- a. Dynamically allocate a MIM trace data set or SYSOUT using the ADDLOG command.

- b. To enable internal MIA tracing, issue the following commands:

```
F MIM,SET GTAF RESETTRACE=(LOCKS,MASKS,SWAP,VARY,UNITALOC)
F MIM,SET GTAF SETTRACE=(LOCKS,MASKS,SWAP,VARY,UNITALOC)
F MIM,SET GTAF RESETPRINT=(LOCKS,MASKS,SWAP,VARY,UNITALOC)
F MIM,SET GTAF SETPRINT=(LOCKS,MASKS,SWAP,VARY,UNITALOC)
F MIM,SET MIM RESETTRACE=TRANSACTION,RESETPRINT=TRANSACTION
F MIM,SET MIM SETT=TRANSACTION=(07,10,14),SETP=TRANSACTION
```

- c. Issue the following command to start the traces:

```
F MIM,SET TRACE=ON
```

- d. Allow the TRACES to run long enough to capture the problem event.

- e. Issue the following command to stop the traces:

```
F MIM,SET TRACE=OFF
```

9. Restart the MIA address space.

Delay Detection and Notification

Use the information in this section to diagnose and respond to Delay Detection and Notification (DDN) warnings.

Contention Detection

Possible Cause:

Received a WAIT/HOLD reply

The response of Wait/Hold was issued to an outstanding allocation recovery message.

Solution:

Provide more eligible devices, wait for devices to be deallocated from other tasks, or cancel the task.

Note: Request more information by issuing the following command:

```
ANALYZE TAPEDELAY DETAIL (JOBNAME=jobname, TYPE=CONTENTION)
```

Possible Cause:

An outstanding WTOR REPLYID *nnnn*

An allocation recovery message with the shown reply ID is awaiting a response. Device allocation group locks are held until a message response is received.

Solution:

Respond to the allocation recovery message.

Note: Request more information by issuing the following command:

```
ANALYZE TAPEDELAY DETAIL (JOBNAME=jobname, TYPE=CONTENTION)
```

Possible Cause:

Unable to Determine

A task has held locks for an extended amount of time but CA MIA is unable to determine the reason for the delay.

Solution:

- On the system of the owning task, look for outstanding allocation recovery messages.
- VARY devices online that can satisfy the tasks request.
- If no reason for the delay can be determined, canceling the task releases held device allocation group locks.
- If the task cannot be canceled or the device allocation group locks are not released, then contact CA Technical Support.

Note: Request more information by issuing the following command:

```
ANALYZE TAPEDELAY DETAIL (JOBNAME=jobname, TYPE=CONTENTION)
```

Wait/NoHold Detection

Possible Cause:

Excessive Wait/Nohold Replies

Two or more tasks are in allocation recovery for two or more devices. When one task replied to allocation recovery with WAIT/NOHOLD, the second task is posted and enters allocation recovery. No tasks on external systems are able to allocate devices from the device group.

This warning is issued when the problem persists for an extended time.

Solution:

To determine tasks contributing to the problem, issue the following command:

```
ANALYZE TAPEDELAY DETAIL (TYPE=WAITNOHOLD)
```

To alleviate the problem, add devices for the tasks to allocate or cancel tasks causing the delay.

If the problem cannot be resolved, contact CA Technical Support at <http://ca.com/support>.

Tape Device Allocation Recovery Problems

This use the information in this section to diagnose and respond to tape device allocation recovery/AUTOREPLY problems.

Note: CA MIA provides tracing that can assist in diagnosing problems in device selection recovery. You activate tracing through the DEVSEL24, DEVSEL78, and RECOVERY parameters on the SET TPCF SETTRACE/SETPRINT command.

TPCF Does Not Reply WAIT to the z/OS IEF238D Message

Possible Cause:

WAIT is not an option on the z/OS IEF238D message. Therefore, TPCF cannot reply WAIT.

Solution:

No action is required.

z/OS Issues Message IEF238D for a Job that Should Have Allocated An Online Device

Possible Cause:

The criteria defined for eliminating devices are too restrictive. Because all online devices were eliminated, the job enters allocation recovery and z/OS issues message IEF238D.

Solution:

Review the criteria defined in the TPCEDLXT exit routine. Use the DISPLAY JOB command to see information about the jobs for which devices are reserved.

If the criteria defined are too restrictive, then make one of these types of device selection criteria, or both types of criteria, less restrictive.

An OFFLINE Device That Should Not Be Used Appears In The OFFLINE Device List

Possible Cause:

The appropriate preferencing status was not assigned to the device.

Solution:

Assign the device overgenned or not-available status through the VARY command.

TPCF Does Not Respond to z/OS IEF238D Message

Possible Cause:

TPCF is not set up to automatically respond to this message.

Solution:

Initiate the TPCF AUTOREPLY feature with the command SETOPTION AUTOREPLY=ON.

Possible Cause:

The reply limits for ALLOC_OFFLN and SPEC_WAIT parameters in SYS1.PARMLIB(ALLOC00) member are set too low.

Solution:

Modify the MAXNWAIT parameters for ALLOC00, or use the TPCF AUTOREPLY options to override the z/OS values.

Possible Cause:

The job may also produce a diagnostic dump with the title 'FAILURE IN MIA ALLOCATION RECOVERY PROCESSING, CALLER NOT IDENTIFIED.'

Solution:

Contact CA Technical Support if you receive this dump.

Message MIM2042 Does Not List All of the Offline Devices the Job Can Use

Possible Cause:

TPCF is eliminating certain devices due to the preferencing values assigned to those devices. Devices can be eliminated by job reserve, overgennd status, tape robot software, and TPCEDLXT. Also, under certain conditions, TPCF removes NOTAVAILABLE devices and externally DEDICATED devices from the offline device list.

Solution:

No action is required.

More information:

[Advanced Topics](#) (see page 21)

NOTAVAILABLE or Externally DEDICATED Devices Are Appearing In The Offline Device List

Possible Cause:

TPCF includes these devices on the offline device list if no other device is available and SETOPTION AUTOREPLY=(NOWAIT=(EXTDEDDISP=YES,NOTAVLDISP=YES)). TPCF includes externally DEDICATED devices first; TPCF includes NOTAVAILABLE devices only if there are no externally DEDICATED devices.

Solution:

For information about how TPCF handles not-available and dedicated devices during allocation recovery, see the chapter "Advanced Topics."

z/OS Issues Message IEF700I

This message indicates that the job is abending with a JCL error

Possible Cause:

The reply WAIT,HOLD was specified for message MIM2060 and some devices in the EDL of the job were reserved for other jobs.

Solution:

Reply WAIT,NOHOLD to message MIM2060 during job reserve.

Tape Device Preferencing Problems

Use the following sections to diagnose and respond to tape device preferencing problems.

Note: CA MIA provides tracing that can assist in diagnosing problems in device selection recovery. You activate tracing through the DEVSEL24, DEVSEL78, and RECOVERY parameters on the SET TPCF SETTRACE/SETPRINT command.

Ineligible Device Allocated

Symptom:

A job allocates a device that should have been eliminated by a user exit routine or a device that is reserved for another job.

Possible Causes:

The criteria defined for eliminating devices is too restrictive. Because the criteria would have eliminated all suitable devices, TPCF ignored the criteria.

Solution:

Make the criteria less restrictive. Review the criteria defined in the exit routine. Use the JOBRESERVE parameter on the DISPLAY command to see information about reserved devices.

Less-preferred Device Allocated

Symptom:

A job allocates a less preferred device when a preferred device is available.

Possible Causes:

TPCF is forced to choose from a group of less preferred devices, due to one or more of these reasons:

- z/OS ACL processing is eliminating more preferred devices
- z/OS generic preferencing, which is created during the SYSGEN process or by HCD, is forcing TPCF to choose from a group of less preferred devices.
- Preferencing done by another software package, for example, a robotic software package, is overriding TPCF preferencing.

Solution:

You must dynamically change the ACL status of your devices or change your preferencing setup.

If another software package is overriding TPCF preferencing, then you may need to contact the software vendor for information on customizing the preferencing of that product.

Tape Device Dual Allocation Problems

Dual allocation occurs when the same device is allocated on two different systems at the same time. GTAF prevents dual allocation in a multiple-system environment by communicating allocation information among systems and extending locking mechanism z/OS uses to serialize allocation. However, under certain conditions, a dual allocation can occur while GTAF is running.

Most dual allocations occur because devices are not properly placed under CA MIA management. Another common cause is the way other software products interact with z/OS and GTAF. If you cannot attribute the dual allocation at your site to one of these possible causes, then you should contact CA Technical Support for assistance.

Possible cause:

GTAF is not active.

Solution:

Use the FACILITIES parameter on the DISPLAY command to see whether GTAF is active. If GTAF is not active, then do the following:

- Place the MIMINIT GTAF=ON statement in the initialization member to start GTAF automatically during CA MIM initialization.
- Manage the device manually until the next time CA MIM is started.

Possible cause:

The device was never placed under CA MIA management.

Solution:

Use the DISPLAY GLOBALUNITS command or inspect the MIM2004 message issued at startup/resynch to see whether CA MIA is managing the device. If CA MIA is not managing the device, then do the following:

- Place the device under CA MIA management through the MIMUNITS member.
- Assign a global name to the device, if necessary.
- Force CA MIA to resynchronize by issuing the RESYNCH command.

Possible cause:

The device was recently defined to the system during the SYSGEN process or using HCD, but not placed under CA MIA management.

Solution:

Use the DISPLAY GLOBALUNITS command or inspect the MIM2004 message issued at startup/resynch to see whether CA MIA is managing the device. If CA MIA is not managing the device, then do the following:

- Place the device under CA MIA management through the MIMUNITS member.
- Assign a global name to the device, if necessary.
- Issue the command SETOPTION GTAF RESYNCH SAMEDEVS=NO and change this value in the MIMCMNDS member to SAMEDEVS=NO for the next restart of CA MIA.
- Force CA MIA to resynchronize by issuing the RESYNCH command.

Possible cause:

The local name of the device was misspelled in the MIMUNITS member. As a result, GTAF is not managing the desired device.

Solution:

Review the MIM2004 messages to see whether the local name of the device was misspelled. You also can use the GLOBALUNITS parameter on the DISPLAY command to obtain this information.

If the local device name is misspelled, then do the following:

- Correct the entry in the MIMUNITS member.
- Force CA MIA to resynchronize by issuing the RESYNCH command.

Possible cause:

The device was not placed under CA MIA management on all systems. This is due to one of the following:

- MIMINIT statements were directed to only some of the systems in the complex. As a result, CA MIA is not managing the device on all systems.
- Device entries in the MIMUNITS member were directed only to certain systems, but not to all systems in the complex. As a result, CA MIA is not managing the device on all systems.
- Separate device control members are being used for each system, and this device was not included in every member.

Solution:

Review the contents of the CA MIM initialization member, which contains the MIMINIT statements, and each MIMUNITS member in use to determine where the error has occurred.

If CA MIA is not managing the device on all systems, then do the following:

- Add MIMINIT statements in the initialization member or add device entries to the MIMUNITS member to place devices under CA MIA management on the remaining systems.
- Assign global device names, if necessary.
- Force CA MIA to resynchronize by issuing the RESYNCH command.

Possible cause:

Global names were not assigned to every device that needs one.

Solution:

When global names are not specifically assigned, GTAF uses the local name of a device as its global name. This can cause dual allocation if two different devices use the same local name on different systems (for example, local name 01A0 refers to one device on System A and a different device on System B). Dual allocation also can occur if the device is allocated under different names on different systems (for example, the same device is allocated under local name 02A0 on System A and 03A0 on System B).

Review the MIM2004 messages to determine where global device names are required.

If global names were not assigned as required, then do the following:

- Assign global device names through the MIMUNITS member.
- Force CA MIA to resynchronize by issuing the RESYNCH command.

Possible cause:

A global name was assigned to that device; however, the global name is identical to the local name of a different device.

Solution:

Use the GLOBALUNITS parameter on the DISPLAY command to see the global and local names of all devices CA MIA is managing.

If the global name for that device is not unique, then do the following:

- Assign a different global device name through the MIMUNITS member.
- Force CA MIA to resynchronize by issuing the RESYNCH command.

Possible cause:

The same global name was not assigned to that device on all systems.

Solution:

Review the MIM2004 messages to see where the error occurred.

If the global device name is different on different systems, then do the following:

- Assign the same global device name on all systems through the MIMUNITS member.
- Force CA MIA to resynchronize by issuing the RESYNCH command.

Possible cause:

Another software product is in use that is bypassing the z/OS allocation process.

GTAF does not control allocation. It simply communicates information about allocations that have occurred during the z/OS allocation process. When software bypasses the z/OS allocation process, GTAF is not aware of those device allocations. Therefore, GTAF cannot prevent dual allocation.

Solution:

Do not use the software while GTAF is active.

More information:

[Resynchronization: RESYNCH Command](#) (see page 39)

How You Activate Tracing

You can activate event tracing for the GTAF and TPCF facilities using the following commands. To activate tracing, you must first issue the SETOPTION MIM TRACE command.

Note: For more information, see the *CA MIM Statement and Command Reference Guide*.

Activate the TRACE Feature

Use the command SETOPTION MIM TRACE=*options* command to activate and deactivate the TRACE feature. Once this is specified, you can begin tracing for GTAF and TPCF. The command also lets you deallocate the existing MIM trace data set and allocate a new SYSOUT data set dynamically, and lets you limit some types of tracing by job name. By default, tracing is off.

Activate Printing for a Specific Facility

Use the command SETOPTION *facility* SETPRINT=(*options*) to activate the printing function for the specified facility and selected options.

The SETPRINT parameter controls whether events that are recorded in the CA MIM internal trace table are also sent to the MIM trace data set. SETPRINT is dependent upon SETTRACE, because only operands that have been specified in both SETTRACE and SETPRINT are sent to the MIM trace data set.

Activate the Tracing Function for a Specific Facility

Use the command `SETOPTION facility SETTRACE=(options)` to activate the tracing function for the specified facility and selected options. The SETTRACE parameter enables the recording of specific program events in the CA MIM internal trace table.

To record trace event information for TPCF allocation recovery, issue the following commands:

```
SETOPTION TPCF SETTRACE=RECOVERY
SETOPTION TPCF SETPRINT=RECOVERY
SETOPTION MIM TRACE=ON
```

Tracing is activated and records are placed in the MIM trace data set or spool file.

Reverse Tracing Settings for a Specific Facility

Use the command `SETOPTION facility RESETTRACE` to reverse or turn off settings that were previously made using SETTRACE. After RESETTRACE is issued, the options you specify are no longer recorded in the CA MIM internal trace table.

Reverse Printing Settings for a Specific Facility

To reverse printing settings

Issue the following command to reverse or turn off settings that were previously made using SETPRINT:

```
SETOPTION facility RESETPRINT
```

facility

Specifies the facility for which you want to set this option.

After RESETPRINT is issued, the options you specify are no longer recorded to the MIM trace data set, but continue to be stored in the CA MIM internal trace table, if RESETTRACE has not been issued.

Note: For more information, see the *CA MIM Statement and Command Reference Guide*.

How You Obtain Dumps

This section discusses how to obtain dumps for diagnostic purposes.

Obtain a Dump of Facility Activity

To obtain a facility-specific dump

Issue the following command:

```
DUMP facility
```

facility

Specifies the facility for which you want the dump. Valid values are GTAF or TPCF

This command is for diagnostic purposes only. Use the DUMP commands only when directed by CA Technical Support.

Obtain a System Dump

The SYSDUMP command lets you obtain a system dump of CA MIM and optionally, a list of other named address spaces. You can request the dump on the issuing system, all systems in the CA MIM complex, and all external systems to the issuing system, or a list of selected systems in the complex.

Keep the following in mind:

- You must issue the SYSDUMP command from a console or a TSO session. You cannot issue this command from the MIMPARMS data set.
- You must be authorized to issue system control commands in order to issue the SYSDUMP command. TSO users generally are not authorized to issue system control commands.

To obtain a system dump

Issue the following command:

```
SYSDUMP COLLECT=MAXIMUM SYSTEM=sysname
```

sysname

Specifies the name of the system for which you want the dump.

Note: For more information, see the *CA MIM Statement and Command Reference Guide*.

How to Restart CA MIA in an ACTIVE MIAplex

During shutdown, CA MIA takes all locally managed unallocated Tape Devices OFFLINE. Locally managed, allocated Tape Devices are marked Pending OFFLINE, and transition to an OFFLINE state when the local allocation of the device completes.

CA MIA takes this action to preserve the integrity of data residing on devices that are currently allocated on other systems in the MIAplex.

If Tape Device Allocation is not QUIESCED prior to CA MIA SHUTDOWN on a system, TASKs attempting to allocate Tape Devices on the system will enter Recovery Allocation. During Recovery Allocation, WTORs are issued listing the set of options available for completing the current allocation request.

While a Recovery Allocation WTOR is outstanding, the TASK for which it is issued holds important system resources that serialize Recovery Allocation processing with other z/OS processes.

For example, while a Recovery Allocation WTOR is outstanding on a system:

- LOCAL VARY OFFLINE processing will be delayed.
- LOCAL Allocation requests for devices within the ESOTERIC or GENERIC UNIT types will be delayed.
- LOCAL HCD Activate completion will be delayed.

CA MIA extends z/OS Tape Device Allocation Serialization across all systems in an MIAplex. As a result, z/OS Recovery Allocation related serialization on one system can affect VARY processing and Tape Device Allocation throughput within an MIAplex in the following ways:

- LOCAL system Recovery Allocations WTORs, while outstanding, will prevent EXTERNAL system VARY ONLINE completion for devices in the GENERIC or ESOTERIC UNIT involved in the Recovery.
- External system Recovery Allocations WTORs, while outstanding, will prevent LOCAL system VARY ONLINE completion for devices in the GENERIC or ESOTERIC UNIT involved in the Recovery.
- ALL RECOVERY Allocation WTORs, while outstanding, will delay Tape Device Allocation throughput for the GENERIC or ESOTERIC UNIT involved in the Recovery on ALL systems in the MIAplex.

EXAMPLE 1: VARY ONLINE processing delayed by Recovery Allocation WTORs

This example shows the effect that an outstanding Recovery Allocation WTOR on system XE03 has on VARY ONLINE processing on an external system system in the MIAplex, XE13. The delay would occur on ALL MIAplex systems.

TASK TAPE1 is started on system XE03. It is unable to allocate a suitable, ONLINE UNALLOCATED device, and enters Recovery Allocation. The MIM2060 Recovery Allocation WTOR is issued and remains outstanding on XE03.

XE03 TASK TAPE1 is in recovery for UNIT=3490 Tape Devices

```

#MIM2120 TAPE1 TAPE1 UNABLE TO ALLOCATE
#MIM2042 TAPE1 TAPE1 TAPEDD DEVICES OFFLINE 068
=> 0740,0741,0742,0743,0744,0745,0746,0747,0760,0761,0762,0763
=> 0764,0765,0766,0767,0768,0769,076A,076B,076C,076D,076E,076F
=> 0E50,0E51,0E52,0E53,0E54,0E55,0E56,0E57,0E58,0E59,0E5A,0E5B
=> 0E5C,0E5D,0E5E,0E5F,0E80,0E81,0E82,0E83,0E84,0E85,0E86,0E87
=> 0E88,0E89,0E8A,0E8B,0E8C,0E8D,0E8E,0E8F,0E90,0E91,0E92,0E93
=> 0E94,0E95,0E96,0E97,0E98,0E99,0E9A,0E9B,0E9C,0E9D,0E9E,0E9F
=> 0E20,0E21,0E22,0E23,0E24,0E25,0E26,0E27,0E28,0E29,0E2A,0E2B
=> 0E2C,0E2D,0E2E,0E2F,0E70,0E71,0E72,0E73,0E74,0E75,0E76,0E77
=> 0E78,0E79,0E7A,0E7B,0E7C,0E7D,0E7E,0E7F
*0117 #MIM2060 TAPE1 - REPLY DEVICE NAME OR 'CANCEL'.

```

While the MIM2060 WTOR (or the z/OS equivalent) is outstanding, TASK TAPE1 holds the Tape Device Group Locks for each of the devices listed on the MIM2042 (or z/OS equivalent). Until a valid response to the Recovery Allocation WTOR is entered, these resources are unavailable on any other system in an active MIAplex. This condition will delay VARY ONLINE command processing and Tape allocation throughput on all MIAplex systems.

MIA on EXTERNAL system XE13 is started and synchronizes. The Recovery Allocation WTOR remains outstanding on XE03. VARY ONLINE commands for the devices involved in Recovery Allocation on XE03 (UNIT=3490) are issued during MIA synchronization on XE13. These commands will be delayed from completing because of the Recovery Allocation condition on XE03, as identified by a DIAGNOSE ALL command issued on XE13.

XE13

```
*#MIM0022I system XE13 in file 00 synchronization UNDERWAY
#MIM0023I system XE13 in file 00 synchronization COMPLETE
#VARY 740-747,ONLINE
#VARY 760-76F,ONLINE
#VARY E50-E5F,ONLINE
#VARY E80-E8F,ONLINE
#VARY E90-E9F,ONLINE
#VARY E20-E2F,ONLINE
#VARY E70-E7F,ONLINE
#MIM0067I Command VARY 310
#MIM2021I VARY process pending
#MIM0067I Command VARY 311
#MIM2021I VARY process pending
#MIM0067I Command VARY 312
#MIM2021I VARY process pending
#MIM0067I Command VARY 313
#MIM2021I VARY process pending
#MIM0067I Command VARY 314
#MIM2021I VARY process pending
#MIM0067I Command VARY 315
#MIM2021I VARY process pending
#MIM0067I Command VARY 316
#MIM2021I VARY process pending
```

The SYSTEMS DISPLAY portion of the MIM2150I command response to a DIAGNOSE ALL command issued on XE13 identifies that Tape Device Group Locks are in use on system XE03. The Tape Device Group Locks are held by TASK TAPE1, which is in Recovery Allocation on XE03.

```
#DIAGNOSE ALL
#MIM0067I Command DIAGNOSE 320
#MIM2150I DIAGNOSE ALLOCATION DISPLAY
BEGIN SYSTEMS DISPLAY =====>
System XE03 has 'AB' locks for devices
740 741 742 743 744 745 746 747 760 761 762 763
764 765 766 767 768 769 76A 76B 76C 76D 76E 76F
E20 E21 E22 E23 E24 E25 E26 E27 E28 E29 E2A E2B
E2C E2D E2E E2F E50 E51 E52 E53 E54 E55 E56 E57
E58 E59 E5A E5B E5C E5D E5E E5F E70 E71 E72 E73
E74 E75 E76 E77 E78 E79 E7A E7B E7C E7D E7E E7F
E80 E81 E82 E83 E84 E85 E86 E87 E88 E89 E8A E8B
E8C E8D E8E E8F E90 E91 E92 E93 E94 E95 E96 E97
E98 E99 E9A E9B E9C E9D E9E E9F
END OF SYSTEMS DISPLAY
```

The JOBSTATUS portion of the MIM2150I command response shows that the GTAF component of CA MIA, which is responsible for Global Serialization of Tape Device Allocation, is delaying requests by the MIA started Task on XE13 for +A+ Tape device Group Locks for 8 of the devices being VARY'd ONLINE, (740-747).

MIA is attempting to serialize these VARY ONLINE requests with z/OS Tape Device Allocation within the MIAplex. MIA cannot achieve the serialization because the Tape Device Group Lock resources required by MIA to do so are held by TASK TAPE1 on system XE03.

The +A+ designation of the Tape Device Group Lock request indicates that the request is for a VARY command.

```
BEGIN JOBSTATUS DISPLAY =====>
JOBSTATUS: DELAYED
    GTAF is delaying MIAB713's request for +A+ group locks for devices      740
    GTAF is delaying MIAB713's request for +A+ group locks for devices      742
    GTAF is delaying MIAB713's request for +A+ group locks for devices      743
    GTAF is delaying MIAB713's request for +A+ group locks for devices      745
    GTAF is delaying MIAB713's request for +A+ group locks for devices      746
    GTAF is delaying MIAB713's request for +A+ group locks for devices      747
    GTAF is delaying MIAB713's request for +A+ group locks for devices      741
    GTAF is delaying MIAB713's request for +A+ group locks for devices      744
JOBSTATUS: RELEASED
    NO ENTRIES FOR JOBSTATUS.RELEASED
JOBSTATUS: WAITING
    NO ENTRIES FOR JOBSTATUS.WAITING
JOBSTATUS: GIVEN
    NO ENTRIES FOR JOBSTATUS.GIVEN
JOBSTATUS: WAITING FOR DEVICES
    NO ENTRIES FOR JOBSTATUS.WAITING FOR DEVICES
END OF JOBSTATUS DISPLAY
```

Lastly, the MANAGED VARY portion of the MIM2150I command response shows the status of all VARY commands for MIA managed devices on the system where the DIAGNOSE ALL command was issued.

It identifies that VARY ONLINE requests for devices 0740-0747 are ACTIVE, but are being delayed by MIA from processing due to Tape Device Group Lock contention (Wait-Rsn: Grp Lock).

MIA on XE13 will requeue these 8 VARY commands, and attempt to process the next 8 SCHEDULED VARY ONLINE requests. If the Recovery Allocation WTOR remains outstanding on XE03, NO VARY ONLINE commands for the Group of Tape Devices (ESOTERIC or GENERIC) involved in Recovery Allocation on XE03 will complete until the Recovery Allocation WTOR is replied to.

```

Begin managed VARY display =====>
Active VARY device queue
Dev State Option Source Requeue# Queue-Tm Wait-Rsn
0740 ONLINE *INTVARY 1 7 Grp Lock
0741 ONLINE *INTVARY 1 7 Grp Lock
0742 ONLINE *INTVARY 1 7 Grp Lock
0743 ONLINE *INTVARY 1 7 Grp Lock
0744 ONLINE *INTVARY 1 7 Grp Lock
0745 ONLINE *INTVARY 1 7 Grp Lock
0746 ONLINE *INTVARY 1 7 Grp Lock
0747 ONLINE *INTVARY 1 7 Grp Lock

Scheduled VARY device queue
Dev State Option Source Requeue# Queue-Tm Wait-Rsn
0760 ONLINE *INTVARY 0 7
0761 ONLINE *INTVARY 0 7
0762 ONLINE *INTVARY 0 7
0763 ONLINE *INTVARY 0 7
0764 ONLINE *INTVARY 0 7
0765 ONLINE *INTVARY 0 7
0766 ONLINE *INTVARY 0 7
0767 ONLINE *INTVARY 0 7
.
.
.
0E7C ONLINE *INTVARY 0 8
0E7D ONLINE *INTVARY 0 8
0E7E ONLINE *INTVARY 0 8
0E7F ONLINE *INTVARY 0 8
End managed VARY display
    
```

EXAMPLE 2: LOCAL VARY OFFLINE processing delayed by Recovery Allocation WTORs

Outstanding Recovery Allocation WTORs will delay the completion of VARY OFFLINE command processing on the system where the WTOR is outstanding.

The delay is caused by z/OS ENQ Serialization of VARY OFFLINE processing with Allocation, NOT by MIA Global Serialization of the request.

While a TASK is in RECOVERY ALLOCATION, it holds a SHR ENQ for z/OS Allocation resource SYSIEFSD/Q4. z/OS VARY OFFLINE processing requires a SYSIEFSD/Q4 EXCL ENQ to serialize VARY OFFLINE with z/OS Allocation.

When processing a VARY OFFLINE request, z/OS will WAIT for 5 seconds to acquire the EXCL ENQ for SYSIEFSD/Q4. If the 5 second timer expires before the SYSIEFSD/Q4 EXCL ENQ can be obtained, z/OS will drop then attempt to reacquire the EXCL ENQ.

The effect is that during periods that a SYSIEFSD/Q4 ENQ is held for extended periods of time, as is the case with outstanding Recovery Allocation WTORs, VARY OFFLINE processing is delayed, and z/OS Allocation requests, which are serialized by z/OS with a SHR SYSIEFSD/Q4 ENQ, will be serviced at 5 second intervals.

Recovery Allocation related SYSIEFSD/Q4 ENQ contention delays z/OS VARY OFFLINE processing from completing for ALL device types while a Recovery Allocation WTOR is outstanding.

This example illustrates how Recovery Allocation related ENQ contention can impact LOCAL system VARY OFFLINE processing.

XE03 JOB in recovery for UNIT=3480

```
#MIM2120 TAPE1 TAPE1 UNABLE TO ALLOCATE
#MIM2042 TAPE1 TAPE1 TAPEDD DEVICES OFFLINE 068
=> 0740,0741,0742,0743,0744,0745,0746,0747,0760,0761,0762,0763
=> 0764,0765,0766,0767,0768,0769,076A,076B,076C,076D,076E,076F
=> 0E50,0E51,0E52,0E53,0E54,0E55,0E56,0E57,0E58,0E59,0E5A,0E5B
=> 0E5C,0E5D,0E5E,0E5F,0E80,0E81,0E82,0E83,0E84,0E85,0E86,0E87
=> 0E88,0E89,0E8A,0E8B,0E8C,0E8D,0E8E,0E8F,0E90,0E91,0E92,0E93
=> 0E94,0E95,0E96,0E97,0E98,0E99,0E9A,0E9B,0E9C,0E9D,0E9E,0E9F
=> 0E20,0E21,0E22,0E23,0E24,0E25,0E26,0E27,0E28,0E29,0E2A,0E2B
=> 0E2C,0E2D,0E2E,0E2F,0E70,0E71,0E72,0E73,0E74,0E75,0E76,0E77
=> 0E78,0E79,0E7A,0E7B,0E7C,0E7D,0E7E,0E7F
*0117 #MIM2060 TAPE1 - REPLY DEVICE NAME OR 'CANCEL'
```

While the MIM2060 Recovery Allocation WTOR is outstanding on XE03, VARY OFFLINE commands are issued for Tape Devices 078C-078F, but the commands do not complete.

Note: Devices 078C-078F are NOT involved in the outstanding Recovery Allocation on the system.

The Managed VARY display within the CA MIA DIAGNOSE ALL command response, issued on system XE03, provides insight into the cause of the delay in VARY OFFLINE processing.

```
Begin managed VARY display =====>
Active VARY device queue
Dev State Option Source Requeue# Queue-Tm Wait-Rsn
078C OFFLINE *INTVARY 1 5 VARY Dev
078D OFFLINE *INTVARY 1 5 VARY Dev
078E OFFLINE *INTVARY 1 5 VARY Dev
078F OFFLINE *INTVARY 2 5 VARY Dev
End managed VARY display
END OF DIAGNOSE COMMAND
```

The Wait-Rsn of VARY Dev, appearing for each of the VARY OFFLINE commands shown in the example above, indicates that MIA has passed the VARY OFFLINE requests to the z/OS VARY Service, IEEVARYD, for processing. MIA is waiting for the z/OS VARY Service to complete the OFFLINE requests and return control. MIA is not delaying the VARY OFFLINE commands.

A GRS display for the SYSIEFSD resource identifies SYSIEFSD resource utilization on system XE03. While TASK TAPE1 is in Recovery Allocation, it OWNS SHR ENQs on SYSIEFSD resources CHNGDEVS, DDRDA, DDRTPUR and Q4.

The MIA started task on XE03, MIAB703, WAITS EXCL for SYSIEFSD/Q4. This EXCL SYSIEFSD/Q4 ENQ, as well as the EXCL SYSIEFSD/VARYDEV ENQs below it, are raised by the z/OS VARY Service, IEEVARYD, on behalf of the caller, MIAB703. Until Recovery Allocation on XE03 is relieved, VARY OFFLINE completion and z/OS Allocation throughput for DASD or TAPE will be impacted.

```

D GRS,RES=(SYSIEFSD,*)

ISG343I 10.55.09 GRS STATUS 229
S=SYSTEM SYSIEFSD CHNGDEVS
SYSNAME      JOBNAME      ASID      TCBADDR  EXC/SHR  STATUS
XE03         TAPE1             0068      007FF890  SHARE    OWN
S=SYSTEM SYSIEFSD DDRDA
SYSNAME      JOBNAME      ASID      TCBADDR  EXC/SHR  STATUS
XE03         TAPE1             0068      007FF890  SHARE    OWN
S=SYSTEM SYSIEFSD DDRTPUR
SYSNAME      JOBNAME      ASID      TCBADDR  EXC/SHR  STATUS
XE03         TAPE1             0068      007FF890  SHARE    OWN
S=SYSTEM SYSIEFSD Q4
SYSNAME      JOBNAME      ASID      TCBADDR  EXC/SHR  STATUS
XE03         TAPE1             0068      007FF890  SHARE    OWN
XE03         MIAB703        0066      006E4108  EXCLUSIVE  WAIT
S=SYSTEM SYSIEFSD VARYDEV
SYSNAME      JOBNAME      ASID      TCBADDR  EXC/SHR  STATUS
XE03         MIAB703        0066      006E4108  EXCLUSIVE  OWN
XE03         MIAB703        0066      006E7900  EXCLUSIVE  WAIT
XE03         MIAB703        0066      006E44D8  EXCLUSIVE  WAIT
XE03         MIAB703        0066      006E4340  EXCLUSIVE  WAIT

```

Contact Technical Support

Procedures for gathering and sending diagnostic data and contacting CA Technical Support--whether by the Internet, telephone, fax, or FTP server--are included in the chapter "Troubleshooting" in the *CA MIM Programming Guide*.

Chapter 5: User Exits

This chapter describes the optional user exits and provides additional information about coding them.

This section contains the following topics:

[How You Manage User Exits](#) (see page 119)

[Exit Routine Programming Considerations](#) (see page 122)

How You Manage User Exits

CA MIA lets you use the following optional exit routines to influence device selection during allocation and recovery and to customize the processing of allocation recovery messages.

TPCEDLXT

This routine marks devices as ineligible in the device list, during allocation, and removes them from the offline device list, during allocation recovery. This routine is available only when you run TPCF.

TPCREXT

This routine customizes the TPCF processing for allocation recovery messages. This routine is available only when you run TPCF.

TPCSRMT

This routine customizes the TPCF device preferencing, when possible, during allocation. This routine is available only when you run TPCF.

Activate Exit Routines

To activate any exit routine, you specify the SETOPTION EXIT command. For example, to activate the TPCSRMXT, you would issue the command:

```
SETOPTION EXIT=TPCSRMT
```

If you use a different module name for the exit routine, then specify the same command with the different module name indicated on the LOAD parameter. For example, if the module name for the TPCSRMXT routine were PRLOGIC, then you would specify the command:

```
SETOPTION EXIT=(TPCSRMT, LOAD=PRLOGIC)
```

Note: For more information, see the *CA MIM Statement and Command Reference Guide*.

View Exit Routine Information

To see information about exit routines you are using, issue a DISPLAY EXIT command. CA MIM displays each logical exit routine name with its load module, address, status, and values set for protection and dump generation. The next screen shows the display for exit routine information:

F MIMGR, DISPLAY EXIT						
MIM0264	MIM	EXIT	DISPLAY			
EXIT	MODULE	ADDRESS	STATUS	PROT	DUMP	DISA
GCMCMDXT	USR	00006910	ACTIVE	YES	YES	NO
GCMCMDXT	USREX	00006920	INACT	YES	YES	NO
GCMSRCXT	USREXIT	00006930	INACT	NO	YES	NO
TPCRECXT	USREXIT	00006940	ACTIVE	NO	NO	YES

You can use the DISPLAY EXIT command to check on the status of an exit routine, except for MIMINIXT. Use the DISPLAY INIT command to find out whether the MIMINIT INITEXIT statement has been set.

MIMINIXT (Common Exit Interface)

CA MIM provides a common exit interface that you can use to dynamically manage any CA MIM exit routine.

Note: For more information see the *CA MIM Programming Guide*.

Rules for Coding Exit Routines

Follow these rules when coding CA MIA exit routines:

- Exits for CA MIA and other product components are invoked using a standard parameter list, pointed to by register 1.
- Assume standard linkage and register saving conventions.
- Preserve all registers except 0, 1, and 15.
- Make the TPCEDLXT, TPCSRMXT, and TPCRECTX routines reentrant. Because these routines do not hold local or CMS locks, which enable them to issue SVCs, multiple allocation requests may be invoked in parallel.
- CA MIM moves the exit routines above the 16-megabyte line if the RMODE value for the routines is 31.
- Assemble and link the routines into the authorized load library that contains the CA MIM load modules. The sample member named ASMEXIT in the CAI.CBTDJCL data set contains JCL that you can use to assemble and link-edit exit routines.

Notes on Using Exit Routines

Consider these points before you invoke a CA MIA exit routine:

- TPCF calls the exit routines whenever a device managed by CA MIA is in the EDL of a job or candidate list.
- You can use exit routines to supplement the other methods of TPCF influencing device selection, such as the VARY command.
- You cannot use the exit routines to add devices to the eligible device list, or to the offline device list.
- You can invoke more than one type of device control logic.
- TPCF will *not* call the TPCRECTX exit routine under two conditions:
 - The combination of job reserve, TPCEDLXT, robotic device allocation, and overgenning processing has eliminated all offline devices, and WAIT is not an option on the IEF238D WTOR, and SETOPT TPCF AUTOREPLY=(NOWAIT=(CANCEL=YES)). In such cases, TPCF replies CANCEL automatically to the IEF238D WTOR.
 - TPCEDLXT responded with a return code of 12. In these cases, TPCF replies CANCEL automatically to the IEF238D WTOR.
- TPCRECTX cannot add options to allocation recovery messages.

Sample User Exits

Sample members for each exit routine are provided in the CAI.CBTDSAMP data set. The DSECTS for the TPCREEXT are contained in the TPCRECD5 member of the CAI.CBTDMAC data set.

Exit Routine Programming Considerations

Use the guidelines in this section when coding exit routines using the samples included on your product tape.

TPCEDLXT Exit

The TPCEDLXT exit routine invokes *elimination logic*, which marks devices ineligible in the EDL (during allocation) and removes them from the offline device list (during allocation recovery). By defining elimination logic, you can remove certain devices from consideration in almost every condition. However, if the devices you remove are the only ones a job can allocate or if the only remaining devices are reserved for other jobs, TPCF ignores your elimination logic so that the job can allocate a device.

Note: Elimination logic does not affect selection of devices in SWAP processing.

Entry Environment:

Supervisor state, key 0, AMODE 31.

You must preserve this entry environment upon entry and restore it upon return.

Register Usage at Entry:

Common Parameter List

R1

This register contains the address of the parameter list, in this format:

+0

Contains the address of the parameter list of the TPCEDLXT exit routine.

+4

Contains the TPCEDLXT communication word. The content of this word is set by the exit during its initialization call and is passed unchanged for all subsequent calls.

+8

Contains the full-word of flags used to pass status information to the exit.

+C

Contains the global communication word.

R13

Contains the address of the standard save area.

R14

Contains the return address for CA MIM

R15

Contains the entry point address for the routine.

All other registers are undefined.

TPCEDLXT Exit-specific Parameter List

+0

This contains the address of the list of devices for this allocation in the following format:

+0

Contains the number of entries in the list.

+4

Contains the UCB address of the first device.

+8

Contains the UCB address of the second device.

+C

Contains the UCB address of the third device, and so on.

+4

Contains the address of the Step I/O Table (SIOT) for this allocation.

+8

Contains the address of the area where the routine can build a revised device list in the following format:

+0

Contains the number of entries in the list.

+4

Contains the UCB address of the first device.

+8

Contains the UCB address of the second device.

+C

Contains the UCB address of the third device, and so on.

The revised list must be a subset of the original list.

Return Code, Elimination Logic, Register 15:

0

This indicates that TPCF should use the revised device list built by this routine to implement device elimination.

4

This indicates that the routine did not build a revised device list.

8

This indicates that TPCF should use the revised device list built by this routine. TPCF, however, should not remove devices from the revised device list due to job reserve processing.

12 (C)

This indicates that the job is canceled.

ALL registers (except 1 and 15) must be restored upon return.

TPCRECXT Exit

This exit routine customizes the TPCF processing of *allocation* recovery. Through this routine, you can address site-specific processing needs that are not completely served by the CA MIA standard recovery processing. For example, you could use this routine to reply CANCEL (rather than replying WAIT) to the IEF238D message in a testing situation when no devices are available.

As CA MIA processes the TPCRECXT exit routine, certain flags and bytes report the status of devices and list the options available to TPCF when responding to allocation recovery messages. The flags and bytes described below can be used to change the TPCF responses.

You can change these fields in the X35LIST DSECT:

X35FLAG1 (+10)

You can modify this field to eliminate DEVICE NAME or WAIT as options on the MIM2060 message by turning off the appropriate bit setting.

X35FLAG2 (+11)

You can modify this field to change the manner in which TPCF replies to MIM2060 messages by turning off the current bit setting and turning on the appropriate bit. The possible replies are:

- WAIT
- CANCEL
- HOLD
- NOHOLD
- Operator reply
- A local device name

X35FLAG3 (+12)

Eliminates groups of devices from the MIM2042 message if you want TPCF to let the operator reply.

X35UNITR (+14)

Specifies the device name with which TPCF replies if the X35DVNMR bit is on in flag X35FLAG2.

You can change this field in the X35DEV DSECT:

X35STAT4 (+9)

Eliminates specific devices from the MIM2042 message if TPCF is to let the operator reply.

Entry Environment:

Supervisor state, key 0, AMODE 31. You must preserve this entry environment upon entry and restore it upon return.

Register Usage at Entry:

Common Parameter List

R1

This register contains the address of the parameter list, in the following format:

+0

Contains the TPCREEXT parameter list.

+4

Contains the TPCREEXT communication word. The content of this word is set by the exit during its initialization call and is passed unchanged for all subsequent calls.

+8

Contains the full-word of flags used to pass status information to the exit.

+C

Contains the global communication word.

R13

Contains the address of the standard save area.

R14

Contains the return address for CA MIM.

R15

Contains the entry point address for the routine.

All other registers are undefined.

TPCREXT Exit-specific Parameter List

This parameter list is mapped by the X35LIST and X35DEV DSECTS. The field descriptions note which fields can be updated by the exit routine. No other fields can be updated.

+0

X35SIOTA

Contains the address of the Step I/O Table (SIOT) for the current job.

+4

X35CANDC

Contains the number of devices in the X35DLIST DSECT.

+8

X35EXTC

Contains the number of devices in the X35DLIST DSECT that are varied as dedicated on external systems.

+C

X35NAVC

Contains the number of devices in the X35DLIST DSECT that are varied as not available on the local system.

+10

X35FLAG1

Contains flags that describe the options on the MIM2060 message. You can use this field to modify the allowable responses to allocation recovery message MIM2060:

X35DVNMO

DEVICE NAME is an option

X35WAITO

WAIT is an option.

You can use the exit to modify this field. You can eliminate any option. However, you cannot add either option if it is not already provided.

+11**X35FLAG2**

Contains flags used to describe how TPCF is to respond to allocation recovery messages IEF238D and IEF433D:

X35DVNMR

Reply to IEF238D with a device name, which you can find in field X35UNITR (+14). If you activate this bit, then place the device name in the X35UNITR field.

X35WAITR

Reply WAIT to IEF238D.

X35CANCR

Reply CANCEL to IEF238D.

X35OPERR

Let the operator reply to MIM2060.

X35433H

Reply HOLD to IEF433D.

X35433NH

Reply NOHOLD to IEF433D.

X35433O

Let operator reply to IEF433D.

Note: You can use the exit to modify this field. You must use a valid reply as defined in X35FLAG1 (+10).

+12**X35FLAG3**

Contains flags used to describe elimination logic of devices that are listed on MIM2042 messages:

X35XEXT

Eliminate externally dedicated devices from MIM2042.

X35XNAV

Eliminate not available devices from MIM2042.

Note: You can use the exit routine to modify this field.

+13**X35FLAG4**

This is reserved for future releases of CA MIM.

+14

X35UNITR

Contains the three- or four-character, left-justified device name to reply to IEF238D message if the X35DVNMR bit is on in flag X35FLAG2 (+11). The device name must be found in the X35DLIST device list.

Note: You can use the exit routine to modify this field.

+18

X35DLIST

Contains the sequential list of offline devices created from IEF247I device lists. Each entry in this list is defined by the X35DEV DSECT:

+0 X35UNIT—Contains the three-character, left-justified device name that appears in the list.

+4 X35STAT1—Contains the status flags:

- X35DEDH—Dedicated here (to this system).
- X35DEDT—Dedicated there (to another system).
- X35NAV—Not available.

+5 X35STAT2—Contains the status flag:

- X35PREFR—A preference was specified.

+6 X35PREF—Contains the half-word preference value of the device if flag X35PREFR is on in field X35STAT2 (+5). This value is not the same as the value entered on the TPCF VARY device, PREFERENCE command. You can calculate the preference value for the device specified on the VARY command from the X35PREF (+6) field for that device with this equation:

Preference value = X'2710' - X35PREF

+8 X35STAT3— Contains the status flags:

- X35NOMAN—Device not managed by CA MIA.
- X35JRES —Device reserved for the current job name.
- X35NACC— Device in NOT ACCESSIBLE list.
- X35OFFL—Device in OFFLINE list.

+9 X35STAT4—Contains status flags. You can use the following flags to selectively override the elimination logic defined in flag X35FLAG3 (+12):

- X35ELIMY—Eliminate from MIM2042.
- X35ELIMN —Do not eliminate from MIM2042.

Note: You can use the exit routine to modify this field.

Return Code, Elimination Logic, Register 15:

0

Indicates that TPCF should honor any updates to the X35LIST DSECT passed to the exit.

4

Indicates that TPCF should ignore any updates to the X35LIST DSECT passed to the exit.

ALL registers (except 1 and 15) must be restored upon return.

TPCSRMXT Exit

This exit routine executes *preference logic*. By defining preference logic, you can make devices more or less preferred than other devices available for the allocation. Preference logic does not affect selection of devices in SWAP processing.

Entry Environment:

Supervisor state, key 0, AMODE 31. You must preserve this entry environment upon entry and restore it upon return.

Register Usage at Entry:

Common Parameter List

R1

This register contains the address of the parameter list, in the following format:

+0

Contains the TPCSRMXT parameter list.

+4

Contains the TPCSRMXT communication word. The content of this word is set by the exit during its initialization call and is passed unchanged for all subsequent calls.

+8

Contains the full-word of flags used to pass status information to the exit.

+C

Contains the global communication word.

R13

Contains the address of the standard save area.

R14

Contains the return address for CA MIM.

R15

Contains the entry point address for the routine.

All other registers are undefined.

TPCSRMT Exit-specific Parameter List

+0

Contains the address of the list of devices for this allocation in the following format:

+0

Contains the number of entries in the list.

+4

Contains the UCB address of the first device.

+8

Contains the UCB address of the second device.

+C

Contains the UCB address of the third device...and so on.

+4

Contains the address of the Step I/O Table (SIOT) for this allocation.

+8

Contains the address of the area where the routine can build a revised device list in the following format:

+0

Contains the number of entries in the list.

+4

Contains the UCB address of the first device.

+8

Contains the UCB address of the second device.

+C

Contains the UCB address of the third device... and so on.

The revised list must be a subset of the original list.

Return Code, Performance Logic, Register 15:

0

This indicates that TPCF should use the revised list for device preferencing. Any other return code indicates that a revised list was not built.

ALL registers (except 1 and 15) must be restored upon return.

Chapter 6: Utilities and Other Interfaces

This chapter introduces the CA MIA application programming interface and SMF reporting, focusing on Tape Device Usage Report.

This section contains the following topics:

[The Application Program Interface](#) (see page 135)

[Report Generation for CA MIA](#) (see page 148)

The Application Program Interface

The CA MIA application program interface (API) is a module that lets you capture information about managed tape devices and route this information to a TSO session, a system log, or user-defined program.

You can use this interface, for example, to trace particular events (such as pending mounts), to measure device usage so that you can tune your systems for optimal performance, or to assemble usage statistics for logging.

With this interface, you can retrieve the following information:

- z/OS local device name
- CA MIA global device name
- GTAF Autopath host device name
- Device type
- ACL installation status
- ACL active status
- Current volser mounted on a device
- z/OS device status: ONLINE, OFFLINE, ALLOCATED, PENDING-OFFLINE, MOUNT-PENDING
- The name of the job allocating this device
- Identity of system where allocating job is running
- TPCF special device status: OVERGENNED, NOTAVAILABLE, DEDICATED
- Identity of the system to which the device is dedicated
- TPCF job name mask for which the device is reserved

- Identity of systems to which the device is reserved
- TPCF preferencing status
- Eight bytes of user data assigned to the device through the USERDATA command
- Bit mask of known systems
- Bit mask of freed systems

Prepare the Interface

During installation, you unloaded the MIMAPI1 interface module and the sample routines and panels that are used with the MIMAPI1 module.

To use the interface

1. Determine what routines and panels you want to use. All routines and panels are optional.

The following sample source programs are installed into data set CAI.CBTDSAMP:

API1SM01

This is a sample source program that invokes the MIMAPI1 interface module and displays tape data on the two sample panels.

API1SM02

This is a sample source program that retrieves the local names or addresses of devices.

The following sample ISPF panels are installed into data set CAI.CBTDPENU:

API1PNL1

This is an optional retrieval panel that lets you request information about devices.

API1PNL2

This is an optional display panel that shows you information about the devices that you have named.

2. If you are using the API1PNL1 or API1PNL2 panels, then place them in an ISPF library that contains other ISPF panels.
3. Tailor and submit the JCL contained in member API1JCL1 of the CAI.CBTDJCL data set. This JCL assembles and link-edits the two source programs and the MIMAPI1 routine.

The MIMAPI1 Module

You can call the MIMAPI1 module from the API1SM01 display routine, or from any other program. Before you invoke the module from a site-defined program, note these points:

- Call the MIMAPI1 module once for each device by default. Or, you can make multiple device requests using the PARMMULT flag in the MIMAPI1 PARMLA field.
- The MIMAPI1 module does not retain information from one execution to the next.
- Member API1CPY2 of data set CAI.CBTDMAC contains a DSECT that maps the input parameter list passed to the MIMAPI1 module.
- You must define a data area to receive information returned by the MIMAPI1 module, as described in the next section.
- You can run the MIMAPI1 module in protected mode. However, this results in additional overhead for each request.

How You Define an Output Data Area for the MIMAPI1 Module

The MIMAPI1 module returns information to an output data area that you previously defined. Define the data area as follows:

1. Begin the data area on a double-word boundary.
2. Consider the addressing mode of the calling program when you are using the MIMAPI1 interface on a z/OS system. If the program is running in 31-bit mode, then you can define the data area anywhere. If the program is running in 24-bit mode, then define the data area below the 16 MB line.
3. Member API1CPY1 of data set CAI.CBTDMAC contains a DSECT that maps this output data area. The minimum length of the area is the length of the DSECT in the API1CPY1 member.

Entry Environment:

Task mode, key 8, AMODE 31 or AMODE 24, RMODE 24 or RMODE ANY.
The entry environment is preserved and restored upon return.

Register Usage at Entry:

R1

Contains the address of the parameter list, in the following format:

+0 PARML1

Address of the local or global unit name of the device for which you wish to obtain information. If PARMRLSE=PARMR41 (X'040100') or higher, then PARML1 must point to a right-justified 4-character field. For example, to request a match on device 380, PARML1 points to C' 380', where a blank space precedes 380.

+4 PARML2

Contains the address of the receiving area. The INFOAREA DSECT in the API1CPY1 member defines the format for this area.

+8 PARMLA

This contains additional processing instructions. You can specify the following values here:

80 PARMLATM—This runs the MIMAPI1 module in protected mode. Use of this option involves an amount of additional overhead on each request.

40 PARMLARC—When running in the protected mode, this recordsabend information in the SYS1.LOGREC data set.

20 PARMR20—The format of the returned data has changed for CA MIA Release 4.0. The API does not support requests for data format for releases prior to 4.0. You must convert your API calls to use the new parameter lists.

10 PARMMULT—This is a multiple device request that tells MIMAPI1 to obtain the number of devices to return as specified in the PARMNREQ field.

08 PARMRLSF—The release number is in PARMRLSE.

0F PARMLARS—Reserved flags.

+9 PARMRSV1

Reserved.

+A PARMNREQ

This is where you specify how many devices should be returned.

+C PARMNOUT

This field indicates how many devices actually were returned.

+E PARMRLSE

This field shows the release number format (right justified).

40 PARMR40—Release 4.0 format.

040100 PARMR41—Release 4.1 format.

040200 PARMR420—Current format.

+12 PARMRSV2—Reserved.

+14 PARMRSV3—Reserved.

R13

Contains the address of the 18-word save area.

R14

Contains the return address of the invoking program.

R15

Contains the address of the interface module.

All other registers are undefined.

Return Values in the Data Area:

+0 INFOXTRN

Contains the four-byte global name of the device.

+4 INFOUNIT

Contains the local UCB name (right-justified) from the UCB on this z/OS system.

+8 INFOVSER

Contains the volser of the volume mounted on the device.

+E INFOPREF

Contains the preference value assigned to the device in EBCDIC. Valid values are C'0001' through C'0255'. If PREF=NONE, the value is C' '. The default value is C' '.

+12 INFOS1

Contains the TPCF preferencing status assigned to the device. One or more of these values is shown:

80 INFOS1HE

The device is dedicated to the local system.

40 INFOS1TH

The device is dedicated to an external system.

20 INFOS1NA

The device has not-available status on the local system.

10 INFOS1OG

The device has overgenned status on the local system.

+13 INFOS2

Contains the device types for this system:

80 INFT3420

Indicates that the device type is 3420.

40 INFT3480

Indicates that the device type is 3480.

20 INFTAACL

Indicates that the device has Automatic Cartridge Loader (ACL) installed.

08 INFTACLA

Indicates that the device has ACL installed and activated.

10 INFTIDRC

Indicates that the device type is 3480/3490 with IDRC.

04 INFT3490

Indicates that the device type is 3490.

02 INFT3495

Indicates that the device type is 3495.

01 INFT3590

Indicates that the device type is 3590.

+14 INFOMASK

Contains the job name or jobmask, if the device is reserved. If it is not reserved, then the value is CL8' '.

+1C INFOMLEN

Contains the machine length of the job name or jobmask (0-7).

+1D INFOMSYS

This identifies the systems on which the device is reserved. If it is reserved on all systems, then the value is CL8'ALL'. If it is not reserved, then the value is CL8' '.

+25 INFOALOC

If this device is allocated, then this contains the job name the device is allocated to. If the device is not allocated, then this field contains the value CL8' '.

+2D INFOUSRD

This contains 8 bytes of user data information or nulls.

+35 INFOUSRRL

Contains the length of the user data returned in the previous field.

+36 INFOINDX

Contains the index of the local system in the table of systems that is returned in the field at +50. The index is 0-based. The value ranges from 0-31.

+37 INFORSV1

Reserved.

+38 INFOKNOW

Contains the mask of known systems.

+3C INFOFREE

Contains the mask of freed systems.

+40 INFOHNAM

Contains the Autopath host name of the device (right justified), or null if not Autopath-managed.

+44 INFORSV2

Reserved.

System name and flag bytes from each system. These contain the eight-byte system name and two-byte device status for each system. This information is shown once per system. When you are running fewer than 32 systems, nulls are inserted for the entries for undefined systems. The flag byte definitions for the two-byte device status are shown next.

Return Values in the Data Area:

+50 INFOSYS0

CL8' ',XL2'00'

+5A INFOSYS1

CL8' ',XL2'00'

+64 INFOSYS2

CL8' ',XL2'00'

+6E INFOSYS3

CL8' ',XL2'00'

+78 INFOSYS4

CL8' ',XL2'00'

+82 INFOSYS5

CL8' ',XL2'00'

+8C INFOSYS6

CL8' ',XL2'00'

+96 INFOSYS7

CL8' ',XL2'00'

+A0 INFOSYS8

CL8' ',XL2'00'

+AA INFOSYS9

CL8' ',XL2'00'

+B4 INFOSYSA

CL8' ',XL2'00'

+BE INFOSYSB

CL8' ',XL2'00'

+C8 INFOSYSC

CL8' ',XL2'00'

+D2 INFOSYSD

CL8' ',XL2'00'

+DC INFOSYSE

CL8' ',XL2'00'

+E6 INFOSYSF

CL8' ',XL2'00'

+F0 INFOSYTO

CL8' ',XL2'00'

+FA INFOSYT1

CL8' ',XL2'00'

+104 INFOSYT2

CL8' ',XL2'00'

+10E INFOSYT3

CL8' ',XL2'00'

+118 INFOSYT4

CL8' ',XL2'00'

+122 INFOSYT5

CL8' ',XL2'00'

+12C INFOSYT6

CL8' ',XL2'00'

+136 INFOSYT7

CL8' ',XL2'00'

+140 INFOSYT8

CL8' ',XL2'00'

+14A INFOSYT9

CL8' ',XL2'00'

+154 INFOSYTA

CL8' ',XL2'00'

+15E INFOSYTB

CL8' ',XL2'00'

+168 INFOSYTC

CL8' ',XL2'00'

+172 INFOSYTD

CL8' ',XL2'00'

+17C INFOSYTE

CL8' ',XL2'00'

+186 INFOSYTF

CL8' ', XL2'00'

Definition of flag bytes from each system.

First flag byte:

80 INFOF1DH

The device is dedicated to this system.

40 INFOF1DT

The device is dedicated to another system.

20 INFOF1NA

The device has not-available status on this system.

10 INFOF1OV

The device has overgennd status on this system.

08 INFOF1AL

The device is allocated on this system.

04 INFOF1OL

The device is online to this system.

02 INFOF1MP

The device has a mount pending on this system.

01 INFOF1JR

The device is reserved for a job or group of jobs on this system.

Second flag byte:

80 INFOF2PO

The device is pending offline on this system.

7F INFOF2FF

Reserved flags.

Return Code, Register 15:

X'00' CODEGOOD

This indicates that the MIMAPI1 module has placed device information in the specified data area.

X'04' CODENMAT

This indicates that CA MIA is not managing the device.

X'08' CODENCPT

This indicates that the data format you requested is incompatible with the release of CA MIA that you are running. The MIM1API1 module returns as much data to the user as is possible.

X'0C' CODENACT

This indicates that CA MIA is not running or is not synchronized.

X'10' CODENMAN

This indicates that the MIMAPI1 module cannot obtain the GETMAIN storage needed to create the ESTAE exit parameter list.

Note: This return code appears only in test mode.

X'14' CODENEST

This indicates that the MIMAPI1 module cannot establish an ESTAE exit.

Note: This return code appears only in test mode.

X'18' CODEABND

This indicates that the MIMAPI1 module encountered an abend, probably due to receiving a system abend 0C4 when moving data to the specified data area.

Note: This return code appears only in test mode.

Note: ALL registers (except 0, 1, and 15) are restored upon return.

How You Retrieve Device Information

You can use the sample routines and panels available with the API to retrieve information about tape devices and display this information at a TSO session. You can use the interface to format and display the following types of information about a tape device:

- Return code
- z/OS local device name
- CA MIA global device name
- GTAF Autopath “host” device name

- Device type
- ACL installation status
- ACL active status
- Volume serial
- Name of job to which device is allocated
- Identity of system where allocating job is running
- Identity of system to which the device is reserved
- z/OS device status: ONLINE, OFFLINE, ALLOCATED, PENDING-OFFLINE, MOUNT-PENDING
- TPCF “special” device status: OVERGENNED, NOTAVAILABLE, DEDICATED, RESERVED
- TPCF preference value
- TPCF job name mask for which device is reserved

You can display information about a single device, a group of devices, or all devices. You can limit the request to certain devices by telling CA MIA to show you devices that have local names in a range you specify. You also can limit the request to devices that have a status you specify, such as allocated devices, offline devices, and so on.

How You Start the Interface

After preparing the application program interface, you can initiate it using the API1SM01 display program by doing any one of the following:

- Specify PGM(API1SM01) on the ISPF menu selection panel to initiate the API1SM01 program as a menu option.
- Place the TSO command ISPEXEC SELECT PGM(API1SM01) in a one-line CLIST to initiate the API1SM01 program.
- If the API1SM01 program is located in a library that contains other TSO commands, then you can initiate it by entering API1SM01 from Option 6.

When you initiate the API1SM01 program, CA MIA displays the retrieval panel shown in the next section. You can use this retrieval panel to identify the devices for which you want information.

Input fields on panels are identified by the ==> notation. You can use the COMMAND input field on either the retrieval panel or the display panel to issue TSO commands.

The Retrieval Panel

The following is a sample retrieval panel (API1PNL1) for the interface.

```

----- CA MIA Tape Sharing Selection Panel -----
COMMAND ==>

You may limit the display to a subset of the available tape device
addresses shown below.

    Select by range of addresses:   Blank = all
    Beginning address ==>
    Ending address   ==>

    Select by device status:       A = Allocated       F = Offline
    Device status   ==>           V = Available       O = OnLine
                                   M = Mount Pending    R = Reserved

Available tape device addresses:
0580 - 058F 0E70 - 0E7F
0740 - 0747 0E80 - 0E8F
0760 - 076F 0E90 - 0E9F
0780 - 078F 1780 - 178F
0E20 - 0E2F 3100 - 340F
0E50 - 0E5F
0E60 - 0E6F

```

The local names of all the devices on your system are listed at the bottom of the panel, under the AVAILABLE TAPE DEVICE ADDRESSES field.

You can identify the devices you want to obtain information about in these ways:

- To see information about a single device, enter the local name of that device in the BEGINNING ADDRESS field.
- To see information about a range of devices, enter the local name of the first device in the BEGINNING ADDRESS field and the local name of the last device in the ENDING ADDRESS field.
- To see information about all devices, leave both address fields blank.

You also can see information about devices that have a certain status. To restrict the display in this way, specify the code for that status in the DEVICE STATUS field. The valid codes are listed on the panel. For example, by specifying A in this field, you can see information about allocated devices only.

When you are finished, press Enter. CA MIA retrieves the tape device information you requested and displays this information on the display panel.

The Display Panel

The sample display panel (API1PNL2) for the interface is shown in the following sample screen:

```
-----CA MIA Tape Sharing Device Display
COMMAND ==>                                SCROLL ==> CSR
                                Local system name: XX03
RC INAM XNAM HNAM TYPE ACL VOLSER JOB-ALOC SYS-ALOC TPC/MVS PREF RESERVE
00 0E20 0E20 PERM 3480 TAPE01 TAPE1 XX03 A-M
00 0E21 0E21 PERM 3480 TAPE02 TAPE2 Xx13 0-M
00 0E22 0E22 PERM 3480 ON
00 0E23 0E23 PERM 3480 ON
***** BOTTOM OF DATA *****
*****
```

Exit from the Interface

You can exit from the display routine at any time in one of the following ways:

- Issue the END command from the command field of either the retrieval or the display panel.
- Press the PF3 key.

CA MIA returns you to TSO.

Report Generation for CA MIA

CA MIM reports provide an effective tool to evaluate the performance of the systems in your complex. The reports use a sampling of statistical data to yield important information about the operating activities in your complex.

Note: This functionality requires the CA Easytrieve Interface. For more CA Common Services requirements information, see the appendix “CCS for z/OS Component Requirements” in the *Installation Guide*.

How You Plan Reports

CA MIM uses a single SMF record for the record collection process. By using record *subtypes*, you can identify the statistical records to be collected for the different reports available under the CA MIM facilities: GDIF, ECMF, TPCF, and so on.

The major steps involved in the report generation process are:

1. Specify the SMF record number on a MIMINIT RECORDTYPE statement in the initialization member. Once this is set, you should not need to change it.
2. Specify the record collection criteria using the SETOPTION command and begin the record collection process.

Note: You must allow the record collection process to run for a while before dumping the SMF records to data sets.

3. Dump the SMF records to a usable data set through the IBM IFASMFDP utility.
4. Run a report or create a report output file.

Important! CA MIM provides a set of reports for general use. If, however, any one of these reports does not meet your specific requirements, then you can write your own customized reports in the language of your choice, using the SMF records supplied by CA MIM.

Note: For more information, see the MIMSTREC member in the CAI.CBTDMAC data set.

Specify Record Type Number

Before any statistical records can be collected for use in the CA MIM report process, you need to specify an SMF record number. To do this, you place a MIMINIT RECORDTYPE initialization statement in the initialization member.

For example, if you decide to use the default SMF record number of 189, then you would specify the following statement in the initialization member:

```
MIMINIT RECORDTYPE=189
```

Now you are ready to specify the record collection criteria for the report or reports you want to run.

Start Record Collection

The record collection process must be started for each individual CA MIM facility for which you want to run a report.

To start record collection

Issue the following command:

```
SETOPTION facility STATCOLLECT(SUBTYPE=value)STATCYCLE=nn, STATINTERVAL=nn
```

facility

Specifies the CA MIA facility. Valid values are GTAF and TPCF.

SUBTYPE

Specifies which subtypes to use. Valid values are:

TP

Specifies the record subtype for CA MIA facilities for both GTAF and TPCF. The resulting Tape Usage Statistics report shows detailed information about tape device usage for devices managed by GTAF and TPCF.

ALL

Collect records for all record subtypes available under that facility. The record subtypes correspond directly to the reports available for each facility.

STATCYCLE

Specifies the sampling cycle time, in seconds, that CA MIM waits between each statistical data sampling. The default value is 60.

STATINTERVAL

Specifies how often, in minutes, statistical data samples are recorded for use in the GTAF or TPCF report. The default value is 15.

Note: The STATCYCLE and STATINTERVAL values apply to all record collection subtypes for a given CA MIM facility (GTAf, GDIF, and so on).

Important! To activate the record collection process for all record subtypes, you must repeat this procedure for each subtype, specifying the SETOPTION STATCOLLECT command and the facility associated with that record subtype.

Example: Record Collection

This example shows how to begin the record collection process for the Tape Usage Statistic report. This example indicates data sampling every 30 seconds, and a statistical record recording interval of every hour:

```
SETOPTION GTAF STATCOLLECT(SUBTYPE=TP) STATCYCLE=30, STATINTERVAL=60
```

After the statistical record collection process has begun, records are collected and written to the SMF data sets. You need to use your site-specific procedures for dumping the records to physical sequential data sets that can be used by CA MIM to produce reports.

Notes:

- For more information on the SETOPTION commands, see the *Statement and Command Reference Guide*.
- For more information on using the report generator, see the chapter “Utilities” in the *CA MIM Programming Guide*.

Sample Tape Usage Statistics Report (TP)

(12) Interval Start	(13) System Name	(14) Local Alloc	(15) Idle ExDed	(16) Max Mtp Secs 3420	(17) Max Mtp Secs 3480	(18) Max Mtp Secs 3480 ACL	(19) Max Mtp Secs 3490	(20) Max Mtp Secs 3495	(21) Max Mtp Secs 359x	
2010/07/13 14:19	CA MIA									
Tape Usage Statistics Report										
(2) System: SYSA	(5) Jobname: MIMGR	Release: 11.9		(7) Service Level: 0000						(3) PAGE 1
(8) From: 2009/10/25 00:00	(9) Each: RECORD	(6)								
(10) To: 2010/03/12 10:52	(11) Shift: 00:00 24:00									
		Max Avg	Max Avg	Max Mtp Secs Total Mnts	Max Mtp Secs Total Mnts	Max Mtp Secs Total Mnts	Max Mtp Secs Total Mnts	Max Mtp Secs Total Mnts	Max Mtp Secs Total Mnts	
2010/03/07 17:03	VMA	3	0	0	0	0	0	0	0	
		3	0	0	0	0	0	0	0	
	SYSA	3	0	0	0	0	0	0	0	
		2	0	0	0	0	63	0	0	
		0	0	0	0	0	55	0	0	
	SYSB	0	0	0	0	0	2	0	0	
		5	0	0	0	0	100	0	0	
		3	0	0	0	0	58	0	0	
		0	0	0	0	0	10	0	0	

Field	Description
(1)	Date and time the report was created.
(2)	Report title.
(3)	Report page number.
(4)	System name of the CA MIA image as defined by the DEFSYS statement.
(5)	Name of the CA MIA started task.
(6)	The CA MIA release number.
(7)	The CA MIA service level.
(8)	Date and timestamp of the earliest record in the reporting interval.

Field	Description
(9)	Summary level for the data line. In this example, each RECORD is displayed in the detail line. If each HOUR was specified, then the detail line would be a summary of the hour's activity.
(10)	Date and timestamp of the latest record in the recording interval.
(11)	Range of records included in the report.
Interval Start (12)	Start date and timestamp that the report line interval data covers.
System Name (13)	The system name, as defined by a DEFSYS statement, to which the subsequent columns of triplet data belong.
Local Alloc (14)	The maximum, average, and minimum number (reading the column downward) of devices that were allocated on the specified system during the reporting interval. The minimum number is the least number of tape devices allocated during the sampling. For example, if there were no tape devices allocated during a sample, then the minimum would be zero. The maximum number is the highest total number of tape devices allocated during the sampling. This number is not the same as the total number of mounts during the sample.
Idle Ex Ded (15)	The maximum, average, and minimum number (reading the column downward) of devices that were externally dedicated and idle with respect to the specified system during the reporting interval.
3420 (16)	The maximum mount pending time in seconds, the average mount pending time in seconds, and the total number of mounts (reading the column downward) for 3420 type devices on the specified system during the reporting interval.
3480 (17)	The maximum mount pending time in seconds, the average mount pending time in seconds, and the total number of mounts (reading the column downward) for 3480 type devices on the specified system during the reporting interval.
3480 ACL (18)	The maximum mount pending time in seconds, the average mount pending time in seconds, and the total number of mounts (reading the column downward) for 3480 type devices (ACL and non-ACL) with the auto cartridge loader feature on the specified system during the reporting interval.
3490 (19)	The maximum mount pending time in seconds, the average mount pending time in seconds, and the total number of mounts (reading the column downward) for 3490 type devices (ACL and non-ACL) on the specified system during the reporting interval.

Field	Description
3495 (20)	The maximum mount pending time in seconds, the average mount pending time in seconds, and the total number of mounts (reading the column downward) for 3495 type devices on the specified system during the reporting interval.
359x (21)	The maximum mount pending time in seconds, the average mount pending time in seconds, and the total number of mounts (reading the column downward) for 359x type devices on the specified system during the reporting interval.

Appendix A: Integration with CA OPS/MVS

This section contains the following topics:

[Overview](#) (see page 155)

[Enable CA OPS/MVS Event Notification](#) (see page 156)

[API Rules](#) (see page 156)

[CA MIA VARY Delay Detected Event](#) (see page 156)

[CA MIA VARY REQUEUE Notification Event](#) (see page 158)

[CA MIA VARY Delay Abort Complete Event](#) (see page 160)

Overview

CA MIM provides seamless integration with CA OPS/MVS in several areas, including VARY delay events.

You do not need to do anything for CA MIM to enable the product interface to CA OPS/MVS and you do not need to issue any CA MIM initialization statement or commands to activate the interface. If CA MIM and CA OPS/MVS are active in the same z/OS image, CA MIM automatically communicates CA MIM automation events to CA OPS/MVS.

Before the CA MIM-to-CA OPS/MVS interface existed, CA OPS/MVS could automate CA MIM events through console messages, and CA OPS/MVS can still use CA MIM console message traffic for automation events. Console messages are processed by CA OPS/MVS)MSG rules. Depending upon the particular automation event, CA OPS/MVS rules may need to correlate console messages and, in some instances, issue commands and interrogate command responses to collect information about a given event.

On the other hand, the CA MIM-to-CA OPS/MVS interface is an integrated two-way communication mechanism. CA MIM passes automation event information directly to CA OPS/MVS, including all pertinent data related to the automation event in the form of OPS REXX variables. These events can be automated using a CA OPS/MVS)API rule. This interface is more efficient and reduces the complexity of CA OPS/MVS automation rules as the event data is readily available in REXX variables. This internal interface also eliminates the dependence on the format and content of console message text.

Also, CA MIM can take action on an automation event as directed by a CA OPS/MVS)API automation rule. CA MIM and CA OPS/MVS seamlessly work together to automate the management of shared-system resources.

Enable CA OPS/MVS Event Notification

To enable the interface from CA OPS/MVS, set the CA OPS/MVS parameter APIACTIVE to ON. This allows CA OPS/MVS to acknowledge and process the events generated by CA MIM.

API Rules

For each automation event presented by CA MIM, the interface triggers an)API rule, and supplies data to the rule in the form of environmental variables. Each type of event has an associated set of variable names. Some variable names are common, regardless of the automation event while other variable names are specific to the automation event.

Note: For more API information, see the *CA OPS/MVS AOF Rules User Guide*.

Important! Unless otherwise noted, all of the variables are a data type of character and are read-only variables; therefore, a CA OPS/MVS rule program cannot store data into these variables. An attempt to alter the content of read-only variables will result in a CA OPS/MVS rule program failure.

CA MIA VARY Delay Detected Event

CA MIA monitors the progress of managed device VARY activity and provides the means to generate notifications, on a cyclical basis, should the VARY activity for a given device exceed an installation specified threshold.

Note: For more information about establishing the threshold and interval for VARY delay processing, see the description of the VARYDELAY keyword of the SETOPTION GTAF command in the *Statement and Command Reference Guide*.

When CA MIA detects that the attempt to VARY a managed device has exceed the specified threshold, it issues a highlighted MIM2211 console message on a cyclical basis until the VARY activity completes. CA MIA also directly notifies CA OPS/MVS of the VARY delay event and permits CA OPS/MVS to evaluate the VARY delay event in an)API rule. The)API rule can instruct CA MIA to continue waiting for the VARY to complete in which case CA MIA will continue waiting and the cyclical process will repeat again if the VARY device activity does not complete within the next VARYDELAY interval. Alternatively, the)API rule can instruct CA MIA to terminate the VARY device activity, in which case, CA MIA abends the target task processing the VARY device that is being delayed.

The CA MIA VARY device delay event is presented to CA OPS/MVS and can be processed by the following rule:

)API MIM2211

The available OPS/REXX variables and value for a CA MIA VARY device delay event are:

API.APPLICATION

MiMgr

API.VERSION

Current release

API.LEVEL

Service level

API.EVENTID

MIM2211

API.MSGID

MIM2211

API.TASKNAME

1 to 8 character CA MIM internal task name. This task name can be correlated to the CA MIM DISPLAY TASK command response.

API.TCB

8 character z/OS TCB address of the MIMTPVDV task that is experiencing the VARY device delay.

API.ACTION

This variable is a read-write character variable. The variable is initialized to the character string 'CONTINUE' by CA MIA before the VARY delay event is presented to CA OPS/MVS. The)API rule can leave the ACTION variable unchanged in which case CA MIA will continue to wait for the VARY device to complete. Otherwise, the)API rule can change the ACTION variable to the text string 'ABORT' which will instruct CA MIA to terminate the VARY device processing that is being delayed.

API.TEXT

MIM2211W VARY *devn* state delayed by *nnn* seconds. The format of the API.TEXT variable is the same as the MIM2211W console message. The MIM2211W message ID within the text is subject to message ID prefixing as controlled by the CA MIM command, SETOPTION MIM MSGPREFIX keyword.

devn

Specifies the target device number

state

Can be either 'ONLINE' or 'OFFLINE'

nnn

Specifies the number of seconds that have elapsed since the VARY device was initiated.

Member APIMIMGR of HLQ.OPS.SAMPLE.RULES demonstrates how to use the MIM2211 API.

CA MIA VARY REQUEUE Notification Event

CA MIA monitors managed device VARY activity and generates notification when a VARY command for a given device is requeued by z/OS. z/OS requeues a VARY command when ENQ resources necessary for z/OS to serialize a VARY command with other system functions cannot be obtained within 5 seconds. VARY REQUEUE Notification occurs when the number of times that a VARY command has been requeued exceeds an installation specified threshold.

Note: For more information about establishing the threshold for VARY REQUEUE Notification, see the description of the *VARYRQNTFY keyword of the SETOPTION GTAF command in the CA MIM Statement and Command Reference Guide.*

When CA MIA detects that the number of times that a VARY command for a managed device has been requeued exceeds the VARYRQNTFY threshold, it issues a MIM2225 console message. This message is issued on an iterative basis until the VARY activity completes. CA MIA also directly notifies CA OPS/MVS of the VARY REQUEUE event and permits CA OPS/MVS to evaluate the VARY REQUEUE event in an 'API rule'. The 'API rule' can instruct CA MIA to continue to retry the VARY command. In this case CA MIA will reset the requeue counter for the VARY command. The VARY REQUEUE Notification process will repeat again if the SYSIEFSD ENQ contention issue remains in effect. Alternatively, the 'API rule' can instruct CA MIA to terminate the VARY device activity. In this case, CA MIA will not retry the VARY command and will remove it from the CA MIA VARY command processing queue.

The CA MIA VARY Requeue Notification event is presented to CA OPS/MVS and can be processed by the following rule:

```
)API MIM2225
```

The available OPS/REXX variables for a CA MIA VARY Requeue Notification event are:

OPS/REXX

Variable Value

API.APPLICATION

MiMgr

API.VERSION

Current release

API.LEVEL

Service level

API.EVENTID

MIM2225

API.MSGID

MIM2225

API.TASKNAME

1 to 8 character CA MIM internal task name. This task name can be correlated to the CA MIM DISPLAY TASK command response.

API.TCB

8 character z/OS TCB address of the MIMTPVDV task that has been requeued.

API.ACTION

This variable is a read-write character variable. The variable is initialized to the character string 'CONTINUE' by CA MIA before the VARY delay event is presented to CA OPS/MVS. The)API rule can leave the ACTION variable unchanged. In this case, CA MIA will continue to retry the VARY command until it eventually completes when ENQ contention has been relieved. Alternatively, the)API rule can change the ACTION variable to the text string 'ABORT'. The 'ABORT' ACTION variable instructs CA MIA NOT to retry a VARY command that has been requeued due to ENQ contention.

API.TEXT

MIM2225W VARY 078F OFFLINE REQUEUED - SYSIEFSD Q4 ENQ CONTENTION. The format of the API.TEXT variable is the same as the MIM2225W console message. The MIM2225W message ID within the text is subject to message ID prefixing as controlled by the CA MIM command, SETOPTION MIM MSGPREFIX keyword.

devn

Specifies the target device number

state

Can be either ONLINE or OFFLINE.

Member APIMIMNW of HLQ.OPS.SAMPLE.RULES demonstrates how to use the MIM2225 API.

CA MIA VARY Delay Abort Complete Event

CA MIA will generate a VARY ABORT COMPLETE event when control has returned from a CA OPS/MVS)API event that set the ACTION variable to the text string 'ABORT'.

A VARY device abort complete event is generated to notify CA OPS/MVS that the ABORT request has completed for two possible events:

MIM2211

When CA MIA has completed the processing necessary to interrupt and terminate delayed VARY device activity.

MIM2225

When CA MIA has removed a VARY command from its processing queues due to ENQ contention related requeueing.

The CA MIA VARY device delay event is presented to CA OPS/MVS and can be processed by the following rule:

)API MIM2214

The available OPS/REXX variables for a CA MIA VARY device abort complete event are:

OPSREXX Variable

Value

API.APPLICATION

MiMgr

API.VERSION

Current release

API.LEVEL

Service level

API.EVENTID

MIM2214

API.MSGID

MIM2214

API.TASKNAME

One through eight character CA MIM internal task name. This task name can be correlated to the CA MIM DISPLAY TASK command response.

API.TCB

Eight character z/OS TCB address of the MIMTPVDV task that is experiencing the VARY device delay.

API.ACTION

This read only variable contains the text string ABORT that the)API MIM2211 or MIM2225 rule returned.

API.TEXT

MIM2214I VARY *devn state* terminated by automation *action* request

The format of the API.TEXT variable is the same as the MIM2214I console message. The MIM2214I message ID within the text is subject to CA MIM message ID prefixing. The CA MIM SETOPTION MIM MSGPREFIX command keyword controls CA MIM Message ID prefix.

devn

Specifies the target device number.

state

Can be either ONLINE or OFFLINE.

action

Specifies the content of the API.ACTION variable.

Note: The CA OPS/MVS API Rule name is APIMIMGR.

Index

A

- ACL status messages, controlling • 18
- ACTIVATE and HCD • 41
- activating and deactivating • 57
- activating and deactivating the ACL • 57
- activating event tracing • 107
- ACTIVE operand • 57
- ACTIVE operand, for the VARY command • 57
- allocation problems • 92
- allocation recovery • 25
- allocation recovery messages, setting automatic response to • 18
- API1JCL1 member • 136
- API1PNL1 retrieval panel • 136
- API1PNL2 display panel • 136, 148
- application program interface (API) • 14, 135
- ASSIGN processing • 58
- ASSIGN type, defining • 17
- assignable tape devices • 57, 58
- assigning global names to devices • 103
- assigning to devices • 30
- automatic cartridge loader (ACL) • 12, 57
- automating allocation responses • 13
- Autopath • 87
 - define participation • 17
 - eligibility of device • 33
 - processing,controlling • 18
- AUTOREPLY operand • 72, 100
- AUTOREPLY operand for the SETOPTION command • 72, 100
- AVAILABLE operand • 55, 56, 57, 76
- AVAILABLE operand, for the VARY command • 56, 57, 76

B

- bringing ONLINE automatically • 45

C

- capturing device information • 14
- changing • 39
- changing device status • 13
- changing managed • 39
- COMMANDS parameter • 39
- compatlevel=, SWAP processing • 78

- configure OFFLINE • 45
- configuring for occasional or part-time • 87
- contacting technical support • 3
- controlling allocation recovery • 72, 77
- controlling recovery processing with SETOPTION
 AUTOREPLY • 72
- creating • 63
- customer support, contacting • 3

D

- DEDICATED operand • 60
- DEDICATED operand , for the VARY command • 60
- dedicating a device to a system • 60
- dedicating to a system • 60
- defining • 17
- defining and attaching drives for VM guests • 82
- defining multiple addresses for • 86
- defining to Unicenter CA-MIA • 38
- determining where to send message MIM2046 • 77
- DEVCLASS parameter • 39
- DEVCLASS parameter, identifying a class of devices • 31
- device
 - allocation • 22
- DEVLIST parameter • 39
- diagnosing problems • 94
- diagnostic information, obtaining • 18
- displays, setting format values for • 18
- dual allocations, responding to • 103
- dynamic device reconfiguration(DDR) • 77
- dynamic I/O reconfiguration • 39

E

- effect on device groups • 63
- eligible device list (EDL) • 77
- eligible device list (EDL), refer to TPCF Device
 Processing Appendix • 77
- ENDIF statements • 36
- esoteric device groups • 63
- event processing order, setting up • 50
- exit for customizing • 125
- exit routine for • 122, 131
- exit routines • 119

F

FACILITIES operand • 103
FACILITIES operand for the DISPLAY command • 103
for a group of jobs • 61
for a single job • 61
for the DISPLAY command • 100
for the GTAINIT statement • 58
for the SETOPTION command • 57
for the VARY command • 61
for troubleshooting • 103
FORCE operand, for VARY NOTAVAILABLE command • 55
FORCE, setting default for job reserve processing • 18
forcing a device offline while allocated to another system • 55

G

generating reports • 148
generic device groups • 63
global names of device • 33
Global Tape Allocation Facility (GTAF) • 12
GLOBALUNITS operand • 103
GLOBALUNITS operand for the DISPLAY command • 103
groups, specifying preference order • 45
GTAF • 12
GTAINIT statement, ASSIGN parameter • 58
GTALOCAL Enqueue Processing • 92

H

Hardware Configuration Definition (HCD) • 39
Hardware Configuration Dialog (HCD) • 74
HCD ACTIVATE • 41, 43
how it limits TPCF processing • 69
how z/OS orders device groups in • 70

I

IBM IFASMFDP utility • 149
identifying a class of devices • 31
IEF238D messages • 56, 57, 99, 100
IEF433D messages • 56, 57, 76
IEF433D operand • 76
IEF433D operand for the SETOPTION AUTOREPLY command • 76
IEFSSNxx member • 50
IFSYS statements • 36

influencing device selection • 13, 59, 67
INTERVAL and CYCLE parameters, setting • 80
IODEVICE macro • 45

J

JOB operand • 61, 100
job reserve processing • 63
JOBRESERVE operand • 102
JOBRESERVE operand for the DISPLAY command • 102

L

local name of devices • 30

M

making a device unavailable for allocation • 53
managing devices with Unicenter CA-MIA • 30
managing with Unicenter CA-MIA • 30
members, modifying facility-specific • 16
MIM2032 operand • 57
MIM2046 operand • 77
MIM2046 operand for the SETOPTION AUTOREPLY command • 77
MIM2060 message • 62, 102
MIMINIXT • 120
MIMUNITS member • 103
multi-application environments • 50
multiple channel interfaces, defining paths to • 83
multiple control unit • 83

N

names • 33
NOJOB operand • 61
NOJOB operand, for the VARY command • 61
non-specific, non-temporary • 71
NOTACTIVE operand • 57
NOTACTIVE operand, for the VARY command • 57
NOTAVAILABLE FORCE operand • 55
NOTAVAILABLE operand • 53
NOTAVAILABLE operand, for the VARY command • 53
NOTDEDICATED operand • 60
NOTDEDICATED operand, for the VARY command • 60

O

offline device list • 72

on SETOPTION TPCF command • 65
on the SETOPTION TPCF command • 54
OVERGENNED devices, marked ineligible in EDL • 65

P

passwords • 58
planning • 20
PREFERENCE operand • 59
PREFERENCE operand, VARY command • 59
preference value assignments • 59

R

record sampling cycle time • 150
record subtypes • 150
records, defining settings for statistical • 18
RECORDTYPE parameter • 149
RECORDTYPE parameter for the MIMINIT statement
• 149
reports • 151
reserves • 61
responding when a job is waiting for a device • 76
RESYNCH command • 39, 103
RESYNCH operand, for the SETOPTION command •
39
RESYNCH processing, defining • 18
resynchronization command • 39
rules, coding in the MIMUNITS member • 35

S

selection of • 13
serialization • 12, 46, 48
serialization of • 44
serialization of HCD ACTIVATE • 44
SETOPTION command • 149
SETOPTION TPCF command. OVGINELIG parameter •
54
setting allocation recovery • 17
setting defaults • 18
setting FORCE default for • 18
SETTRACE parameter, for the SETOPTION MIM
command • 108
sharing • 36
single channel interface • 83
single control unit • 83
SMF record number • 149
specific, temporary • 71
specifying an SMF record number • 149
specifying devices in • 33

support, contacting • 3
SWAP compatlevel= • 78
SWAP processing • 77, 78

T

tape devices, sharing assignable • 13
Tape Preferecing and Control Facility (TPCF) TPCF •
13
tape swapping • 61
technical support, contacting • 3
TPCEDLXT • 67, 77
TPCEDLXT exit routine • 62, 63, 67, 77, 119, 122
TPCRECXT • 67, 77
TPCRECXT exit routine • 67, 77, 119, 125
TPCSRMXT • 67, 77
TPCSRMXT exit routine • 67, 77, 119, 131
tracing events • 107

U

unavailable, making a device • 53
Unicenter CA-MIA and • 27
Unit Control Block (UCB) • 30
UNIT parameter • 63
UNITS MIM file • 87
using VARY ONLINE in • 45

V

VARY command • 52, 100
VARY ONLINE, in MIMSYNCH member • 45
varying ONLINE and OFFLINE • 52

W

wait-eligible device list • 72
waking jobs waiting for devices • 56, 57

Z

z/OS
ACL processing • 103
z/OS SYSGEN statement • 63