

# **MAINVIEW<sup>®</sup> AutoOPERATOR Solutions Guide**

**Version 6.3.01**

**November 21, 2003**



Copyright 2003 BMC Software, Inc. All rights reserved.

BMC Software, the BMC Software logos, and all other BMC Software product or service names are registered trademarks or trademarks of BMC Software, Inc. IBM and DB2 are registered trademarks of International Business Machines Corp. All other trademarks belong to their respective companies.

BMC Software considers information included in this documentation to be proprietary and confidential. Your use of this information is subject to the terms and conditions of the applicable End User License Agreement for the product and the proprietary and restricted rights notices included in this documentation.

## Restricted Rights Legend

U.S. Government Restricted Rights to Computer Software. UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure of any data and computer software by the U.S. Government is subject to restrictions, as applicable, set forth in FAR Section 52.227-14, DFARS 252.227-7013, DFARS 252.227-7014, DFARS 252.227-7015, and DFARS 252.227-7025, as amended from time to time. Contractor/Manufacturer is BMC Software, Inc., 2101 CityWest Blvd., Houston, TX 77042-2827, USA. Any contract notices should be sent to this address.

---

## Contacting BMC Software

You can access the BMC Software Web site at <http://www.bmc.com>. From this Web site, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

### United States and Canada

**Address** BMC Software, Inc.  
2101 CityWest Blvd.  
Houston TX 77042-2827

**Telephone** 713 918 8800 or  
800 841 2031

**Fax** 713 918 8000

### Outside United States and Canada

**Telephone** (01) 713 918 8800

**Fax** (01) 713 918 8000

---

---

## Customer Support

You can obtain technical support by using the Support page on the BMC Software Web site or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, please see “Before Contacting BMC Software.”

### Support Web Site

You can obtain technical support from BMC Software 24 hours a day, 7 days a week at <http://www.bmc.com/support.html>. From this Web site, you can

- read overviews about support services and programs that BMC Software offers
- find the most current information about BMC Software products
- search a database for problems similar to yours and possible solutions
- order or download product documentation
- report a problem or ask a question
- subscribe to receive e-mail notices when new product versions are released
- find worldwide BMC Software support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

### Support by Telephone or E-mail

In the United States and Canada, if you need technical support and do not have access to the Web, call 800 537 1813. Outside the United States and Canada, please contact your local support center for assistance. To find telephone and e-mail contact information for the BMC Software support center that services your location, refer to the Contact Customer Support section of the Support page on the BMC Software Web site at [www.bmc.com/support.html](http://www.bmc.com/support.html).

### Before Contacting BMC Software

Before you contact BMC Software, have the following information available so that Customer Support can begin working on your problem immediately:

- product information
  - product name
  - product version (release number)
  - license number and password (trial or permanent)
- operating system and environment information
  - machine type
  - operating system type, version, and service pack or other maintenance level such as PUT or PTF
  - system hardware configuration
  - serial numbers
  - related software (database, application, and communication) including type, version, and service pack or maintenance level
- sequence of events leading to the problem
- commands and options that you used
- messages received (and the time and date that you received them)
  - product error messages
  - messages from the operating system, such as `file system full`
  - messages from related software



---

---

# Contents

<b>About This Book</b> .....	<b>xv</b>
<b>Chapter 1</b>	<b>Introduction to MAINVIEW AutoOPERATOR Solutions</b>
	EXEC Distribution ..... 1-1
	Rules Distribution ..... 1-2
	Documentation Boxes ..... 1-2
	Documentation Box Fields ..... 1-4
	Notes about Solutions ..... 1-5
<b>Chapter 2</b>	<b>Managing Critical Applications with Continuous State Manager</b>
	What Continuous State Manager Is ..... 2-2
	What the Benefits of CSM Are ..... 2-3
	What CSM Requires ..... 2-4
	Defining CSM Terms and Concepts ..... 2-4
	CSM Terms and Concepts ..... 2-4
	What an Object Is ..... 2-5
	What Object Relationships Are ..... 2-6
	What an Object State Is ..... 2-9
	What a Schedule Is ..... 2-11
	What Else CSM Must Know and Why ..... 2-13
	How CSM Starts, Stops, and Cancels Objects ..... 2-14
	How CSM Recognizes when an Object Starts or Stops ..... 2-14
	How CSM Connects and Disconnects Objects ..... 2-15
	How CSM Works ..... 2-16
	When CSM Starts ..... 2-16
	Routine Scheduling ..... 2-17
	Recovering Failed Objects ..... 2-17
	Interfering with CSM Processing ..... 2-19
	Managing Objects across BBI-SS PASs ..... 2-20
	Examples of CSM Processing ..... 2-21
	How CSM Automates the IPL Process ..... 2-22
	How CSM Automates the Shutdown Process ..... 2-25
	How CSM Manages Routine Object Scheduling ..... 2-29
	How CSM Handles Global Calendar Overrides ..... 2-30

---

How to Effectively Plan for CSM Implementation . . . . .	2-32
Determining Object Relationships . . . . .	2-34
Planning Schedules . . . . .	2-34
Listing Start, Stop, Cancel and (Dis)Connect Commands . . . . .	2-35
Choosing Start and Stop Text-IDs . . . . .	2-36
Choosing Text-IDs for Abnormal Termination Events . . . . .	2-36
Before You Begin . . . . .	2-37
Setting Up the Shared Repository . . . . .	2-37
Cross-System Object Management: Using Group . . . . .	2-37
Serializing the CSM Shared Repository . . . . .	2-38
Securing Access to CSM . . . . .	2-38
Backing Up the CSM Repository . . . . .	2-39
Using CSM Panels . . . . .	2-40
Accessing CSM for the First Time . . . . .	2-42
How to Qualify (Mask) Panels in CSM . . . . .	2-43
What the CSM Global Overview Panel Is . . . . .	2-44
Viewing an Object's Documentation . . . . .	2-45
Viewing Objects Whose States Do Not Match . . . . .	2-46
Filtering Groups That Are Displayed on Global Overview . . . . .	2-47
Defining CSM Objects . . . . .	2-51
Creating a Group . . . . .	2-51
Entering the Object's Information . . . . .	2-55
Entering Command Specifications for an Object . . . . .	2-57
Completing the Additional Fields . . . . .	2-60
Specifying Conditional Command Specifications for an Object . . . . .	2-61
Specifying Start and Stop Events: Event Specifications 1 . . . . .	2-63
Specifying Abnormal Termination Events: Event Specifications 2 . . . . .	2-64
Specifying User Notification . . . . .	2-66
Associating an Object with a Group . . . . .	2-68
Moving an Object between Groups . . . . .	2-70
Defining the Object's Parents . . . . .	2-71
Defining the Object's Servers . . . . .	2-72
Defining a Transient Object . . . . .	2-74
Defining a Grouping Object . . . . .	2-75
Comparing CSM Object Types . . . . .	2-75
Using the Repeat Line Command to Create Objects . . . . .	2-76
Modifying Object Definitions . . . . .	2-79
Adding Parents . . . . .	2-79
Deleting Parents . . . . .	2-83
Adding Servers . . . . .	2-86
Deleting Servers . . . . .	2-90
Modifying an Object's Routine Schedule . . . . .	2-93
Determining the Schedule of the Object's Parent . . . . .	2-93
Modifying the Object's Schedule . . . . .	2-96
Modifying an Object's Schedule with CSM Schedule Tags . . . . .	2-100
Dealing with Scheduling Conflicts at Modification . . . . .	2-104
Overriding the Calendar for Special Situations . . . . .	2-106

Enabling Cross-System Object Management . . . . .	2-108
Performing Routine Object Tasks . . . . .	2-110
Starting an Object . . . . .	2-110
Stopping an Object . . . . .	2-114
Bouncing an Object . . . . .	2-116
Resetting an Object . . . . .	2-119
Canceling an Object . . . . .	2-122
Moving an Object: Using the Moveable Attribute . . . . .	2-125
Scheduling an EXEC Associated with a CSM ALERT . . . . .	2-128
Displaying and Modifying Object States . . . . .	2-130
What the Possible Desired States Are . . . . .	2-130
What the Possible Actual States Are . . . . .	2-133
Monitoring CSM Object States Manually . . . . .	2-133
Using the Command Line Interface . . . . .	2-139
Describing CSM Panels . . . . .	2-142
CSM Global Overview Panel . . . . .	2-142
Defining and Modifying Object Panels . . . . .	2-151
Object Detail Control Panel . . . . .	2-152
Command Specifications Panel . . . . .	2-155
Conditional Command Specifications Panel . . . . .	2-160
Event Specifications 1 Panel . . . . .	2-163
Event Specifications 2 Panel . . . . .	2-166
User Notification Panel . . . . .	2-168
Object Groups Panel . . . . .	2-170
Object Requisites Panel . . . . .	2-172
Client/Server Relationships Panel . . . . .	2-174
Object Scheduling Panel . . . . .	2-177
Global Calendar Override Panel . . . . .	2-180
Object Groups Panel . . . . .	2-184
Group Detail Control . . . . .	2-188

## Chapter 3

### JES2 Solution

JES2 Monitoring Solution . . . . .	3-1
Variables . . . . .	3-2
Invocation . . . . .	3-2
Processing Flow . . . . .	3-3
EXECs . . . . .	3-3
Rules . . . . .	3-4

## Chapter 4

### Storage Subsystems Solutions

Shared DASD Control Solution . . . . .	4-1
Variables . . . . .	4-2
Invocation . . . . .	4-2
Processing Flow . . . . .	4-2
EXECs . . . . .	4-2
Rules . . . . .	4-3

LMS Solution .....	4-3
Variables .....	4-3
Invocation .....	4-3
Processing Flow .....	4-3
EXECs .....	4-4
Rules .....	4-4
Storage Reply Solution .....	4-4
Variables .....	4-4
Invocation .....	4-4
Processing Flow .....	4-4
EXECs .....	4-5
Rules .....	4-5

## Chapter 5

### Monitoring Solutions

Dump Data Sets Monitoring Solution .....	5-1
Variables .....	5-1
Invocation .....	5-2
Processing Flow .....	5-2
EXECs .....	5-3
Rules .....	5-3
WTO Buffers Monitoring Solution .....	5-3
Variables .....	5-3
Invocation .....	5-4
Processing Flow .....	5-4
EXECs .....	5-4
Rules .....	5-5
SMF Data Set Monitoring Solution .....	5-5
Variables .....	5-5
Invocation .....	5-6
Processing Flow .....	5-6
EXECs .....	5-7
Rules .....	5-7
LOGREC Data Set Monitoring Solution .....	5-8
Variables .....	5-8
Invocation .....	5-8
Processing Flow .....	5-9
EXECs .....	5-9
Rules .....	5-9
RMF Monitoring Solution .....	5-10
Variables .....	5-10
Invocation .....	5-10
Processing Flow .....	5-10
EXECs .....	5-10
Rules .....	5-10

<b>Chapter 6</b>	<b>Access Methods Solution</b>	
	VSAM Failures Solution . . . . .	6-1
	Variables . . . . .	6-1
	Invocation . . . . .	6-2
	Processing Flow . . . . .	6-2
	EXECs . . . . .	6-2
	Rules . . . . .	6-2
<b>Chapter 7</b>	<b>Job Scheduling Solution</b>	
	Introduction . . . . .	7-1
	Variables . . . . .	7-1
	Invocation . . . . .	7-1
	Processing Flow . . . . .	7-2
	EXECs . . . . .	7-2
	Rules . . . . .	7-2
<b>Chapter 8</b>	<b>CONTROLM Solution</b>	
	Variables . . . . .	8-1
	Invocation . . . . .	8-1
	Processing Flow . . . . .	8-2
	EXECs . . . . .	8-2
	Rules . . . . .	8-2
<b>Chapter 9</b>	<b>MVS Performance Management Solutions</b>	
	Load Balancing Solution . . . . .	9-1
	Variables . . . . .	9-2
	Invocation . . . . .	9-2
	Processing Flow . . . . .	9-2
	EXECs . . . . .	9-3
	Rules . . . . .	9-3
	DASD Reserve Analysis Solution . . . . .	9-4
	Variables . . . . .	9-4
	Invocation . . . . .	9-4
	Processing Flow . . . . .	9-4
	EXECs . . . . .	9-5
	Rules . . . . .	9-5
	Exception Monitoring Solution . . . . .	9-5
	Variables . . . . .	9-5
	Invocation . . . . .	9-6
	Processing Flow . . . . .	9-7
	EXECs . . . . .	9-7
	Rules . . . . .	9-8
<b>Chapter 10</b>	<b>Error Recovery Solutions</b>	
	Storage Violation Solution . . . . .	10-2
	Variables . . . . .	10-2
	Invocation . . . . .	10-2
	Processing Flow . . . . .	10-3

EXECs	10-3
Rules	10-3
CICS Abnormal Termination Solution	10-4
Variables	10-4
Invocation	10-4
Processing Flow	10-4
EXECs	10-5
Rules	10-5
VSAM Subtask Abnormal Termination Solution	10-5
Invocation	10-6
Processing Flow	10-6
EXECs	10-6
Rules	10-6
CICS Temporary Storage Suspensions Solution (CICSTSS)	10-7
Variables	10-7
Invocation	10-7
Processing Flow	10-7
EXECs	10-8
Rules	10-8
Temporary Storage (TS) Data Set Extension Failure Solution	10-9
Variables	10-9
Invocation	10-9
Processing Flow	10-9
EXECs	10-10
Rules	10-10
Terminal Errors Solution	10-10
Variables	10-10
Invocation	10-11
Processing Flow	10-11
EXECs	10-11
Rules	10-11

## Chapter 11

### CICS Performance Management Solutions

File Degradation Analysis Solution	11-1
Variables	11-2
Invocation	11-2
Processing Flow	11-2
EXECs	11-3
Rules	11-4
File Allocation - Deallocation Solution	11-4
Variables and JCL	11-4
Invocation	11-5
Processing Flow	11-5
EXECs	11-6
Rules	11-6

---

	VSAM Control Area Split Monitor Solution . . . . .	11-6
	Variables and JCL . . . . .	11-7
	Invocation . . . . .	11-7
	Processing Flow . . . . .	11-7
	EXECs . . . . .	11-8
	Rules . . . . .	11-8
	Transaction Response Time Monitor Solution . . . . .	11-8
	Variables and JCL . . . . .	11-8
	Invocation . . . . .	11-9
	Processing Flow . . . . .	11-9
	EXECs . . . . .	11-9
	Rules . . . . .	11-9
<b>Chapter 12</b>	<b>VTAM/NCP Solution</b>	
	TCAS Reply Solution . . . . .	12-1
	Variables . . . . .	12-1
	Invocation . . . . .	12-1
	Processing Flow . . . . .	12-2
	EXECs . . . . .	12-2
	Rules . . . . .	12-2
<b>Chapter 13</b>	<b>Initial Customization Steps</b>	
	Customization Steps for DB2 Solutions . . . . .	13-1
<b>Chapter 14</b>	<b>DB2 Global Operations Solutions</b>	
	DB2 Major Messages (DB2ALRT Solution) . . . . .	14-1
	Variables . . . . .	14-2
	Invocation . . . . .	14-2
	Processing Flow . . . . .	14-5
	EXECs . . . . .	14-5
	Rules . . . . .	14-5
	DB2 Environment Set-Up (DB2INIT Solution) . . . . .	14-5
	Variables . . . . .	14-6
	Invocation . . . . .	14-7
	Processing Flow . . . . .	14-7
	EXECs . . . . .	14-7
	Rules . . . . .	14-7
	DMR Major Messages (DMRALRT Solution) . . . . .	14-8
	Variables . . . . .	14-8
	Invocation . . . . .	14-8
	Processing Flow . . . . .	14-9
	EXECs . . . . .	14-9
	Rules . . . . .	14-10
	DMR Utilities (DMRUTIL Solution) . . . . .	14-10
	Variables . . . . .	14-10
	Invocation . . . . .	14-10
	EXECs . . . . .	14-11
	Rules . . . . .	14-12

<b>Chapter 15</b>	<b>DB2 Resource Contention Analysis Solutions</b>	
	Deadlock/Timeout Analysis (DB2DLOK Solution) . . . . .	15-1
	Variables . . . . .	15-2
	Invocation . . . . .	15-3
	Processing Flow . . . . .	15-3
	EXECs . . . . .	15-6
	Rules . . . . .	15-7
	DB2 Table Space Filling Up (DB2TFUL Solution) . . . . .	15-7
	Variables . . . . .	15-8
	Invocation . . . . .	15-9
	Processing Flow . . . . .	15-9
	EXECs . . . . .	15-9
	Rules . . . . .	15-9
	DB2 Thread Control (DB2THRD Solution) . . . . .	15-10
	Variables . . . . .	15-10
	Invocation . . . . .	15-10
	Processing Flow . . . . .	15-11
	Rules . . . . .	15-11
<b>Chapter 16</b>	<b>DB2 Performance Management Solutions</b>	
	DB2 Runaway Query Control (DB2RNWY Solution) . . . . .	16-1
	Variables . . . . .	16-2
	Invocation . . . . .	16-2
	Processing Flow . . . . .	16-2
	EXECs . . . . .	16-3
	Rules . . . . .	16-3
	DB2 Response Time (DB2RESP Solution) . . . . .	16-4
	Variables . . . . .	16-4
	Invocation . . . . .	16-5
	Processing Flow . . . . .	16-5
	EXECs . . . . .	16-6
	Rules . . . . .	16-6
<b>Chapter 17</b>	<b>E-Mail Solution</b>	
	EXECs . . . . .	17-1
<b>Chapter 18</b>	<b>SNMP Solution</b>	
	EXECs . . . . .	18-1
<b>Chapter 19</b>	<b>Using the MAINVIEW AutoOPERATOR Web Sample</b>	
	Introduction . . . . .	19-1
	What AutoOPERATOR Web Is . . . . .	19-2
	How AutoOPERATOR Web Is Used . . . . .	19-2
	What the Sample Web Pages Look Like . . . . .	19-3

Prerequisites . . . . .	19-3
What You Need to Know . . . . .	19-3
Warning about Case Sensitivity . . . . .	19-4
What Information to Gather . . . . .	19-4
What Software Must Be Installed . . . . .	19-5
Installing and Customizing the Samples . . . . .	19-6
Running the Installation EXEC . . . . .	19-6
Configuring the HTTP Server . . . . .	19-6
Making BBSAMP Available to AutoOPERATOR . . . . .	19-7
Meeting AOAnywhere Requirements . . . . .	19-7
Reviewing Security . . . . .	19-8
Accessing Multiple Systems . . . . .	19-8
Accessing the Samples . . . . .	19-9
Viewing the Entry Page . . . . .	19-9
Viewing the About Page . . . . .	19-10
Viewing the CSM Page . . . . .	19-11
Viewing the ALERTs Page . . . . .	19-14
Viewing the Automation Reporter Page . . . . .	19-17
Viewing the Commands Page . . . . .	19-20
Debugging the Distributed CGI Scripts . . . . .	19-21
Understanding CGIs . . . . .	19-22
Debugging the CGI Script without HTML Decoding . . . . .	19-22
Avoiding Frames in the Display . . . . .	19-22
Displaying the HTML Source in a Frame . . . . .	19-23
Refreshing HTML Source Code . . . . .	19-23
Common Error Messages . . . . .	19-23

## Chapter 20

### Using the Paging Sample

EXECs . . . . .	20-1
QAOSNPP1 . . . . .	20-1
QAOSNPP2 . . . . .	20-2
Processing Flow . . . . .	20-3
Variables Returned from QAOSNPP2 . . . . .	20-4

## Chapter 21

### Using Sample Solutions

Distributed Sample Solutions . . . . .	21-1
Naming Conventions for EXECs . . . . .	21-1
Standards for Sample Solutions . . . . .	21-6
Sample VTAM Solutions . . . . .	21-6
VTAM/NCP Solution . . . . .	21-6
VTAM Commands Solution . . . . .	21-7
Sample Monitoring Solution . . . . .	21-9
Sample Operations Monitors Solution . . . . .	21-9
Sample CICSplex Solution . . . . .	21-12
Invoking the Sample CICSplex Solution . . . . .	21-12
EXECs . . . . .	21-13
Rules . . . . .	21-13

Sample Info/System Management Solution . . . . .	21-14
Invoking the Sample Info/System Management Solution . . . . .	21-14
EXECs for the Sample Info/System Management Solution. . . . .	21-15
Rules for the Sample Info/System Management Solution . . . . .	21-15
Sample MVALARM Solution. . . . .	21-15
Enabling the MVALARM Solution . . . . .	21-16
Stopping the MVALARM Solution . . . . .	21-16

**Appendix A**

**Diagnosing CSM Errors**

Creating a Cross-Reference Report. . . . .	A-1
Gathering Environmental Information . . . . .	A-3
Generating CSM Exec Diagnostics for Support. . . . .	A-3
Tracing Starts and Stops of CSM EXECs. . . . .	A-4
Tracing All Activity in a CSM EXEC . . . . .	A-5
Stopping a Trace . . . . .	A-5
Creating CSM Start and Stop Rules . . . . .	A-6
Creating CSM Start and Stop Rules for the APPC Object. . . . .	A-6
Creating CSM Start and Stop Rules for the TCPIP Object . . . . .	A-7
Common Questions and Answers. . . . .	A-8
Starting/Stopping Errors. . . . .	A-8
Restart-Only Objects. . . . .	A-10
Defining Special Objects . . . . .	A-10
CSMACT Usage . . . . .	A-14
BBI Terminal Sessions (BBI-TS) . . . . .	A-16
TSO/E or Environmental Problems . . . . .	A-16
CSM Installation. . . . .	A-16
Effective Implementation . . . . .	A-17
Other Tips and Questions . . . . .	A-18

**Appendix B**

**Sample EXECs**

@TIMER . . . . .	B-1
JESDOWN . . . . .	B-1

**Appendix C**

**Sample REXX EXECs**

REXX Compound Variable Initialization EXECs. . . . .	C-2
REXX DASD Reserve Analysis EXEC . . . . .	C-4
REXX QUEUE/PULL and VPUT/VGET Usage . . . . .	C-8

**Appendix D**

**MVS Solution Variables**

**Appendix E**

**CICS Solution Variables**

**Appendix F**

**DB2 Solution Variables and Parameters**

Data Sets Containing Variables. . . . .	F-1
Data Sets Containing Threshold Parameters . . . . .	F-2
Tables That Describe Variables and Parameters . . . . .	F-2

**Index**

---

---

# About This Book

This book documents MVS, CICS, DB2, and many other solutions that are distributed by BMC Software to run under MAINVIEW AutoOPERATOR. This book is intended for installers who install MAINVIEW AutoOPERATOR solutions and administrators who maintain MAINVIEW AutoOperator solutions.

Throughout this book, references to OS/390 support also include support for MVS and z/OS.

## How This Book Is Organized

This book is organized as follows. In addition, this book contains a glossary of terms and an index.

Chapter/Appendix	Description
Chapter 1, "Introduction to MAINVIEW AutoOPERATOR Solutions"	introductory material that applies to all solutions
Chapter 2, "Managing Critical Applications with Continuous State Manager"	describes how to use the Continuous State Manager application (also referred to by the acronym CSM)
Chapter 3, "JES2 Solution"	describes how to use the JES2 Monitoring solution application that alerts data center personnel of any JES2-related problems.
Chapter 4, "Storage Subsystems Solutions"	describes how to use the TCAS reply solution, which provides automatic replies to TCAS WTOR
Chapter 5, "Monitoring Solutions"	describes how to use the various BMC Software monitoring solutions that are designed to assist data center personnel in monitoring events occurring in the system
Chapter 6, "Access Methods Solution"	describes how to use the Access Methods solution, which is designed to alert the data center personnel when failures occur during I/O processing

<b>Chapter/Appendix</b>	<b>Description</b>
Chapter 7, "Job Scheduling Solution"	describes how to use the Job Scheduling solution, which can be used to assist data center personnel in scheduling and managing time-dependent tasks
Chapter 8, "CONTROLM Solution"	describes how to use the CONTROLM solution, which provides a command interface to CONTROLM
Chapter 9, "MVS Performance Management Solutions"	describes how to use the MVS performance management solutions, which are designed to assist data center personnel in monitoring and adjusting system parameters that affect system performance
Chapter 10, "Error Recovery Solutions"	describes how to use the Error Recovery solutions to assist data center personnel in handling critical errors that occur in the CICS environment
Chapter 11, "CICS Performance Management Solutions"	describes how to use the CICS Performance Management solution to assist data center personnel to achieve maximum availability and increased response time
Chapter 12, "VTAM/NCP Solution"	describes how to use the TCAS Reply solution, which provides automatic replies to TCAS WTORs
Chapter 13, "Initial Customization Steps"	describes the steps to take to customize your automation environment to use the DB2 Solutions
Chapter 14, "DB2 Global Operations Solutions"	describes the initialization routines as well as utilities and minimum alert management for DB2
Chapter 15, "DB2 Resource Contention Analysis Solutions"	concentrates on a few major resources that are critical to DB2 operations
Chapter 16, "DB2 Performance Management Solutions"	describes the DB2 Performance Management solutions, which can be used to assist data center personnel achieve maximum availability and increased response time
Chapter 17, "E-Mail Solution"	contains a pair of sample EXECs that can be used to send an e-mail from a REXX EXEC
Chapter 18, "SNMP Solution"	contains a pair of sample EXECs that can be used to generate SNMP Traps
Chapter 19, "Using the MAINVIEW AutoOPERATOR Web Sample"	contains MAINVIEW AutoOPERATOR web samples that demonstrate MAINVIEW AutoOPERATOR automation possibilities in Web browser environments
Chapter 20, "Using the Paging Sample"	contains sample code that enables you to communicate with wireless devices, including both text and numeric pagers, which could be one- or two-way devices
Appendix A, "Diagnosing CSM Errors"	describes how to create a cross-reference report and how to gather information about your system environment; concludes with common questions and answers
Appendix B, "Sample EXECs"	includes information about the @TIMER and JESDOWN EXECs
Appendix C, "Sample REXX EXECs"	includes sample EXECs for REXX
Appendix D, "MVS Solution Variables"	lists all the MVS Solution variables, their default values, and the solutions that each one affects.
Appendix E, "CICS Solution Variables"	lists all CICS Solution variables, their default values, and the solutions that each one affects

---

## Related Documentation

BMC Software products are supported by several types of documentation:

- online and printed books
- release notes and other notices

## MAINVIEW AutoOPERATOR Product Library

MAINVIEW AutoOPERATOR is available with the following options:

- MAINVIEW AutoOPERATOR for OS/390
- MAINVIEW AutoOPERATOR for IMS
- MAINVIEW AutoOPERATOR for CICS
- MAINVIEW AutoOPERATOR Access NV
- MAINVIEW AutoOPERATOR TapeSHARE
- MAINVIEW AutoOPERATOR for MQSeries
- MAINVIEW AutoOPERATOR Elan Workstation
- MAINVIEW AutoOPERATOR for SAP High Availability

---

The base product and these options are documented in the following MAINVIEW AutoOPERATOR manuals:

- *MAINVIEW AutoOPERATOR Customization Guide*
- *MAINVIEW AutoOPERATOR Basic Automation Guide*
- *MAINVIEW AutoOPERATOR Advanced Automation Guide*
- *MAINVIEW AutoOPERATOR Options User Guide*
- *MAINVIEW AutoOPERATOR for MQSeries Installation and User Guide*
- *MAINVIEW AutoOPERATOR Reference Summary*
- *MAINVIEW AutoOPERATOR Solutions Guide*

## Related Reading

To customize some products, you might need additional information not documented in this manual. Refer to the following publications:

- *MAINVIEW for DB2 User Guide*, which describes how to use the online monitor and analyzer services for use by the DB2 database administrator, system programmer, or other performance analyst
- *MAINVIEW Common Customization Guide*, which describes how to customize the product manually; includes common steps for all MAINVIEW products
- *MAINVIEW Administration Guide*, which includes administrative procedures
- *Using MAINVIEW*, which provides general information about how to use MAINVIEW products
- *IBM Advanced Communications Function for VTAM (ACF/VTAM)*, SC38-0256, which describes VTAM parameters for subsystem communication
- *IBM System Modification Program Extended*, SC28-1107, which gives a complete explanation of the IBM System Modification Program (SMP)

---

# Product Requirements

You must have MAINVIEW AutoOPERATOR installed to run MAINVIEW AutoOPERATOR solutions. In addition, the following MAINVIEW components are required for Rule Sets as shown in the following table.

<b>MAINVIEW Components</b>	<b>Rule Set: TYPE</b>	<b>AAORUL00 MVS</b>	<b>AAORULM1 MVS</b>	<b>AAORULM1 CICS</b>	<b>AAORULD1-D8 DB2</b>
MAO - MV/AO for OS/390	Yes	Yes	Yes	Yes	Yes
CAO - MV/AO for CICS			Yes	Yes	
IAO - MV/AO for IMS					Yes
RES + (Full RESOLVE product) - MV SYSPROG SVCS			Yes		
IMF - MV/IMS					Yes
CMR - MV/CICS				Yes	
DMR - MV/DB2					Yes

---

# Naming Conventions for EXECs

The naming convention for the MAINVIEW AutoOPERATOR solutions EXECs uses the following format:

**gffnnnl**

where

**g** first letter of the group (for example, M for MVS, I for IMS, C for CICS, D for DB2)

**ff** two-character abbreviation for the functional area within the group:

<b>Code</b>	<b>Functional Area</b>
AM	access methods
ER	error recovery
IN	info sys/management
J2	JES2
J3	JES3
JS	job scheduling
MN	monitors
PE	performance
SD	shut-down
ST	storage subsystems
SU	start-up
UT	utilities
VT	vtam/ncp

**nnn** number from 001 to 999

**l** language of the particular EXEC (for example, X for REXX, C for CLIST)

---

The following two categories of EXECs do not follow this naming convention:

- operator-initiated EXECs, which are executed as a result of an operator command
- text ID-driven EXECs, which are executed as a result of a Write-To-Operator (WTO) or a Write-To-Operator Reply (WTOR)

The names for text-ID-driven EXECs are the text-IDs.

## Online and Printed Books

The books that accompany BMC Software products are available in online and printed formats. Online books are formatted as Portable Document Format (PDF) files. Some online books are also formatted as HTML files.

### To Access Online Books

To view any online book that BMC Software offers, visit the Customer Support page of the BMC Software Web site at <http://www.bmc.com/support.html>. You can also access PDF books from the documentation compact disc (CD) that accompanies your product.

Use the free Acrobat Reader from Adobe Systems to view, print, or copy PDF files. In some cases, installing the Acrobat Reader and downloading the online books is an optional part of the product-installation process. For information about downloading the free reader from the Web, go to the Adobe Systems site at <http://www.adobe.com>.

### To Request Additional Printed Books

BMC Software provides some printed books with your product order. To request additional books, go to <http://www.bmc.com/support.html>.

## Release Notes and Other Notices

Printed release notes accompany each BMC Software product. Release notes provide current information such as

- updates to the installation instructions

- 
- last-minute product information

In addition, BMC Software sometimes provides updated product information between releases (in the form of a flash or a technical bulletin, for example). The latest versions of the release notes and other notices are available on the Web at <http://www.bmc.com/support.html>.

---

# Chapter 1 Introduction to MAINVIEW AutoOPERATOR Solutions

BMC Software provides a platform for developing automated solutions to problems within a data center. This platform consists of Rules, EXECs, timer facilities, and SYSPROG Services commands.

Using this platform, BMC Software has developed solutions to problems that are common across many data centers. These solutions assist you in accomplishing many of the initial automation tasks quickly and efficiently after you install the MAINVIEW AutoOPERATOR product.

## EXEC Distribution

All EXECs for solutions are distributed in the BBPROC data set allocated to the SYSPROC DD statement in the MAINVIEW AutoOPERATOR subsystem.

## Rules Distribution

**Note:** All Rules for MAINVIEW AutoOPERATOR Solutions are distributed DISABLED in the BBPARM data set.

The distribution of Rules within this data set differs depending on the environment of the solution.

**Table 1-1** Rules Distribution in Different Environments

Solution Environment	Distribution of Rules
MVS and CICS	one set of Rules for all solutions in one member (AAORULM1 and AAORULC1, respectively)
DB2	one set of Rules for each solution in a separate member (AAORULD*)

For information on how to enable any of these sets of Rules, any Rules within the individual solutions, or other sets of Rules in the MAINVIEW AutoOPERATOR product, see the Rules chapters of the *MAINVIEW AutoOPERATOR Basic Automation Guide*.

## Documentation Boxes

The MAINVIEW AutoOPERATOR Solutions EXECs are distributed with a documentation box. Figure 1-1 on page 1-3 shows an example of a documentation box.

**Figure 1-1 Example of a Documentation Box**

```

/*****
/* DOC GROUP(MVS) FUNC(MONITORS) CODE(MN)
/* DOC DISP(YES) AUTHOR(BMC)
/* DOC DESC(RESOLVE WTO BUF SHORTAGE)
/*-----
/* NAME:
/*     IEE249I
/*
/* DESCRIPTION:
/*     RESOLVE THE WTO BUFFER SHORTAGE SITUATION
/*
/* SEE ALSO:
/*     IEA404A
/*     IEA405E
/*
/* INVOKED BY:
/*     IEA404A
/*     IEA405E
/*
/* INPUT PARAMETERS:
/*     NONE
/*
/* OUTPUT:
/*     NONE
/*
/* OUTBOARD CALLS:
/*     NONE
/*
/* EXTERNAL ROUTINES CALLED:
/*     MVS "K" OR "CONTROL" COMMANDS ARE ISSUED.
/*     NOTE: FOR THE "K" COMMANDS TO BE PROPERLY AUTHORIZED, THEY
/*     ---- MUST BE ISSUED FROM A NON-SUBSYSTEM CONSOLE.  EDIT
/*           THE MEMBER, BBISSP00, IN YOUR BBPARM DATA SET AND
/*           ADD THE KEY WORD, CMDCON=1; THIS WILL DIRECT ALL COM-
/*           MANDS WHICH DO NOT REQUIRE A RESPONSE TO CONSOLE
/*           ID 1.
/*
/* ALERTS ISSUED:
/*     NONE
/*
/* VARIABLES USED:
/*     NAME:      DESCRIPTION:      DEFAULT VALUE:
/*     MSG        MESSAGE ID        IEE249I
/*     P1          WORD 1            CUA OF CONSOLE OR SYSLOG
/*     P2          WORD 2            CONSOLE ID
/*     P3          WORD 3            COND=
/*     P4          WORD 4            AUTH=
/*     P5          WORD 5            NBUF=
/*
/* TEST SCRIPT INFO:
/*     GENERATE A WTO FROM A TEST EXEC TO TRIGGER A CALL TO
/*     EITHER OF THE DRIVER EXECS: IEA404A OR IEA405E
/*
/* RETURN CODES:
/*     NONE
/*
/* CHANGE LOG:
/*     03-05-92 BY MDR: ADDED DOCUMENTATION BOX

```

The information contained at the top of the documentation box is used by the MAINVIEW AutoOPERATOR EXEC Management Application to display information about the EXECs in the SYSPROC libraries. GROUP, FUNC, CODE, DISP, AUTHOR, and DESC are displayed on one line by the MAINVIEW AutoOPERATOR EXEC Management Application (refer to the *MAINVIEW AutoOPERATOR Basic Automation Guide* for a detailed explanation). The rest of the information in the documentation box is for self-documentation and easy reference.

## Documentation Box Fields

Table 1-2 describes the title fields shown in Figure 1-1 on page 1-3.

**Table 1-2 Documentation Box Fields**

Field	Description
GROUP	group to which this EXEC belongs (for example, MVS, CICS, DB2, or IMS)
FUNC	functional category within the group
CODE	two-character function code
DISP	whether information about this EXEC should be displayed under the MAINVIEW AutoOPERATOR EXEC Management Application (refer to the <i>MAINVIEW AutoOPERATOR Basic Automation Guide</i> )
AUTHOR	author of the EXEC
DESC	description of what the EXEC does (25 characters or fewer)
NAME	name of the EXEC
DESCRIPTION	detailed description of what the EXEC does
SEE ALSO	cross-reference listing
INVOKED BY	list of other EXECs that call this one
INPUT PARAMETERS	parameter name, description, and default value required for input parameters
OUTPUT	information or names of variables that are returned
OUTBOARD CALLS	whether any pager calls have been placed
EXTERNAL ROUTINES CALLED	other routines or services called, along with any special considerations for those calls
ALERTS ISSUED	brief text of long EXECs where ALERTs are issued; otherwise, a YES or NO is returned
VARIABLES USED	variable name, description, and default value of any variables used
TEST SCRIPT INFO	necessary environment and steps that are needed to test the EXEC if changes are required
RETURN CODES	what return codes (if any) are returned
CHANGE LOG	chronology of changes

## Notes about Solutions

Some national characters do not translate correctly when you translate from EBCDIC format to ASCII format and then back to an EBCDIC format. In particular, the national characters #, !, %, and @ translate differently depending on the country in which the translation takes place. BMC Software recommends that you avoid using national characters in your naming conventions.

You might want to customize or modify the solutions. If you do modify a solution, BMC Software recommends that you retain the original EXEC or Rule for reference. To do so, follow these steps:

- Step 1** Copy the solution's components into the UBBPROC data set.
- Step 2** Make the modifications.
- Step 3** Concatenate the UBBPROC data set to the front of the BBPROC data set in your BBI-SS JCL.



---

---

# Chapter 2 Managing Critical Applications with Continuous State Manager

This chapter describes the Continuous State Manager (also referred to by the abbreviation CSM) application and includes the following discussions.

**Note:** BMC strongly recommends that you use the MAINVIEW Total Object Manager (TOM) application for managing Started Tasks. For information about this product, refer to the *MAINVIEW Total Object Manager User Guide*.

To Read about	See Page
"What Continuous State Manager Is"	2-2
"Defining CSM Terms and Concepts"	2-4
"How CSM Works"	2-16
"How to Effectively Plan for CSM Implementation"	2-32
"Before You Begin"	2-37
"What the CSM Global Overview Panel Is"	2-44
"Defining CSM Objects"	2-51
"Performing Routine Object Tasks"	2-110
"Displaying and Modifying Object States"	2-130
"Using the Command Line Interface"	2-139
"Describing CSM Panels"	2-142

# What Continuous State Manager Is

To best understand how to use Continuous State Manager and how it can help you automate critical processes, you must understand four basic concepts:

- objects, page 2-5
- relationships between objects, page 2-6
- states of objects, page 2-9
- schedules, page 2-11

***Objects Relationships*** With the ISPF dialogs provided by Continuous State Manager, you define *objects* (for example, CICS, JES, or NetView) and establish *relationships* and *dependencies* between them, thereby enabling the application to automatically manage your system in an ordered way.

***States*** To refer to an object's accessibility is to describe the *state* of the object. An object that is available is in an UP state and when it is not available, it is in a DOWN state.

***Schedules*** By default, Continuous State Manager assumes that any object you define is UP 24 hours a day, 7 days a week. The object's *schedule* is what the application uses to determine when to make objects UP (available) or DOWN (unavailable). The schedule consists of all the times an object should be down. For example, an object might need to be brought down every Thursday from 1:00 A.M. to 3:00 A.M. for maintenance.

If you do not define a schedule for an object to be DOWN, Continuous State Manager creates a default schedule where the object is UP 24 hours a day, 7 days a week.

Ideally, by using CSM, you are enabling important processes such as IPL and orderly shutdowns to be managed and controlled automatically. However, if you do encounter situations where you must change the state of an object manually, CSM also provides a facility that enables you to do this modification. The section "Using the CSMACT EXEC to Modify Object States Manually" on page 2-135 contains more information about manually modifying an object's state.

---

# What the Benefits of CSM Are

CSM improves data center performance by

- eliminating potential errors due to manual interventions

Once CSM begins managing your system, manual interventions should become minimal and hence, the system operates more smoothly, with fewer errors.

- reducing IPL time up to 50%

One of the key processes that CSM performs is OS/390 IPL, which reduces the amount of time spent performing an IPL.

- reducing the time needed for an orderly system shutdown

Likewise, CSM can be used to perform an orderly, controlled shutdown of the system that also saves time.

- restarting failed objects (if needed)

Once they are properly defined to CSM, all the Started Tasks can be controlled and managed by CSM, including the automatic recovery of failed Started Tasks.

- starting and stopping groups of objects as if they were a single object

For example, you can use CSM to start or stop all your CICS regions with a single command.

- notifying you of problems

For example, you can be notified automatically about objects that have changes of states outside of CSM control or about objects that fail to start or stop correctly.

- enabling you to manage objects defined on different BBI subsystems (BBI-SS PASs)

For example, CSM enables you to see objects defined on another BBI-SS PAS target and issue commands to start, stop, cancel, and bounce those objects.

## What CSM Requires

The MAINVIEW AutoOPERATOR for OS/390 option is required for CSM to operate and a valid key for MAO must be present in the BBPARM member BBKEYS. For more information about BBKEYs, refer to the *MAINVIEW Common Customization Guide*.

The MAINVIEW AutoOPERATOR for OS/390 option is required for CSM processing, which includes issuing OS/390 commands and using SYSPROG services. For more information about activating CSM, refer to the *MAINVIEW AutoOPERATOR Customization Guide*.

## Defining CSM Terms and Concepts

- “What an Object Is” on page 2-5.
- “What Object Relationships Are” on page 2-6
- “What an Object State Is” on page 2-9
- “What a Schedule Is” on page 2-11
- “What Else CSM Must Know and Why” on page 2-13
- “Using CSM Panels” on page 2-40

## CSM Terms and Concepts

You should be familiar with the following CSM terms and concepts:

<b>Term</b>	<b>Definition</b>
CSM-PLEX	a group of BBI-SS PASs, each of which is running CSM and share a common object repository
CSM partner	one BBI-SS PAS within a CSM-PLEX
object group	a set of objects managed by a CSM partner

An object can be associated with one, several or as many as 20 object groups. Object groups replace the need for alternate databases that were used in previous releases of MAINVIEW AutoOPERATOR because once an object is defined and is associated with an object group, it (and its schedule) can be shared with other BBI-SS PASs on other OS/390 system images that are running CSM.

The name of an object group can be up to eight alphanumeric characters in length.

**Object repository** a keyed VSAM data set shared by all members of the CSM-PLEX

A record of this data set contains all of the information related to a particular object.

The object repository enables you to define CSM objects and use them and their schedules across all the CSM-PLEX.

## What an Object Is

You can define the following different types of objects in CSM:

- normal
- transient
- grouping

### Normal Objects

A normal object is a Started Task (STC) that, once defined to CSM, has its state completely controlled by CSM. Examples of these objects are TSO, CICS regions, and JES.

CSM becomes completely responsible for automatically starting, stopping and, in some cases, restarting or recovering normal objects according to the schedule you specify when you define the object. For information about how to define normal objects, refer to “Entering the Object’s Information” on page 2-55.

### Transient Objects

A transient object is a Started Task (STC) which is started by CSM once during the life of CSM but its state is *not managed by CSM* after it is started. This STC might be an object that starts, performs a batch process, and terminates normally when the job is completed. CSM will start these objects and from that point on, CSM ceases to monitor or manage the object.

When a transient object is started, its state is COMPLETE, where a normal object is UP. In other words, CSM does not monitor whether a transient object is UP or DOWN and it never attempts to change the object’s state. For information about how to define transient objects, refer to “Defining a Transient Object” on page 2-74.

## Grouping Objects

A grouping object is a user-defined imaginary object that enables you to group other objects together. This function is very important within CSM because by using grouping objects, you can *start* or *stop* multiple objects with a *single command*.

For example, you might decide to stop all your CICS regions (CICS1, CICS2, CICS3) together. You can create a grouping object, CICSPROD, to be a parent to all these objects (refer to “Parent-Child Relationships” on page 2-7). By stopping object CICSPROD, the children CICS1, CICS2, and CICS3 stop automatically, too.

The grouping object is defined either as a parent to a group of objects or as a child to a group of objects; therefore, you can use grouping objects to start or stop groups of objects.

In this example, if you define a grouping object named CICSPROD2 as a child to CICS1, CICS2, and CICS3 and then start CICSPROD2, the three parent objects will automatically start. For more information about grouping objects and other relationships between objects, refer to What Object Relationships Are. For information about how to define grouping objects, refer to “Defining a Grouping Object” on page 2-75.

## What Object Relationships Are

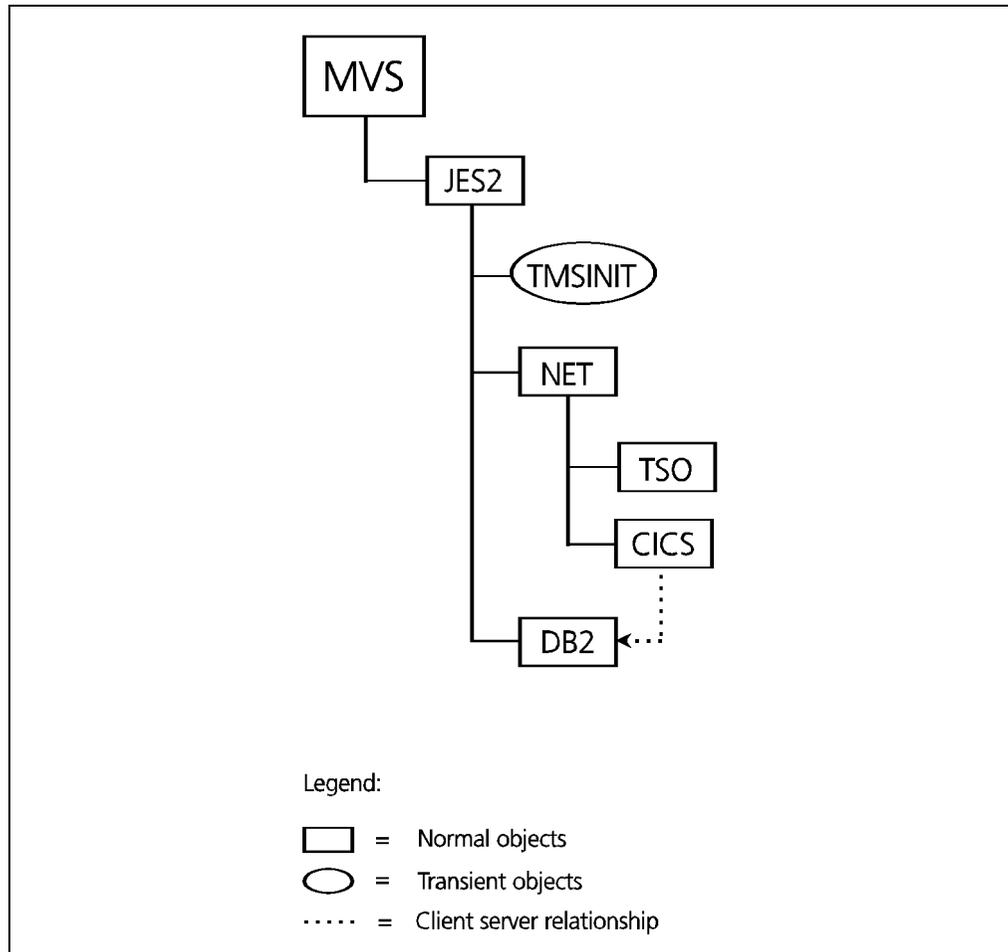
Another aspect of an object’s definition is its relationship to other objects. When objects are defined to CSM, they are defined according to their relationships to other objects in the system. You can define the following types of object relationships:

- parent-child
- client-server

You can easily define relationships between the objects in CSM and there is no limitation to the number of relationships you can establish between objects.

The following sections describe these types of relationships.

Figure 2-1 CSM Relationships



### Parent-Child Relationships

A parent-child relationship is one in which the child object is completely dependent on the parent object for operation. For example, NET (the VTAM address space) must be fully active before starting TSO, so TSO is a child (or dependent) of parent NET. Refer to Figure 2-1.

In a parent-child relationship, when a parent object is stopped, CSM automatically stops all the child objects. Likewise, when a child object is started, CSM automatically starts the parent objects.

Every object must have at least one parent. Therefore, since all objects cannot be active without OS/390, OS/390 is ultimately the parent of all objects in the system, although in CSM, you need to define only an object's direct parents. You do not need to define "grandparents" of objects.

To continue with the example, NET (a child of JES2) must be started before TSO starts so you must define NET as a parent of object TSO. (Note that you do not need to define JES2 and OS/390 as direct parents of TSO.) However, when CSM stops JES2, both NET and TSO are stopped first. When TSO is started, the parents NET and JES2 are started first.

For information about how to define parent-child relationships refer to “Defining the Object’s Parents” on page 2-71.

### Client-Server Relationships

In a client-server relationship, the client object receives part, but not all, of its service from a server object. In this relationship, the server object can be stopped *without stopping the client object*. For example, in Figure 2-1 on page 2-7, server DB2 could be stopped without stopping client CICS. For more information about how to define client-server relationships, refer to “Defining the Object’s Servers” on page 2-72.

The following table describes all of the CSM object relationships that are shown in Figure 2-1 on page 2-7.

Object	Object Type	Parent	Child	Client	Server
OS/390	NORMAL	T Parent of JES2			
JES2	NORMAL	T Parent of NET, TMSINIT, and DB2	T Direct child of OS/390		
TMSINIT	TRANSIENT		T Child of JES2		
NET	NORMAL	T Parent of TSO and CICS	T Direct child of JES2		
DB2	NORMAL		T Direct child of JES2		T Server of CICS
CICS	NORMAL		T Direct child of NET	T A client of DB2	
TSO	NORMAL		T Direct child of NET		
<p><b>Note:</b> Because OS/390 must be active for any other object to be active, OS/390 is by definition, the parent of all objects in the system. However, you do not need to explicitly define OS/390 as every object's parent. This table shows only the direct relationship of every object.</p>					

CSM can manage complex object configurations in your data center. You might find it helpful to create a diagram of the system and the relationships that objects should have before you begin defining objects to CSM.

## What an Object State Is

Two categories of states exist in CSM:

- object desired states

When you define an object to CSM, you can include a schedule of when the object is supposed to be DOWN. The desired state of an object is the state of the object according to how you scheduled the object.

In other words, if the object is supposed to be UP 24 hours a day, Monday through Friday, its desired state is that the object is UP during the week and DOWN on the weekends. If you do not define a schedule for an object, CSM creates a schedule for the object where the object's desired state is UP 24 hours a day, 7 days a week.

For more information about desired states, refer to “Object Desired States” on page 2-10.

- object actual states

An object's actual state is the current status of the object and it might not always match the desired state.

CSM attempts to *change the actual state of each object to match the desired state*; for example, if CSM finds that the actual state of an object is DOWN when it is scheduled to be UP (the desired state), CSM tries to start the object.

CSM also recognizes the actual states that exist between UP and DOWN (see “Actual (Intermediate) States” on page 2-10).

For information about determining an object's state, refer to “Displaying and Modifying Object States” on page 2-130.

## Object Desired States

Every time an event affects an object, CSM first modifies the desired state. An event that affects the state of an object can be

- a scheduled change that specifies the object must be brought DOWN or brought back UP

For example, a planned IPL or shutdown of the system.

- a manual command that affects the state of the object

For example, an unplanned IPL or shutdown of the system.

CSM checks the database where the object is defined and attempts to change the actual state of the object to match the desired state. CSM can change not only the object's actual state but also the object's children or parents' states.

If, for any reason, CSM cannot change the actual state to match the desired state, an ALERT is issued that explains the discrepancy. The ALERT

- uses a Help panel to explain the discrepancy
- proposes a solution
- enables you to attempt to correct the situation by entering a response in the ALERT's response field

For more information about how an object's state can change or what state an object can be in, refer to "Displaying and Modifying Object States" on page 2-130.

## Actual (Intermediate) States

Between the UP and DOWN states, CSM recognizes intermediate states that are between UP and DOWN. They are useful to both CSM and to you for keeping track of the life of an object.

For example, when an object is brought UP (the desired state), the actual states that the object passes through are

- DOWN
- STARTING
- UP

If any object stays in the STARTING state, probably some problem exists with your object definitions. The problem is usually in the defined commands or the message IDs that are used to indicate that the object is UP.

Some intermediate states also are specific to the type of object. For example, the COMPLETE state is equivalent to UP for only transient or grouping objects. Normal objects do not have a state of COMPLETE.

Another instance of an intermediate state is when you manually change the object's state. For example, if you bring an object UP while the object is in the middle of scheduled DOWN period, the desired state is UPEARLY (because you want the object UP earlier than planned).

Use the CSM Global Overview panel to see the actual and desired states for every object defined to CSM. Refer to "What the CSM Global Overview Panel Is" on page 2-44 for more information.

## What a Schedule Is

From "What an Object Is" on page 2-5, you know that once you have created an object, it is scheduled to be UP all the time unless you specify a schedule of DOWN times. You will find that some objects need to be DOWN and this state is accomplished by creating a schedule as part of the object's definition.

CSM enables you to define a weekly schedule for all objects. Once the object is defined with a schedule, CSM tries to match the desired state (defined by the schedule) with the object's actual state. CSM manages the object according to the schedule, automatically bringing it UP or DOWN at the specified times.

When the object has children, they are brought UP and DOWN automatically with the parent object's schedule. When the object has clients, they are disconnected from it and reconnected to it.

Two types of object schedules can be defined: routine schedules and exception schedules (where a routine schedule is overridden).

## Routine Scheduling for Objects

A routine schedule consists of the day of the week and the time of each day that an object is to be brought DOWN. The following example shows a schedule for when an object should be DOWN:

```
Wednesdays from 3:00 P.M. to 7:00 P.M.  
Fridays from 2:00 A.M. to 3:00 A.M.  
Fridays from 6:00 A.M. to 9:00 A.M.  
Sundays from 3:00 P.M. to 6:00 P.M.
```

Outside of these times, CSM assumes the object should be UP.

When you define objects to CSM, you should plan and compare the schedule that the object needs with the schedule of any children objects. This planning will help you when an object's parent might have a schedule that conflicts with the child because *stopping the parent at the scheduled time will stop the child first*.

Therefore, the schedule for a child object must include the DOWN time of its parents. If it does not, a message prompts you during the definition of the object to correct the situation.

You can also schedule DOWN times of CSM objects with

- DAILY  
specifies that an object should be DOWN at a specified time every day of the week (Monday through Sunday) every week
- WEEKDAY  
specifies that the object should be DOWN at a specified time on Monday through Friday of every week
- WEEKEND  
specifies that the object should be DOWN at a specified time on every Saturday and Sunday

For more information about creating schedules for objects, refer to "Modifying an Object's Routine Schedule" on page 2-93.

## Exceptional Scheduling for Objects

CSM enables you to create Global Calendar Overrides for special one-time situations where your plans require that objects must be brought UP or DOWN outside the routinely scheduled UP or DOWN time.

For example, you might discover that for a particular day an object that is scheduled to be DOWN at 4:00 P.M. might need to stay UP for an additional hour. You can schedule a Global Calendar Override for the object to stay UP until 5:00 P.M. for that one day.

The Global Calendar Overrides are valid only once. You want to use Global Calendar Overrides to alter an object's schedule whenever you foresee a change to routine schedules.

**Important Note**

The Global Calendar Override does not check for conflicting schedules between parents and dependents. Therefore, when an override is specified for an object, that object and *all its dependents* are taken down on the specified date and time.

When the object is returned to its routine schedule, the dependents also are brought back on schedule, which eliminates the need to specify the special down times for each object affected by the temporary schedule change.

For more information about creating a calendar override, refer to “Overriding the Calendar for Special Situations” on page 2-106.

## What Else CSM Must Know and Why

In addition to defining objects, their relationships, and schedules, you must define additional information that CSM uses to actually manage the objects. This information includes

- which commands are to be executed to start, stop, or cancel the object

CSM uses OS/390 commands (or EXECs) that you specify to start, stop, and cancel objects.

- which messages indicate to CSM that an object is UP or DOWN

CSM uses OS/390 message IDs (also called text-IDs) that are generated by objects when they are started or stopped to keep track of object states.

- which connect and disconnect commands are to be used when CSM manages client-server relationships

CSM uses the connect or disconnect commands that you specify to create client-server relationships between objects.

## How CSM Starts, Stops, and Cancels Objects

In the object's definition you must specify the actual OS/390 command or EXEC that CSM uses when CSM starts, stops, or cancels the object. Cancel commands are used by CSM when a stop command does not work. CSM executes these commands when an IPL, a shutdown, or an event affects an object.

You can specify either the OS/390 command that you might use to start, stop, or cancel the object from an OS/390 console or a user-written EXEC that issues the command.

You might choose to create an EXEC because you need several commands to start the object or you want to perform some additional function such as set a variable when the object starts, stops, or is canceled.

You must enter these start, stop, and cancel commands or EXECs on the same object definition panel where new objects are created. They are a mandatory part of an object's definition.

## How CSM Recognizes when an Object Starts or Stops

The OS/390 messages that are generated by objects when they start or stop are very important to CSM processing. These messages are used by CSM to keep track of the object states.

CSM uses the text-IDs to automatically generate Rules that are fired every time a message is received that signals an object to either start or stop. These Rules are called Start or Stop Rules; in turn, they schedule CSM EXECs that change the actual states of the object. These Rules reside in a CSM Rule Set that is specified in the object group definition and is not user-modifiable. All CSM Rules are created with a strategy of FIRST.

You must enter these text-IDs for objects on the same object definition panel where new objects are created. They are a mandatory part of an object's definition.

CSM does provide default text-IDs for objects that trigger the Start and Stop Rules, but you can choose to specify other text-IDs. For example, for a CICS region starting, you might choose the message: `DFH1517 Region Control is being given to CICS`. This message is generated when the CICS region is completely operational.

## Creating Start and Stop Rules for CSM Objects

Occasionally, you might find that the CSM-generated Start and Stop Rules for an object do not meet all of your needs. If this situation happens, you can create your own CSM Start and Stop Rules for that object.

Note that, any time you create a Start Rule and Stop Rule, you must not attempt to add them to Rule Sets AAORULBC, AAORULBD, AAORULBE, AAORULBG or to the Rule Set identified in the object group definition. The Rules can be added to any other Rule Set. Refer to “Creating CSM Start and Stop Rules” on page A-6 for an example of creating Start and Stop Rules for an object.

For more information about how CSM works, refer to “How CSM Works” on page 2-16. For information about defining objects, start, stop, cancel commands, and text-IDs, refer to “Defining CSM Objects” on page 2-51.

## How CSM Connects and Disconnects Objects

You can define a client-server relationship between objects for an object that receives part of its service from another object. For example, you might have a client-server relationship between CICS (client) and DB2 (server).

CSM uses connect and disconnect commands to link these objects together after the objects have come UP. These connect and disconnect commands are specified as part of the server’s definition and are the same commands you might use if you issued them through an OS/390 console. For more information about defining client-server relationships, refer to “Defining the Object’s Servers” on page 2-72.

## How CSM Works

The following discussion describes how CSM actually works and the order that CSM performs its processes.

### When CSM Starts

CSM initializes when the BBI-SS PAS initializes. The first thing CSM does is read the CSM database where all the object definitions, their start and stop commands, and the text-IDs are stored. You can initialize CSM with a different group, for example in the event of a disaster recovery situation. For more information, refer to the *MAINVIEW AutoOPERATOR Customization Guide*.

The initialization phase checks the actual state of an object and attempts to make the actual state of the object match its desired state (which is defined by the schedule). Based on the schedules for each object, CSM issues the start commands for all the objects that are scheduled to be UP.

For example, if CSM is initialized at 9:00 A.M. Monday, CSM reads the database and finds all the objects that are, according to their schedules, supposed to be UP at 9:00 A.M. Monday.

CSM issues the start commands (or EXECs) for these objects, updates the actual state to *STARTING*, and waits for the messages that signify the objects have started. When CSM receives these messages, CSM Rules are triggered that mark the object's actual state as UP and CSM has successfully started the objects in your system.

In summary, when CSM receives an object's start message:

1. The CSM Start Rule fires.
2. The CSM Start Rule schedules the CSMUP EXEC to process the event and change the object's state to UP.

When CSM receives an object's stop message:

1. The CSM Stop Rule fires.
2. The CSM Stop Rule schedules the CSMDOWN EXEC to process the event and change the object's state to DOWN.

Sometimes the address space might terminate with an end-of-memory event. When MAINVIEW AutoOPERATOR detects an address space end-of-memory event:

3. The IMFEOM EXEC is scheduled, which you can set up to call the CSMEOM EXEC. This EXEC processes the event and schedules the CSMDOWN EXEC.

## Routine Scheduling

At initialization, CSM also determines which objects need to be brought UP or DOWN at their scheduled times. For these objects, CSM schedules time-initiated EXECs that will bring the objects DOWN or UP at their scheduled times.

For example, suppose that at 10:00 A.M. Monday, a CICS object is scheduled to be brought UP. When CSM initializes at 9:00 A.M., a time-initiated EXEC is scheduled to issue the start command defined for the CICS object at 10:00 A.M.

## Recovering Failed Objects

An object might fail in CSM in the following situations:

- The start command is issued for an object but the start message is not received within the Start Command Time Out period (as defined for each object).
- The object abends unexpectedly.

When CSM does not receive the start message for an object within the time-out period, CSM issues an ALERT. No automatic recovery is attempted but an ALERT is issued, notifying you of the failure.

However, in the second case, if an object abends and comes DOWN when it is not scheduled to be DOWN, CSM automatically attempts object recovery. When an object stops and is DOWN (actual state) and it is scheduled to be UP (desired state), CSM attempts to make the actual state match the desired state.

CSM attempts this action when CSM receives the stop message for the object and fires a Rule. The Rule checks the desired state of the object, and if the desired state (according to the object's schedule) is UP, CSM automatically issues the start command to try to start the object. If CSM cannot start the object, CSM issues an ALERT.

You can specify how many times CSM will issue the start command to recover an object; refer to “Specifying Start Command Limits.”

### Specifying Start Command Limits

You can define the number of times CSM attempts to start a failed object. Suppose you specify that the start command limit is three. Then, when CSM detects a failed object, CSM automatically issues the object’s start command, trying to restart the object (first attempt). If CSM cannot start the object, an ALERT is issued that states the object’s state is DOWN and restarting has failed.

If the object does start and fails again at some later time, CSM will issue the start command (second attempt). If restarting the object is successful and the object fails again, CSM will issue the start command (third and final attempt).

If the object fails a fourth time, CSM does not attempt to restart the object and issues an ALERT to notify you of the failure.

When an object has exceeded its Start Command Limit, it can be restarted in one of following ways:

- You can reply to the ACM750A ALERT.
- You can use the RESET line command (and answer Y (for yes) on the last prompt) on the Global Overview panel.
- You can issue the command **%CSMACT object RESETC** on the **COMMAND** line and start the object from the panel.
- You can schedule the EXEC **%CSMXLCMD object RETRY**.

### Specifying Abnormal Termination Events

As part of an object’s definition, you can specify an abnormal termination event. This event occurs when an object unexpectedly terminates (abends) and a message is issued. The abnormal termination event informs CSM that an object that is supposed to be UP is no longer UP.

If an abnormal termination message ID (or text-ID) is included as part of the object’s definition, when an abend occurs CSM receives the text-ID indicating the object has terminated. CSM automatically creates a MAINVIEW AutoOPERATOR ALERT to notify you about the abend. The state of the object is ABENDED.

In addition, you can specify that an EXEC or command is issued when CSM receives the text-ID. The command or EXEC can attempt to rectify the situation or even restart the object. For example, when an IMS database procedure ends unexpectedly, a database recovery job must be run before IMS can be restarted. You can specify that when an IMS event abends and CSM receives the abnormal termination text-ID, an EXEC is scheduled. The EXEC can run the database recovery program.

If an EXEC or command is not specified and an abnormal termination event is specified, when the event occurs CSM creates a MAINVIEW AutoOPERATOR ALERT stating that no recovery command or EXEC has been defined.

If a recovery command or EXEC does start the object, the object's state is UPMANUAL. To restart an object with CSM and maintain the object's state, issue the command:

#### **CSMACT object RESET**

from the recovery EXEC with IMFEXEC SELECT EXEC WAIT(NO) statement or as the recovery command. Restarting the object this way gives the object a state of UP.

## **Interfering with CSM Processing**

Ideally, you should use CSM to handle the routine management of all the objects in your system, which means that all objects are defined in proper relationship to other objects and the schedules are correctly and accurately planned. An override to the routine schedule of an object can be defined and CSM will handle the one-time deviation from a routine schedule.

However, you might decide to (or have to) start, stop, or cancel an object at the last minute. You can do these actions from CSM in three ways:

- from the CSM panels
- with the CSMACT EXEC
- with the CSM command line interface

For more information about the CSM command line interface, refer to "Using the Command Line Interface" on page 2-139.

If you use any of these methods, CSM is still able to track the state of the object.

## Starting, Stopping, and Canceling from CSM Panels

You can issue start, stop, and cancel commands for an object from the Global Overview panel when you want to change the state of the object outside of its regular schedule.

You can use the following line commands to manage objects from the panel:

- **(S)tart/sto(p)**
- **(C)ancel**
- **b(O)unce** (where an object is stopped and immediately restarted)
- **rese(T)** (where an object is returned to its defined schedule)
- **(M)ove**
- **e(X)ec** (where an EXEC that is associated with a CSM ALERT is scheduled)

Refer to “Performing Routine Object Tasks” on page 2-110, which contains information about how to issue these commands from CSM Operations mode.

## Managing Objects across BBI-SS PASs

When cross-system object management is enabled, CSM enables you to monitor and manage objects that are managed by CSM on another BBI-SS PAS. Cross-system management can be performed from several CSM panels.

The Global Overview panel enables you to issue commands against any CSM object running on a CSM enabled subsystem through either the default monitoring mode or the exception mode.

If you want see the objects for only a particular group, you can select that specific group. On the Object Group panel, use the LIST line command to show the objects of a specific group.

While viewing CSM objects on a remote BBI-SS PAS, you can enter the following line commands to remotely manage objects on the other BBI-SS PAS:

- **(S)tart/sto(p)**
- **(C)ancel**

- **b(O)unce** (where an object is stopped and immediately restarted)
- **rese(T)** (where an object is returned to its defined schedule)
- **(M)ove**
- **e(X)ec** (where an EXEC that is associated with a CSM ALERT is scheduled)

To enable and perform cross-system management, refer to “Enabling Cross-System Object Management” on page 2-108.

### Starting, Stopping, and Canceling with the CSMACT EXEC

You can use the BMC Software distributed EXEC CSMACT to start and stop objects outside of the schedule. You might use the CSMACT EXEC when

- changing the state of an object from the OS/390 console
- changing the state of an object from an EXEC
- changing the state of an object from any MAINVIEW AutoOPERATOR panel

For more information about using CSMACT, refer to “Using the CSMACT EXEC to Modify Object States Manually” on page 2-135.

## Examples of CSM Processing

The following sections provide examples of CSM processing and apply the terms and concepts discussed up to this point:

- “How CSM Automates the IPL Process” on page 2-22

This section describes how CSM brings up objects during the IPL process with a sample IPL scenario.

- “How CSM Automates the Shutdown Process” on page 2-25

This section describes how CSM shuts down the system in an orderly, controlled way with a sample shutdown scenario.

- “How CSM Manages Routine Object Scheduling” on page 2-29

This section describes how CSM brings objects UP or DOWN according to their schedules.

- “How CSM Handles Global Calendar Overrides” on page 2-30

This section describes how CSM uses Global Calendar Overrides to either add or extend an object’s UP or DOWN time.

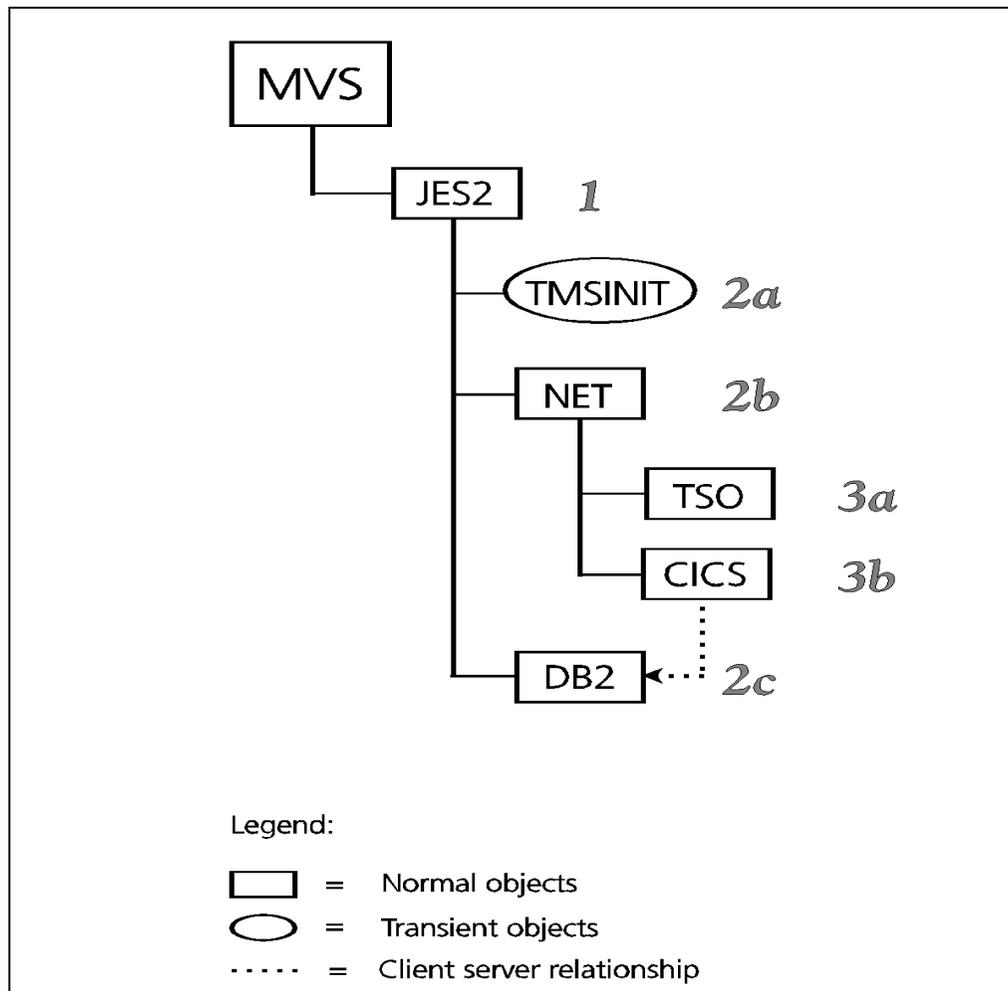
## How CSM Automates the IPL Process

From “When CSM Starts” on page 2-16, you know that when CSM initializes after the BBI-SS PAS, the first thing CSM does is read the database where all the object definitions, schedules, and relationships are stored. CSM issues the start commands for all the objects that are scheduled to be brought UP and schedules time-initiated EXECs for the objects that are scheduled to be brought UP or DOWN at a later time.

This section provides an example of the order in which objects are brought UP when CSM initializes.

The basic principle during CSM initialization is that CSM always attempts to start all the direct children of an object automatically after the parent object is UP. In parent-child relationships, children objects cannot operate without the parent object. When CSM initializes after an IPL, none of the objects’ actual states are UP (except OS/390) so CSM tries to start all the objects that are defined in its database.

Figure 2-2 Example of IPL



**Note:** The numbers next to the objects in this figure denote the order in which CSM attempts to start the objects.

### During IPL

Suppose the simplified system that CSM is performing an IPL on is shown in Figure 2-2 and all of these objects are scheduled to be UP when CSM initializes. In this figure, OS/390 is the first object UP and the parent (either directly or indirectly) of all the other objects in the system.

As IPL continues, CSM checks to see if JES2 (the only direct child of OS/390) is UP. Finding JES2 to be DOWN, CSM attempts to bring JES2 UP and issues the start commands specified for the JES2 object. When CSM receives the start message indicating that JES2 is UP, the Start Rule fires and calls the CSMUP EXEC, which changes the actual state of JES2 to UP. Starting of the JES2 object is now complete.

Then, CSM attempts to start the three direct children of JES2: TMSINIT, NET, and DB2. CSM uses the same process to start these objects as it did for JES2.

When CSM recognizes that NET is UP, CSM attempts to start the two direct children of NET: TSO and CICS. When CICS is UP, CSM connects CICS and DB2 in a client-server relationship.

**Note:** In the event that a parent object fails to start, the children are not started. In this example, if NET failed to start, CSM would not attempt to start its children objects.

### After IPL is Complete

Once the IPL process is complete, CSM is constantly aware of the state of all the objects so that it continually tries to match the actual state of the objects to the desired state (which is dictated by the object's schedule). CSM takes actions when

- an object's schedule dictates that it is supposed to be brought DOWN at a specific time and then brought back UP at a later time

For example, CICS might have a scheduled down time every other Wednesday from 12:00 P.M. to 1:00 P.M. CSM automatically performs scheduled processes at the specified times (see "How CSM Manages Routine Object Scheduling" on page 2-29).

- an object fails unexpectedly

If an object fails or abends, CSM attempts object recovery by issuing the start command for the failed object (see "Recovering Failed Objects" on page 2-17).

- an operator requests an action

For example, from the CSM panels or from the CSMACT EXEC, you might start, stop, cancel, or bounce an object.

- an Administration task is performed

For example, if you have access to Administration functions in CSM, you might create or change a schedule for an object. Or you might issue a Global Calendar Override for an object.

## Benefits

Having CSM manage the IPL process greatly reduces the chances of operator errors. You also can save time because CSM automatically starts objects immediately after CSM recognizes that their parent objects are UP. Another benefit is that CSM notifies you automatically with an ALERT if an object fails to start.

## How CSM Automates the Shutdown Process

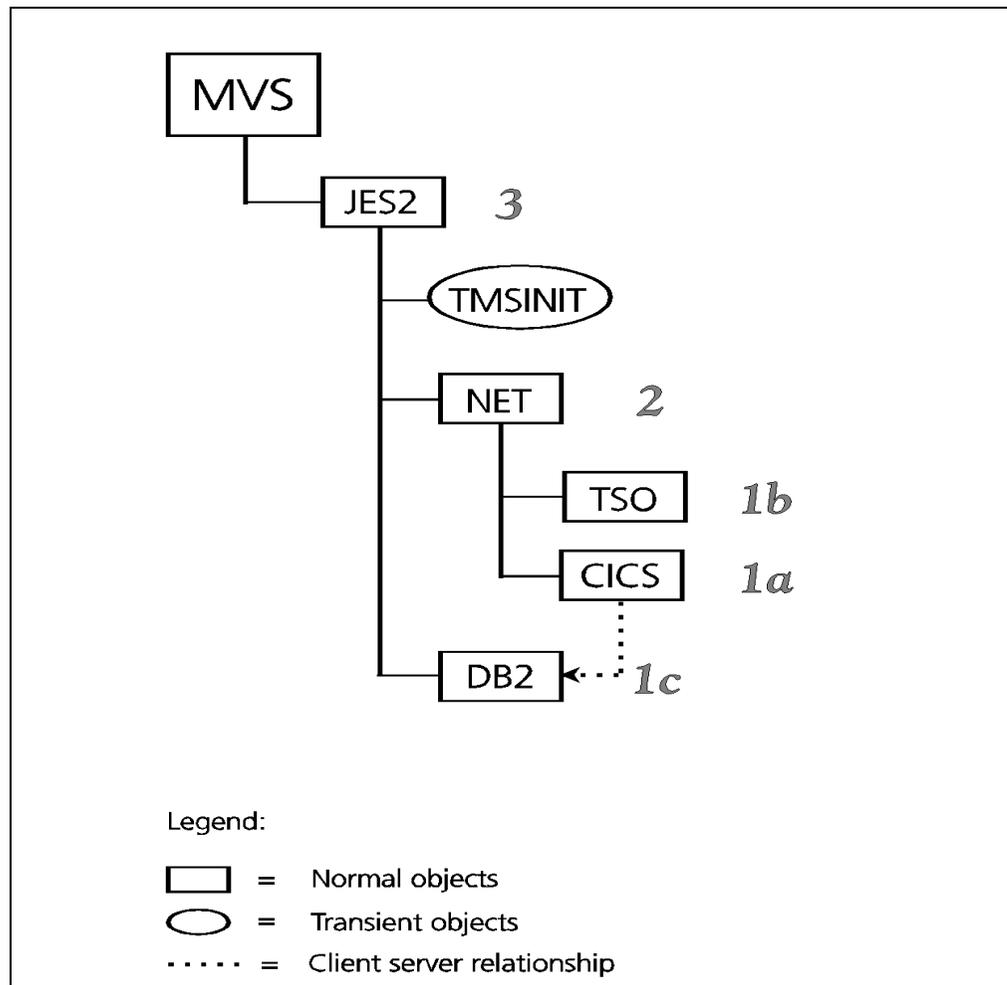
Using CSM for system shutdowns enables you to perform the shutdown in an orderly manner and by issuing only a single command **SHUTSYS**.

Just as in the IPL of a system where CSM brings parent objects UP before bringing any children objects UP, during shutdown, CSM stops children objects first before stopping parent objects. This chain of events is initiated by simply issuing the **SHUTSYS** command for the group of objects that you want to stop.

## SHUTSYS

From the COMMAND line on the CSM panels, type **SHUTSYS** *group name*. You are prompted for confirmation. By specifying **Yes**, CSM begins the shutdown of the entire system beginning with the objects furthest away from the OS/390 parent.

Figure 2-3 Example of Shutdown



**Note:** The numbers next to the objects in this figure denote the order in which CSM attempts to stop the objects.

### During Shutdown

As shown in Figure 2-3, when the **SHUTSYS** command is issued against a group of objects, CSM finds the child that is both furthest away from OS/390 and has the most dependencies to operate. CSM proceeds to stop the children objects of NET (TSO and CICS) and the child object of JES2 (DB2).

CSM issues the stop commands for these objects and receives the stop messages. The Stop Rules fire and the CSMDOWN EXEC changes the actual state of each of the objects to DOWN.

The same process occurs for the next object stopped, which is NET. Then JES2 is stopped.

Note that TMSINIT is omitted during this process because TMSINIT is a transient object and CSM only starts transient objects: CSM does not issue stop commands for transient objects.

TMSINIT might have already stopped when the shutdown began. If not, when CSM completes the shutdown, TMSINIT might still be UP and you will have to stop it manually.

### **After Shutdown**

After shutdown, all the objects have a desired state of DOWNFORCE. If any objects in the system were not defined to CSM, they might still be running and you must shut them down manually.

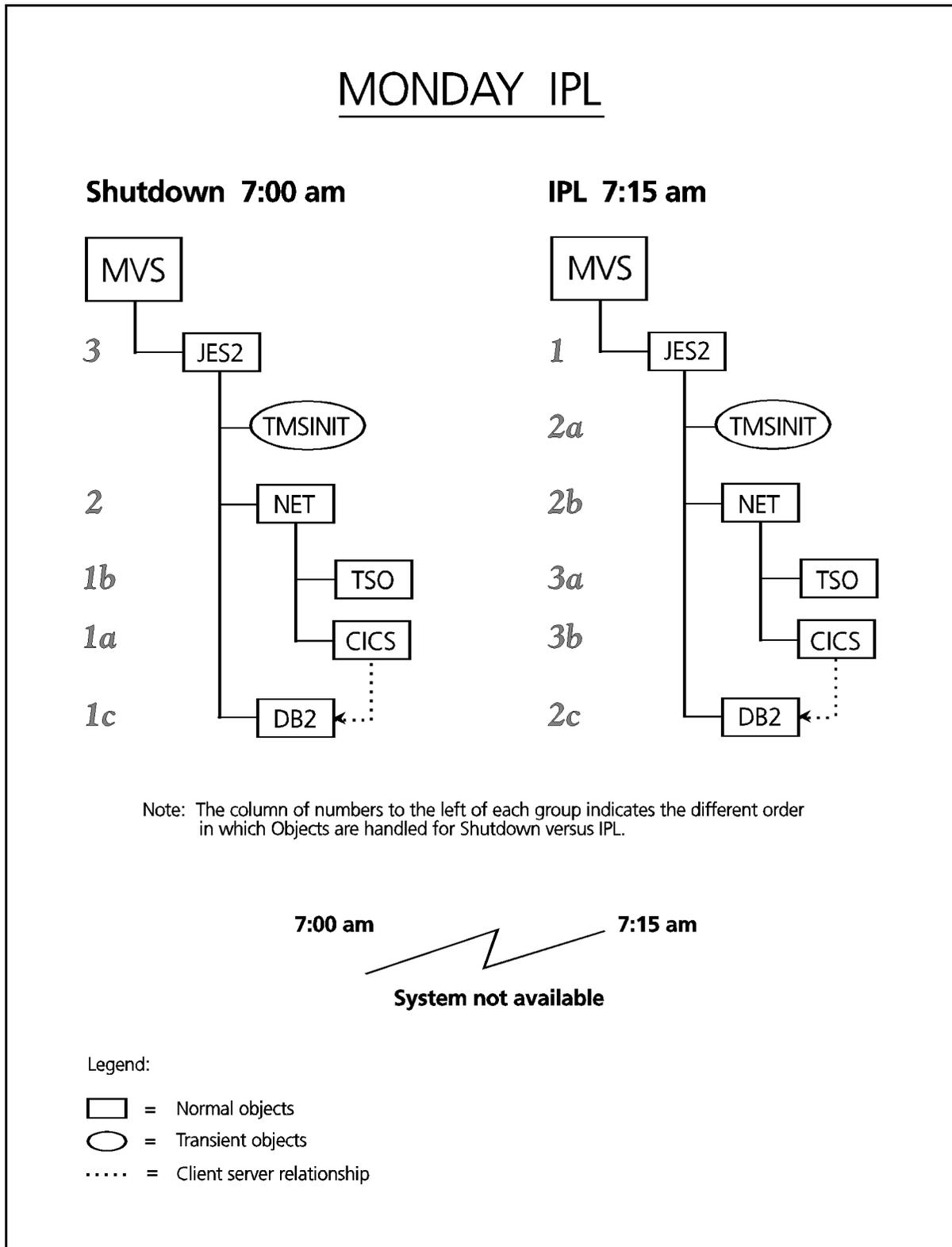
### **Benefits**

The greatest benefit is that you do not have to manually shut down all the objects, which can be time-consuming and tedious. Also, CSM already knows which objects to stop first. Using CSM to shut down the system means that time and effort are saved.

### **Shutdown and IPL**

Figure 2-4 on page 2-28 shows a shutdown, a 15 minute system unavailability (note that the shutdown itself might take as much as 15 minutes), and an IPL. The numbers next to the objects denote the order in which CSM brings objects DOWN during shutdown and UP during IPL.

Figure 2-4 Example of Shutdown and IPL



## How CSM Manages Routine Object Scheduling

As described in “Routine Scheduling” on page 2-17, during IPL, CSM schedules time-initiated EXECs to start and stop objects.

### Scheduling to Start Objects

For example, suppose IPL occurs at 9:00 A.M. and an object is scheduled to be started at 10:00 A.M. Prior to 10:00 A.M., the desired and actual states are DOWN. At 10:00 A.M., a time-initiated EXEC sets the desired state to UP and CSM detects that the object’s actual state is still DOWN.

CSM attempts to make the two states match by automatically issuing the start command for the object. The object’s start message is received, the Start Rule is fired, and the CSMUP EXEC executes and changes the actual state to UP. Now the actual and desired state match (UP).

### Scheduling to Stop Objects

The scheduled stopping of an object happens much the same way as starting an object.

At 10:00 P.M., the time-initiated EXEC changes the desired state to DOWN, which does not match the object’s actual state of UP so CSM issues the stop commands. The object’s stop messages are received, the Stop Rule is fired, and the CSMDOWN EXEC executes and changes the actual state to DOWN. Now, the actual and desired state match (DOWN).

### Example

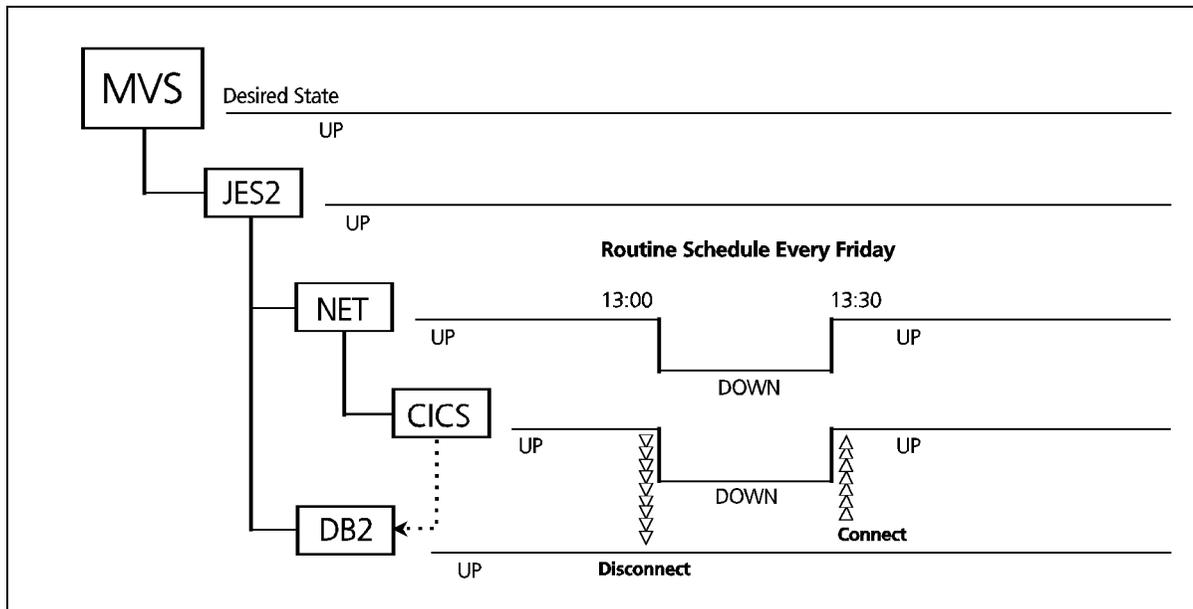
Figure 2-5 on page 2-30 shows a scheduled down time for the object NET and illustrates what happens to the dependents when NET is scheduled for DOWN time every Friday from 1:00 P.M. to 1:30 P.M. NET has a child object of CICS. DB2 is a server to client CICS.

CSM stops the dependents of an object first; in this case, CICS. First, CSM disconnects DB2 from CICS with the disconnect command that was specified when DB2 was defined as a server to CICS. DB2 does *not have to come down* because it does not rely on CICS to function.

With DB2 disconnected, CSM stops CICS and, when CICS is completely DOWN, CSM stops NET. Note that the shutdown of CICS and NET might each take a few minutes so they might not actually be DOWN until 1:10 P.M.

At 1:30 P.M., CSM starts the parent object NET and, when NET is completely UP, starts child object CICS. After CICS is UP, CSM reconnects CICS to DB2. This process will occur every Friday.

**Figure 2-5 Example of Scheduled Down Time**



## How CSM Handles Global Calendar Overrides

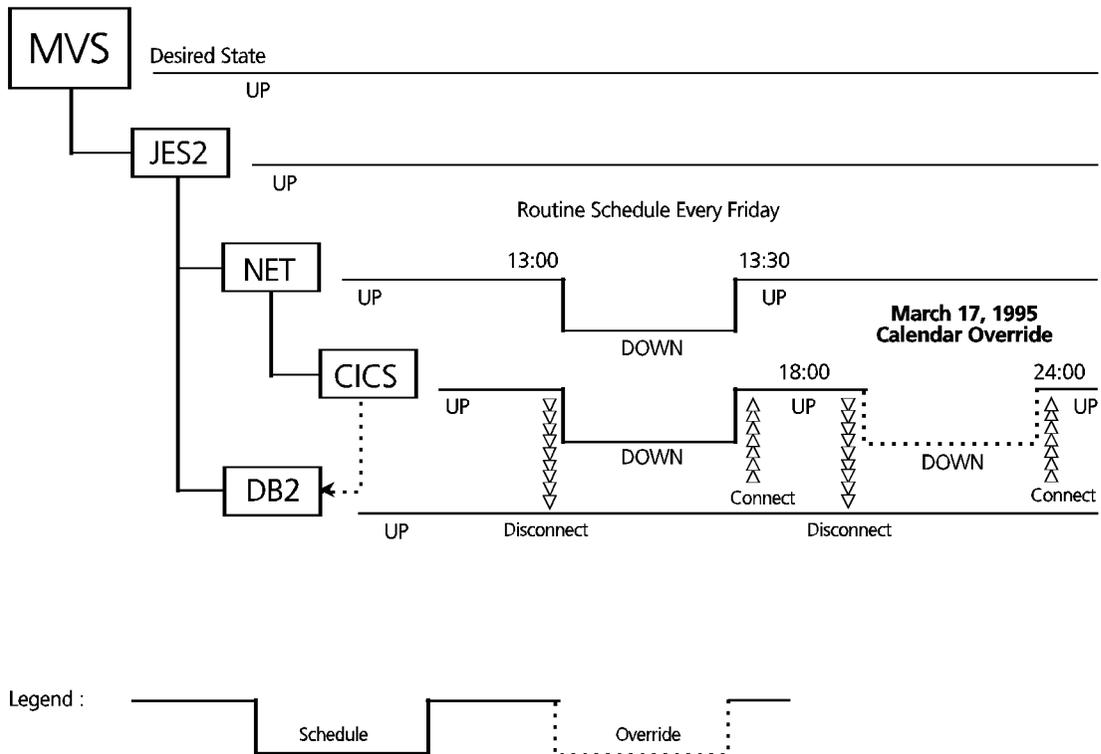
Occasionally, you might have to schedule an override to an object's routine schedule. There are two types of overrides:

- One type is where you add or delete UP or DOWN times from an object's routine schedule.
- Another type is where you extend the object's routine scheduled UP or DOWN times.

### Example 1

Figure 2-6 on page 2-31 shows a Global Calendar Override where an additional down time is added to the routine schedule of object CICS. CICS has a routine schedule to be DOWN on Fridays from 1:00 P.M. to 1:30 P.M. when its parent, NET, is DOWN (because the child object cannot operate without the parent object).

Figure 2-6 Example of Global Calendar Override: Add Down Time



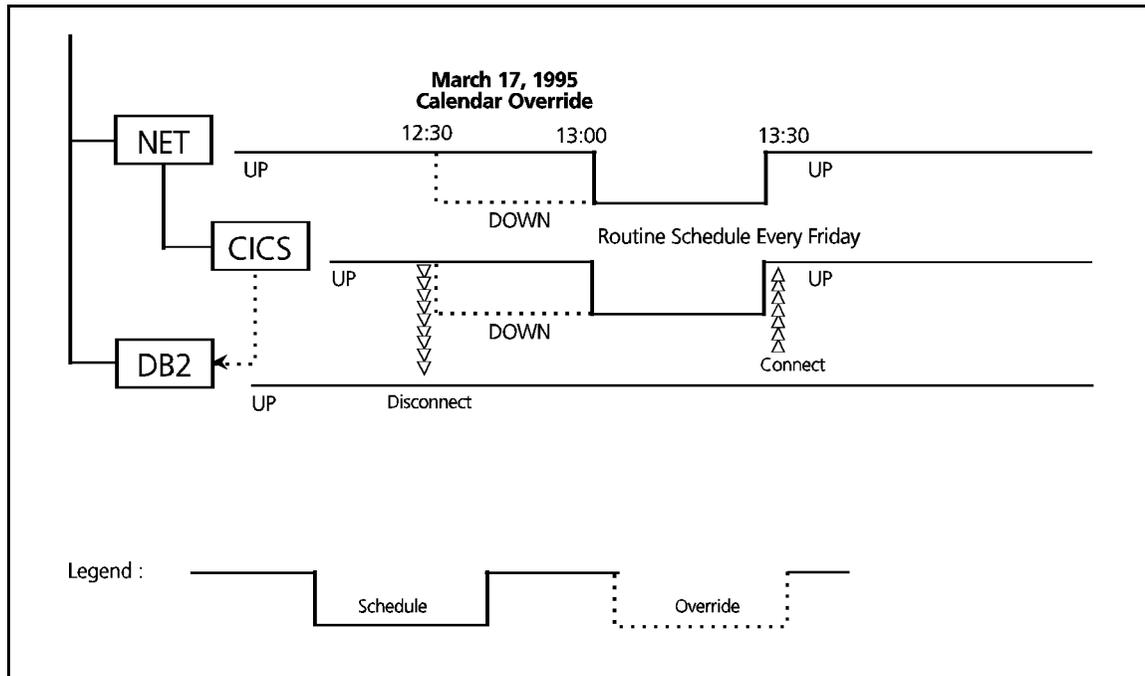
This example also shows CICS has a Global Calendar Override scheduled that brings CICS DOWN from 6:00 P.M. to midnight in addition to its routine down time. At 6:00 P.M., CSM disconnects CICS from DB2, brings CICS DOWN, and reconnects to DB2 when CICS comes UP at midnight. Notice that the server, DB2, can stay UP through both of CICS' scheduled down times.

All Global Calendar Overrides are valid only once. Once an override is complete, CSM returns to routine schedules.

## Example 2

Figure 2-7 on page 2-32 shows an instance where a Global Calendar Override is scheduled to bring NET DOWN an hour earlier than the routine schedule specifies.

**Figure 2-7 Example of Global Calendar Override: Extend Down Time**



CSM does not explicitly check to see if the CICS object’s schedule conflicts with its parent for an override. In this example, even though the override is for the object NET, the override also applies to CICS because CICS is a child of NET.

When CSM brings DOWN NET and CICS an hour earlier than the routine scheduled time, DB2 is disconnected from CICS an hour earlier. When NET is started, CICS is also started and reconnected to DB2.

## How to Effectively Plan for CSM Implementation

Effective planning, before you ever define the first object, is the most important part of getting CSM to manage your system. The elements you must plan in advance are

- identifying objects and their relationships

You must identify all the objects and how they are related or dependent on one another because CSM uses these dependencies to manage the system in a controlled way.

- mapping out periods of object inactivity

All the objects need to have their DOWN periods identified and then scheduled. Remember, all objects that have parents must include the DOWN time for the parents with their own DOWN time.

- listing start, stop, cancel, and (dis)connect commands

You should identify all the start, stop, and cancel commands that will be used by CSM to start, stop, and cancel all the objects. For client-server relationships, you should list the connect and disconnect commands that CSM will use to establish those relationships.

- choosing Start and Stop text-IDs

The text-IDs actually notify CSM of an object's change in status.

- choosing text-IDs for objects where you want to define an abnormal termination event

As part of an object's definition, you can define that an abnormal termination event occurs and a MAINVIEW AutoOPERATOR ALERT is issued if the object terminates unexpectedly (abends). In addition, CSM can automatically issue an EXEC or command, but first you must determine the text-ID of the message that is issued when the object abends.

This section discusses each of these tasks and points out how to avoid problems.

**Note:** The best source of information for developing your object definitions is the operators who regularly IPL and shut down the system. Another good source of information is the systems programming group. Both of these groups should have a good idea about which objects should be started or stopped in the system.

Many data processing installations also have documentation that describes the Started Task structure in their systems. Operations training materials might also cover this subject.

## Determining Object Relationships

The first step in defining a system to CSM is to list all the objects running in that system and determine the relationships the objects have to one another. You might find it helpful to create a chart similar to Figure 2-1 on page 2-7.

**Guidelines:** Put OS/390 at the top (OS/390 collectively refers to all those tasks automatically started by the operating system at IPL time).

Follow OS/390 with the objects as they are started manually by Operations during an IPL. Pay special attention to points in the process where the startup of one object is the prerequisite for the startup of one or more other objects. This type of relationship is a direct parent-child relationship and you should note these in the diagram.

**Parent-Child relationships:** A good example of a direct parent-child relationship is the group of objects that usually start after the VTAM network starts. In this case, the VTAM network Started Task is the parent and the group of objects that can be started immediately after the VTAM are the children.

**Client-Server relationships:** Another type of relationship to note is when an object acquires services from other objects: this type of relationship is a client-server relationship. An example of a client-server relationship is one between a CICS terminal-owning region and a CICS application-owning region.

Both regions can run on their own and have functions that can be performed without the other. The terminal-owning region (client), however, requires the services of the application-owning region (server) to access CICS applications.

Noting the direct parent-child and client-server relationships both simplifies and ensures the accuracy of the definition of these objects in the database.

## Planning Schedules

Objects defined to CSM that do not specify a schedule are given a default schedule of UP 24 hours a day, 7 days a week. Most objects fall into this category.

Some objects, usually online inquiry tasks, require periods of inactivity for system maintenance and batch file processing. CSM can control the schedule for these objects and stop the object (change its actual state to DOWN) and then restart the object after a specified period of time.

You need to determine the DOWN time and the DOWN duration for all the objects and then input these times to CSM as part of the object's definition.

**Hint**

The best place to gather object scheduling information is from the Operations staff or systems programming group.

If the object you are defining a schedule for is the child of an object other than OS/390, you must take the scheduling of the parent into consideration to avoid conflicts. A child object can be UP only when its parent is UP.

Remember, an object that does not have a schedule always has a desired state of UP unless it is altered by a schedule, a Global Calendar Override, or a CSMACT EXEC. Any object that has a schedule, at some point in time, will have desired states that are not UP.

Refer to "Modifying the Object's Schedule" on page 2-96 for information about defining a schedule for an object that previously had no schedule defined. Refer to "Dealing with Scheduling Conflicts at Modification" on page 2-104 for information about scheduling conflicts, how to avoid them, and how to deal with them, if they occur.

## Listing Start, Stop, Cancel and (Dis)Connect Commands

You must include the OS/390 commands that CSM uses to start, stop, and cancel the object as part of an object's definition. If the object has a server, you also need to include the commands issued to connect (and disconnect) to (and from) the server.

To obtain this information, you might consider consulting the operations or systems programming groups at your site. You also can check the manual provided for the object by its vendor. Many times, a manual published by a vendor has a command, implementation, installation, or customization chapter in which you can find the commands.

**Note:** Not having correct commands defined to manage an object could cause command time-outs and the object might not start or stop properly. These objects will remain in the intermediate states of STARTING or STOPPING.

## Choosing Start and Stop Text-IDs

As described in “How CSM Recognizes when an Object Starts or Stops” on page 2-14, the text-IDs that are issued when an object has started or stopped are an integral part of an object’s definition. CSM uses these text-IDs to keep track of the objects.

Therefore, if the wrong message is defined for the object, CSM will not be able to determine the actual state of the object. This situation could lead to objects being left in an actual state of STARTING or STOPPING and their states will not be recognized by CSM as UP or DOWN.

The correct Start and Stop text-IDs for an object can be determined by looking at a sampling of message traffic for an object. Find this message traffic in the SYSLOG or JES message log file of the object’s output listing.

You can specify up to five Start and Stop text-IDs when defining an object.

<b>Hint</b>
If the object you are gathering information for starts before JES2, look in the SYSLOG for its start and stop messages.

## Choosing Text-IDs for Abnormal Termination Events

As part of an object’s definition, you can define that a MAINVIEW AutoOPERATOR ALERT is issued if the object terminates unexpectedly (abends). In addition, CSM can automatically issue an EXEC or command, but first you must determine the text-ID of the message that is issued when the object abends.

One place you should look for abnormal termination text-IDs is the job log of an abended object. Note that an object might issue several messages when it abends.

You can specify up to five text-IDs for abnormal terminations when defining an object. Abnormal termination events can also specify a particular text string to look for in a WORD variable (used by the Rules Processor) such as ABEND=S0C7 that will enable greater control over which abnormal termination event is handled and which is not. You must, however, specify the number of the WORD variable that they are to access (for example, specify 3 for WORD3, and 7 for WORD7 and so on).

## Before You Begin

The following sections describe other aspects of CSM that you should be familiar with before you begin using the application.

### Setting Up the Shared Repository

CSM objects, once they are defined, reside in the shared repository data set. This shared repository data set must be allocated before CSM can be used by any system in a CSM-PLEX. Once it is allocated, every BBI-SS PAS that is part of the CSM-PLEX must have the data set name of the shared repository that is specified in the BBIVARxx member used by the BBI-SS PAS.

Once the shared repository is allocated, objects can be used across all the BBI-SS PASs in the CSM-PLEX. Refer to “Cross-System Object Management: Using Group” for more information.

For information about the implementation steps that must be taken before you can bring up CSM, refer to the chapter “Implementing Continuous State Manager” in the *MAINVIEW AutoOPERATOR Customization Guide*.

### Cross-System Object Management: Using Group

Once the shared repository is designated in each of the BBIVARxx members in the CSM-PLEX, an object must be defined only one time. The object and its schedule can be used by any system in the CSM-PLEX by associating the object with one or more groups.

A group is a collection of objects designated to run on a particular CSM partner. When a CSM object is defined, you must identify which group the object is a member of. By default, the object is associated with the active group of the local CSM. Cross-system management enables you to associate an object with up to 20 groups.

In addition, by default each CSM system on running its own BBI-SS PAS uses the object group whose name matches the BBI-SS PAS subsystem ID. To override this default, specify the CSMALTDDB parameter in the OS/390 START command for the BBI-SS PAS.

You can also use the line commands on the Object Groups panel to manually enable another group and disable the existing group.

## Serializing the CSM Shared Repository

CSM shared repository serialization is provided to make sure multiple users cannot update individual records in the repository data set at the same time. Serialization occurs on a record-by-record basis so that more than one person can make updates at one time, but one user cannot update a record at the same time that another user is updating it.

Object entries are serialized on an object by object (or group by group) basis. If you have security access to edit objects, you issue the **EDIT** line command and CSM tries to get an exclusive enqueue. If it can get it, you can edit the object. The enqueue is held until you issue the **SAVE** command or the **CANCEL** command. No one else can edit that object. If the resource is being held exclusively, you receive a message indicating that another user is presently accessing the repository and the request is changed to browse.

As far as groups are concerned, you issue the **EDIT** line command. If CSM can get the exclusive enqueue for the group record, you can edit that group. When you are finished editing the group definition and press **END**, the group definition is immediately written to the repository and the enqueue is released. If another user tries to edit the group while you are doing so, they receive the appropriate message and their request is changed to browse.

## Securing Access to CSM

You can secure who has access to CSM and its various functions. For more information, refer to *Implementing Security for MAINVIEW Products*.

## Backing Up the CSM Repository

CSM does not create backups of the shared repository automatically. You can run a simple job stream that allocates a backup repository data set and copy the current one into the backup one. The JCL might look like Figure 2-8:

**Figure 2-8 Sample JCL to Back Up CSM Shared Repository**

---

```
//BAOKMZR JOB (3911), 'ZIOLKOWSKI',
//          CLASS=K,
//          MSGCLASS=R,
//          NOTIFY=&SYSUID
//BUILD EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
DEL (BAOKMZ.CSM.BACKUP)
SET MAXCC=0
DEF CL( NAME(BAOKMZ.CSM.BACKUP) OWNER(BAOKMZ) KEYS(128
0) -
RECORDSIZE(14237 14237) ) -
DATA( NAME(BAOKMZ.CSM.BACKUP.DATA) SPEED -
CYL(20,52) VOL(BAB325) SHAREOPTIONS(4 )) -
INDEX( NAME(BAOKMZ.CSM.BACKUP.INDEX) CISZ(14237) -
TRK(120,1) VOL(BAB325) SHAREOPTIONS(4 ))
/*
//COPY EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//F1      DD DSN=BAOKMZ.CSM.REPOSIT,DISP=SHR
//F2      DD DSN=BAOKMZ.CSM.BACKUP,DISP=SHR
//SYSIN   DD *
REPRO IFILE(F1) OFILE(F2)
/*
```

---

The repository is a keyed VSAM file. Any backup or restoration application that handles VSAM files can do the job. The repository can be copied while CSM is running and using the repository to be copied. You might want to consider using MAINVIEW AutoOPERATOR to schedule the backup job at a particular time each day.

## Using CSM Panels

The CSM panels that enable you to define objects and their schedules can all be accessed in a ring from the Object Detail Control panel. To proceed through the ring one panel at a time, press the **Enter** key. The panels in the ring are

- Object Detail Control
- Command Specification
- Conditional Commands
- Event Specification 1
- Event Specification 2
- User Notification
- Object Groups
- Object Prerequisites
- Object Scheduling
- Client/Server Relationships

Figure 2-9 on page 2-41 shows an example of the Object Detail Control panel.

**Figure 2-9 Object Detail Control Panel**

```

BMC Software ----- Object Detail Control - Add ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     TGT --- KMZ1

The following options are displayed in sequence, or may
be selected by entering the three-character code

    CMD - Command Specifications           EV1 - Event Specifications 1
    EV2 - Event Specifications 2         NOT - User Notification
    GRP - Object Groups                  REQ - Object Prerequisites
    CLS - Client/Server Relationships     SCH - Object Scheduling

Object Name           ==> AAOCSM01       Member of Group      ==> KMZ1____
Object Type           ==> NORM____       Type of object ( ? for list)

Verify Force Down    ==> N (Y/N)         Restart Only Control ==> N (Y/N)

Application information:
Author               ==> BAOKMZ1
Description          ==> Child of MVS_____

Last Modified by BAOKMZ2 on 01/30/2001 at 15:43

Enter END command to process and return, ENTER to continue, or CANCEL

```

As noted on the panel, CSM enables you to display any of the panels in the ring by using shortcut commands entered on the COMMAND line. The commands are listed in Table 2-1.

**Table 2-1 Shortcut Commands for Navigating CSM Panels**

Command	Displays This Panel
DET	Detail Control
CMD	Command Specification
EV1	Event Specification 1
EV2	Event Specification 2
NOT	User Notification
GRP	Object Groups
REQ	Object Prerequisites
SCH	Object Scheduling
CLS	Client/Server Relationships

**Note:** Pressing **END** from the Detail Control Panel processes any changes made to the object definition and returns you to Global Overview; pressing **END** from any other panel in the ring returns you to the Detail Control Panel.

## Accessing CSM for the First Time

After the CSM shared repository is defined as part of implementing and customizing CSM (as described in the chapter “Implementing Continuous State Manager” in the *MAINVIEW AutoOPERATOR Customization Guide*) you can access CSM from either an ISPF-based or a MAINVIEW Alternate Access terminal session. CSM initialization begins after MAINVIEW AutoOPERATOR initialization has completed. At this time, CSM will run in minimum mode where no objects are monitored (managed by CSM).

If you are a new user and do not have any CSM objects created from previous releases of MAINVIEW AutoOPERATOR, the first thing you will do is create a group and then define objects.

If you are migrating CSM objects and definitions from a previous release of MAINVIEW AutoOPERATOR, such as MAINVIEW AutoOPERATOR 4.1 or MAINVIEW AutoOPERATOR 5.1, you should run the CSM conversion utility as described in the *MAINVIEW AutoOPERATOR Customization Guide*.

To access the CSM application, select Option 3, Continuous State Manager, from the Automation Menu (Figure 2-10).

**Figure 2-10 MAINVIEW AutoOPERATOR Automation Menu**

```

BMC Software ----- AUTOMATION MENU ----- MAINVIEW AutoOPERATOR
OPTION ==>                                     DATE -- 00/08/17
                                                TIME -- 11:36:58

Basic Automation:
  1 Event Activity Statistics
  2 Display/Modify Rules and Rule Sets
  3 Continuous State Manager - Global Overview

Advanced Automation:
  6 Shared Object Facility
  7 Display/Modify EXEC Status
  8 Time-Initiated EXEC Requests
  9 Open Systems Procedural Interface (OSPI)

                                                PF1/13 HELP PF3/15: EXIT

Copyright 1999-2002, BMC Software, Inc. All rights reserved.
    
```

The Global Overview panel is displayed (Figure 2-11 on page 2-43).

**Figure 2-11 Global Overview Panel**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     SCROLL ==> CSR
                                                TGT --- KMZ1
Primary Commands: EXCeption  Group    SORT    SAVE    ADD    PROFile
                  CALEndar   CANcel  UNsort  CONVERT BUILD
                  Locate    CMDSHOW Xref    SAVERULE SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)  depe(N)dents  (A)dd
                  sto(P)   (EX)ec   (E)dit   c(L)ients    (D)delete
                  (CAN)cel (M)ove   (R)epeat (B)rowse    s(Y)nc

LC  Object      Group    System    Actual    Desired    Ind    Type
___  _____  _____  _____  _____  _____  _____  _____
___  MVS
___  AAOCSM01  KMZ1     SJSD      UP         UP         _____  NORM
___  AAOCSM01  KMZ5     _____  INACTIVE  INACTIVE  _____  NORM
___  AAOCSM01  KMZ6     _____  INACTIVE  INACTIVE  _____  NORM
___  AAOCSM02  KMZ1     SJSD      COMPLETE  COMPLETE  _____  TRAN
___  AAOCSM02  KMZ2     _____  INACTIVE  INACTIVE  _____  TRAN
___  AAOCSM02  KMZ5     _____  INACTIVE  INACTIVE  _____  TRAN
___  AAOCSM02  KMZ6     _____  INACTIVE  INACTIVE  _____  TRAN
___  AAOCSM08  KMZ1     SJSD      UP         UP         _____  NORM
___  AAOCSM08  KMZ2     _____  INACTIVE  INACTIVE  _____  NORM
___  AAOCSM08  KMZ5     _____  INACTIVE  INACTIVE  _____  NORM
___  AAOCSM08  KMZ6     _____  INACTIVE  INACTIVE  _____  NORM
    
```

For more information about the Global Overview panel, refer to “What the CSM Global Overview Panel Is” on page 2-44.

The first time you access the Global Overview panel, the only object that appears is the OS/390 object. The example in Figure 2-11 shows what the panel might look like after groups and objects have been defined.

Therefore, the first task you will want to complete is to create a group. Once a group is created, you might begin defining objects. For information about creating a group, refer to “Creating a Group” on page 2-51.

## How to Qualify (Mask) Panels in CSM

Throughout the panels in CSM, you will see column headings that are underlined, for example, in the Global Overview panel:

```

LC  Object      Group-Id  System    Actual    Desired    Ind    Type
___  _____  _____  _____  _____  _____  _____  _____
    
```

The underlined areas under the column headings are input masking fields where you can enter a text string that qualifies (or masks) the data that is displayed to show specific information. For example, if you enter a specific SSID in the **System** column and press **Enter**, the panel shows only those objects running on that subsystem.

Use these fields to mask any of the data in any of the CSM panels where the column headings have input areas. You can also mask in more than one field.

In the input areas, you also can use an asterisk (\*) as a wildcard character.

## What the CSM Global Overview Panel Is

The first time you access the Global Overview panel in CSM, the only object that appears is the OS/390 object. The example in Figure 2-11 on page 2-43 shows what the panel might look like after groups and objects have been defined.

Use this panel to monitor and manage all CSM objects and their various states, regardless of where the objects are executing. This panel shows

- the object name
- the name of the group with which the object is associated
- the OS/390 SYSNAME where the object and group are executing
- the actual and desired state of each object and what type each object is (for example, a normal object is NORM)
- whether the object has a pending ALERT and, if so, does that ALERT have a follow-up EXEC associated with it

You can administer (add, modify, delete, and so forth) or manage (start, stop, cancel, and so forth) any object shown on either the monitoring or the exception overview panels (refer to Figure 2-13 on page 2-46).

## Viewing an Object's Documentation

To view additional information about each object, scroll left by pressing **PF10/PF22** (see Figure 2-12).

**Figure 2-12 Global Overview Panel: Scrolled LEFT**

```

BMC Software ----- Global Overview - Desc/Stats ----- MAINVIEW AutoOPERATOR
COMMAND ==>
                                     Scroll ==> CSR
                                     TGT --- KMZ1
Primary Commands: EXCeption  Group    SORT    SAVE    ADD    PROFile
                  CALEndar   CANCEL  UNsort  CONVERT BUILD
                  Locate    CMDSHOW Xref    SAVERULE SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)  depe(N)dents  (A)dd
                  sto(P)   (EX)ec   (E)dit   c(L)ients    (D)elete
                  (CAN)cel (M)ove  (R)epeat (B)rowse    s(Y)nc

LC  Object      Description                                     Id      Date      Time
---  ---
___  AAOCSM01    Child of MVS                                     BAKMZ2  07/10/2000  15:58
___  AAOCSM01    Child of MVS                                     BAKMZ2  07/10/2000  15:58
___  AAOCSM01    Child of MVS                                     BAKMZ2  07/10/2000  15:58
___  AAOCSM02    Child of MVS, Also                             BAKMZ2  07/10/2000  09:05
___  AAOCSM02    Child of MVS, Also                             BAKMZ2  07/10/2000  09:05
___  AAOCSM02    Child of MVS, Also                             BAKMZ2  07/10/2000  09:05
___  AAOCSM02    Child of MVS, Also                             BAKMZ2  07/10/2000  09:05
___  AAOCSM08    Child of MVS Again                             BAKMZ2  06/21/2000  19:35
___  AAOCSM08    Child of MVS Again                             BAKMZ2  06/21/2000  19:35
___  AAOCSM08    Child of MVS Again                             BAKMZ2  06/21/2000  19:35
___  AAOCSM08    Child of MVS Again                             BAKMZ2  06/21/2000  19:35
    
```

This following information for each object is available for viewing:

- first thirty bytes of the object description
- user ID of the last person to modify the object
- time and date of the last modification to the object

**Note:** If an object is inactive (because the group that it belongs to is not enabled), the **Ind** column (as seen in Figure 2-11 on page 2-43) will show no information.

To return to the original panel, press **PF11/PF23**. Figure 2-11 on page 2-43 is displayed again.

## Viewing Objects Whose States Do Not Match

The Global Overview panel shows the state of all the objects. To see only the objects whose actual and desired states do not match, enter the **(EXC)eption** primary command on the **COMMAND** line to display the CSM Exceptions panel (see Figure 2-13).

To toggle back to the complete display of objects, enter the **(MON)itor** primary command on the **COMMAND** line.

**Figure 2-13 CSM Exceptions Panel**

```

BMC Software ----- CSM Exceptions ----- MAINVIEW AutoOPERATOR
COMMAND ==>>                                     SCROLL ==>> CSR
                                                    TGT --- KMZ1
Primary Commands: MONitor      Group      SORT      SAVE      ADD      PROFile
                   CALendar    CANcel    UNSort    CONVERT    Build
                   Locate      CMDSHOW  Xref      SAVERULE  SHUTSYS

LC  Object      Group      System      Actual      Desired      Ind      Type
---  _____  _____  _____  _____  _____  _____  _____
___  AAOCSM65    KMZ1      SJSD      STARTING    UP           XE      NORM
***** Bottom of data *****
    
```

The CSM Exceptions panel shows a list of those objects where

- any object Actual and Desired states do not match

In Figure 2-13, the Actual and Desired states for object AAOCSM65 do not match.

- objects have at least one outstanding ALERT

ALERTs are shown in the **Ind** column. In this figure, **XE** denotes that an ALERT is associated with this object and the ALERT has an EXEC.

When an object's Actual and Desired states do not match (as shown in the fifth and sixth columns), you can issue the following line commands against these objects to try to change their states:

- **(S)tart/sto(p)**
- **(CAN)cel**
- **b(o)unce**
- **(EX)ec**
- **(M)ove**
- **rese(T)**

You can administer (add, modify, delete, and so forth) or manage (start, stop, cancel, and so forth) any object shown on either the Global Overview or the Exceptions panels.

## Filtering Groups That Are Displayed on Global Overview

The Global Overview panel shows information for all of the objects in all of the groups in the local repository. Additionally, it shows information for objects associated with groups in remote repositories communicating with the local BBI-PAS. Information is also displayed for objects monitored by pre-Version 6 CSM applications that communicate with the local BBI-PAS.

If a large number of systems are monitored, the results can be a large amount of information displayed that might be difficult to view on the Global Overview panel. Refreshing the information might also prove to be time consuming.

Figure 2-14 shows an example of a large number of systems being monitored:

**Figure 2-14 Pre-Version 6.2 Example of the Global Overview Panel**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>>                                     SCROLL ==>> CSR
                                                    TGT --- KMZ1

Primary Commands: EXception  Group      SORT      SAVE      ADD
                  CAleNDAR  CANCEL    UNsort    CONVERT   Build
                  Locate    CMDSHOW   Xref      SAVERULE  SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)  depe(N)de(N)ts (A)dd
                  sto(P)   (EX)ec   (E)dit   c(L)ients  (D)elete
                  (CAN)cel (M)ove   (R)epet (B)rowse  s(Y)nc

LC  Object      Group      System      Actual      Desired      Ind      Type
---  ---
___  AAOCSM01    KMZ1       SJSD        UP          UP           _____  STC-NORM
___  AAOCSM01    KMZ5              INACTIVE    INACTIVE    _____  STC-NORM
___  AAOCSM01    KMZ6              INACTIVE    INACTIVE    _____  STC-NORM
___  AAOCSM02    KMZ1       SJSD        COMPLETE    COMPLETE    _____  STC-TRAN
___  AAOCSM02    KMZ2              INACTIVE    INACTIVE    _____  STC-TRAN
___  AAOCSM02    KMZ5              INACTIVE    INACTIVE    _____  STC-TRAN
___  AAOCSM02    KMZ6              INACTIVE    INACTIVE    _____  STC-TRAN
___  AAOCSM08    KMZ1       SJSD        UP          UP           _____  STC-NORM
___  AAOCSM08    KMZ2              INACTIVE    INACTIVE    _____  STC-NORM
___  AAOCSM08    KMZ5              INACTIVE    INACTIVE    _____  STC-NORM
___  AAOCSM08    KMZ6              INACTIVE    INACTIVE    _____  STC-NORM
    
```

In this example, suppose that groups KMZ5 and KMZ6 are rarely enabled. Group KMZ1 is always enabled and group KMZ2 is occasionally inactive.

Resources are wasted and you can be confused by the appearance of groups that are rarely used. You do not want to delete the groups and the mask above the **Group** column cannot be used to keep the undesired groups off of the panel.

You can select the CSM object groups that are to be included or excluded from the Global Overview panel with the (PROF)ile command. Entering the (PROF)ile command displays the Group Display Selection panel.

**Figure 2-15 Using the PROFile Command to Display the Group Display Selection - Initial Visit**

```

BMC Software ----- Group Display Selection ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     SCROLL ==> PAGE
                                                TGT --- KMZ1

Primary Commands: INCLudeall  CMDSHOW

LC CMDS ----- (E)xclude Group    (I)nclude Group

Enter END command to process and return, ENTER to continue, or CANCEL

  Group      Status      Repository  Description
-----
- KMZTST20  INCLUDED      LOCAL
- KMZTST21  INCLUDED      LOCAL
- KMZTST22  INCLUDED      LOCAL
- KMZTST23  INCLUDED      LOCAL
- KMZ1       INCLUDED      LOCAL      Main Group
- KMZ2       INCLUDED      LOCAL      Converted on: 08/17/2001 06:02
- KMZ5       INCLUDED      LOCAL      Converted on: 08/17/2001 08:55
- KMZ6       INCLUDED      LOCAL      Converted on: 08/17/2001 08:56
- KMZ69      INCLUDED      LOCAL      Converted on: 08/17/2001 09:01
- KMZ7       INCLUDED      LOCAL
- KMZ70      INCLUDED      LOCAL      Converted on: 08/17/2001 09:05
- KMZ8       INCLUDED      LOCAL
- KMZ9       INCLUDED      LOCAL
    
```

The first time you visit the Group Display Selection panel, you see that all of the groups are marked as INCLUDED. By default CSM shows all of the available groups on the Global Overview panel.

In this example, the goal is to include only groups KMZ1 and KMZ2 on the Global Overview panel. To accomplish this task, use the (I)nclude and (E)xclude Group line commands on this panel to select the groups to be shown on the Global Overview panel. As a group is included or excluded, its status changes. The names of the groups to include on the Global Overview panel are kept in the user's profile. All changes that are made on the Group Display Selection panel are good only for the TSO user ID that is making the changes. The changes made by one user do not affect other users.

Groups targeted to be displayed on the Global Overview panel have a status of **INCLUDED**. Groups that are not targeted have a status of **EXCLUDED**. The **Repository** column indicates to the user whether the group's definition resides in the local repository (**LOCAL**), in a remote repository, or is running on a CSM application older than version 6.1 (**REMOTE**).

You can choose to include all of the available groups by issuing the primary command (**INCL**)udeall.

If you decide to exclude all of the groups shown in Figure 2-15 on page 2-48 except for groups KMZ1 and KMZ2, complete the panel as shown in Figure 2-16.

**Figure 2-16 Group Display Selection Panel Showing Groups to Exclude**

```

BMC Software ----- Group Display Selection ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     SCROLL ==> PAGE
                                                TGT --- KMZ1

Primary Commands: INCLudeall  CMDSHOW

LC CMDS ----- (E)xclude Group   (I)nclude Group

Enter END command to process and return, ENTER to continue, or CANCEL

  Group      Status      Repository  Description
  -----
E  KMZTST20  INCLUDED    LOCAL
E  KMZTST21  INCLUDED    LOCAL
E  KMZTST22  INCLUDED    LOCAL
E  KMZTST23  INCLUDED    LOCAL
_  KMZ1       INCLUDED    LOCAL      Main Group
_  KMZ2       INCLUDED    LOCAL      Converted on: 08/17/2001 06:02
E  KMZ5       INCLUDED    LOCAL      Converted on: 08/17/2001 08:55
E  KMZ6       INCLUDED    LOCAL      Converted on: 08/17/2001 08:56
E  KMZ69      INCLUDED    LOCAL      Converted on: 08/17/2001 09:01
E  KMZ7       INCLUDED    LOCAL
E  KMZ70      INCLUDED    LOCAL      Converted on: 08/17/2001 09:05
E  KMZ8       INCLUDED    LOCAL
E  KMZ9       INCLUDED    LOCAL

```

Press the **Enter** key to mark the groups as **INCLUDED** or **EXCLUDED**. After the panel is refreshed, the status changes as shown in Figure 2-17 on page 2-50.

**Figure 2-17 Group Display Selection Panel Showing Change in Group Status**

```

BMC Software ----- Group Display Selection ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     SCROLL ==> PAGE
                                                TGT --- KMZ1

Primary Commands: INCLudeall  CMDSHOW

LC CMDS ----- (E)xclude Group   (I)nclude Group

Enter END command to process and return, ENTER to continue, or CANCEL

  Group      Status      Repository  Description
  -----
_ KMZTST20  EXCLUDED      LOCAL
_ KMZTST21  EXCLUDED      LOCAL
_ KMZTST22  EXCLUDED      LOCAL
_ KMZTST23  EXCLUDED      LOCAL
_ KMZ1      INCLUDED      LOCAL      Main Group
_ KMZ2      INCLUDED      LOCAL      Converted on: 08/17/2001 06:02
_ KMZ5      EXCLUDED      LOCAL      Converted on: 08/17/2001 08:55
_ KMZ6      EXCLUDED      LOCAL      Converted on: 08/17/2001 08:56
_ KMZ69     EXCLUDED      LOCAL      Converted on: 08/17/2001 09:01
_ KMZ7      EXCLUDED      LOCAL
_ KMZ70     EXCLUDED      LOCAL      Converted on: 08/17/2001 09:05
_ KMZ8      EXCLUDED      LOCAL
_ KMZ9      EXCLUDED      LOCAL
    
```

The changes to the list of groups become active when you enter the END command or press PF3/PF15. You can reject all changes made from the time you entered the Group Display Selection panel by issuing the CANCEL command.

When you are ready to apply the changes made to the list of groups to be displayed, enter END and return control to a Global Overview panel that looks like Figure 2-18:

**Figure 2-18 Filtered Global Overview Panel**

```

BMC Software ----- Global Overview - * Filtered * ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     SCROLL ==> CSR
                                                TGT --- KMZ1

Primary Commands: EXCeption  Group  SORT  SAVE  ADD  PROFile
                  CAleNDAR  CANCEL  UNsort  CONVERT  Build
                  Locate    CMDSHOW  Xref  SAVERULE  SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)  depe(N)dents (A)dd
                sto(P)  (EX)ec  (E)dit  c(L)ients  (D)elete
                (CAN)cel  (M)ove  (R)epeat  (B)rowse  s(Y)nc

LC  Object  Group  System  Actual  Desired  Ind  Type
---  -----
___ AAOCSM01  KMZ1  SJSD  UP  UP  STC-NORM
___ AAOCSM02  KMZ1  SJSD  COMPLETE  COMPLETE  STC-TRAN
___ AAOCSM02  KMZ2  INACTIVE  INACTIVE  STC-TRAN
___ AAOCSM08  KMZ1  SJSD  UP  UP  STC-NORM
___ AAOCSM08  KMZ2  INACTIVE  INACTIVE  STC-NORM
    
```

The indicator \* **Filtered** \* appears in the title line of Global Overview panels that have been filtered with the PROFILE command.

Changes can be made to the user's list of included groups as often as necessary. These changes last from one CSM-TS session to another.

## Defining CSM Objects

This section describes

- how to create a group (which is the first step required when you bring CSM up for the first time)
- how to create an object with CSM panels
- how to make routine modifications to objects once they are defined from the Global Overview panel.

Task	Pages
creating a group	2-51 through 2-54
creating a CSM object with CSM panels	2-55 through 2-78
modifying object definitions	2-79
modifying an object's schedule	2-93 through 2-107
enabling cross-system management	2-108

## Creating a Group

Every object must belong to a group. Therefore, when you bring CSM up for the first time, the first task you must complete is to create a group before you can begin defining objects. A group name can be one- to eight-characters long but it is recommended that you name the first group with the SSID of the subsystem that you bring CSM up on.

**Note:** The default action of CSM is to bring up the object group that has the same name as the SSID. To change this default behavior, refer to “Chapter 18, Implementing Continuous State Manager” in the *MAINVIEW AutoOPERATOR Customization Guide*.

To create a group for a subsystem named KMZ1, follow these steps:

- Step 1** From the Global Overview panel (Figure 2-19), enter the **Group** primary command on the **COMMAND** line.

**Figure 2-19 Global Overview Panel: Adding a Group**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     SCROLL ==> CSR
                                                TGT --- KMZ1
Primary Commands: EXCeption  Group      SORT      SAVE      ADD      PROFile
                  CALEndar   CANcel   UNsort    CONVERT   Build
                  Locate     CMDSHOW Xref      SAVERULE  SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)   depe(N)dents (A)dd
                sto(P)   (EX)ec   (E)dit    c(L)ients  (D)elete
                (CAN)cel (M)ove  (R)epeat  (B)rowse   s(Y)nc

LC  Object  Group  System  Actual  Desired  Ind  Type
---  -----  -----  -----  -----  -----  ---  -----
MVS  KMZ1   SJS   UP      UP      NORM

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

The Object Groups panel is displayed (Figure 2-20).

**Figure 2-20 Object Groups Panel**

```

BMC Software ----- Object Groups ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     SCROLL ==> CSR
                                                TGT --- KMZ1
Primary Commands: ADD      Locate  CMDSHOW  BUILD

LC CMDS ----- (BU)ild  (S)ync  (E)dit  E(n)able  (D)isable  (O)ffload
                (A)dd   (DE)lete  (L)ist  (R)epeat  (B)rowse

LC  Group  System  SSID  Objects  Description
---  -----  -----  -----  -----  -----
    
```

- Step 2** On the **COMMAND** line, enter the **ADD** primary command to add a new object group to the repository.

**Note:** If you enter the ADD primary command with the name of the new group, for example: ADD SYSBGRP, the **Group Name** field is filled in when the Group Detail Control - Add panel is displayed.

The Group Detail Control - Add panel is displayed, as shown in Figure 2-21 on page 2-53.

**Figure 2-21 Group Detail Control - Add Panel**

```

BMC Software ----- Group Detail Control - Add ----- MAINVIEW AutoOPERATOR
COMMAND ==>
Group Name      ==> _____
Command Prefix  ==> _____

Rule information:
Ruleset Dataset ==> _____
Ruleset Id      ==> ___   Rule Prefix ==> ___   Suffix ==> ___

Object Statistics:
Dataset Name    ==> _____
Disposition     ==> ___ (SHR/MOD)

Application information:
Author          ==> BAOKMZ1
Description     ==> _____

Enter END command to process and return or CANCEL to leave

```

**Step 3** Fill in the information for these required fields:

- Group Name
- Command Prefix
- Ruleset/Dataset
- Ruleset Id
- Rule Prefix
- Suffix

**Note:** The words *Minimal* and *Local* are reserved words in CSM. Group names cannot contain these words. If you attempt to create a group with the name MINIMAL or LOCAL, the error message “Reserved Name” is displayed.

Additions, changes, and deletion of object groups take place immediately. You do not need to issue a SAVE command after adding or changing a group.

Figure 2-22 on page 2-54 shows an example of a completed panel.

**Figure 2-22 Group Detail Control - Add Panel (Completed)**

```

BMC Software ----- Group Detail Control - Add ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                                    TGT --- KMZ1

Group Name          ==> SYSB_____
Command Prefix      ==> CSMKM_____

Rule information:
Ruleset Dataset     ==> BAOKMZ.AAO63.KMZ1.UBBPARM
Ruleset Id          ==> G2          Rule Prefix ==> ACM1      Suffix ==> 0001

Object Statistics:
Dataset Name        ==> BAOKMZ.KMZ1.OFFLOAD
Disposition         ==> MOD (SHR/MOD)

Application information:
Author              ==> BAOKMZ1
Description         ==> GROUP LIST FOR SYSB_____

Enter END command to process and return or CANCEL to leave
    
```

**Step 4** Press **PF3/END** to process the changes on this panel and return to the Object Groups panel (Figure 2-23).

**Figure 2-23 Object Groups Panel: One Group Added**

```

BMC Software ----- Object Groups ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                                    SCROLL ==> CSR
                                                    TGT --- KMZ1

Primary Commands: ADD      Locate      CMDSHOW      BUILD

LC CMDS ----- (BU)ild (S)ync (E)dit E(n)able (D)isable
                  (A)dd (DE)lete (L)ist (R)epeat (B)rowse

LC  Group      System      SSID      Objects      Description
__  _____  _____  _____  _____  _____
   KMZ1        SJSJ        KMZ1        20        GROUP LIST FOR SYSB
    
```

**Step 5** Press **PF3/END** again to return to the Global Overview panel.

## Entering the Object's Information

The following information describes some characteristics of objects:

- When an object is added to the repository it is associated with (or a made a member of) the group running on the local subsystem. The object can be associated with as many as 20 groups. The object is associated with a group as a member by using the (M)ember line command. Being a member of a group means that the object starts if it is not already up when the group is made active.
- If the object can be moved from one system to another, CSM assumes that it is intended to be running on *only one system at a time*. In this case, the object must be a member of only one group. It can be eligible to be moved to as many as 19 groups, but can only be a member of one group. When an object has a M(o)vable association with a group, the object will not start when that group is made active until you instruct CSM to move the object to that group's control, even if the group is running on another OS/390 image in the CSM-PLEX.
- The M(o)vable line command associates the object with the group it is able to move to. If the object is a member of more than one group and the user issues a M(o)vable line command against another group, CSM will issue an error message indicating that the object is presently a member of more than one group.
- An object's association with a group can be dissolved by using the (R)emove line command.
- If a user wants to know what other objects are associated with a particular group, the (L)ist line command will display the names of the objects in that group.

This section shows how to define a Normal object (from the Global Overview panel) named AAOCSM01. In this example, the new object is defined as a child of the OS/390 parent.

To define a new object named AAOCSM01, follow these steps:

- Step 1** From the Global Overview panel (Figure 2-11 on page 2-43), enter the **ADD** primary command on the **COMMAND** line,

or

Enter the **ADD** line command next to the name of the group to which you want the object to belong.

When using the ADD primary command, you can include the name of the object that is shown in the **Object Name** field when the Object Detail Control panel is displayed.

The Object Detail Control panel is displayed.

**Figure 2-24 Object Detail Control Panel: Adding a New Object**

```

BMC Software ----- Object Detail Control - Add ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                           TGT --- KMZ1

The following options are displayed in sequence, or may
be selected by entering the three-character code

CMD - Command Specifications          EV1 - Event Specifications 1
EV2 - Event Specifications 2         NOT - User Notification
GRP - Object Groups                  REQ - Object Prerequisites
CLS - Client/Server Relationships    SCH - Object Scheduling

Object Name          ==> AAOCSM01      Member of Group      ==> KMZ1____
Object Type          ==> NORM____      Type of object ( ? for list)

Verify Force Down   ==> N (Y/N)        Restart Only Control ==> N (Y/N)

Application information:
Author              ==> BAKMZ1
Description         ==> Child of MVS_____

Last Modified by BAKMZ1 on 01/30/2001 at 15:43

Enter END command to process and return, ENTER to continue, or CANCEL
    
```

When you use the ADD primary command, OS/390 is always indicated as the object's parent in the description field.

**Step 2** Fill in the information for these required fields:

- Object Name
- Object Type
- Member of Group
- Verify Force Down
- Restart Only Control

By default, the **Member of Group** field is filled in with the local subsystem ID when you use the ADD primary command from the Global Overview panel. You can also choose to enter the name of another group for the object to be associated with.

**Step 3** If you would like to see a list of Object Types, enter a question mark (?) in the **Object Type** field. A list of available object types is displayed (Figure 2-25 on page 2-57).

**Figure 2-25 CSM Object Types**

```

BMC Software ----- CSM Object Types ----- MAINVIEW AutoOPERATOR
COMMAND ==>

  Started Tasks:
S NORM   - Persistent Started Tasks
_ TRAN   - Transient Started Tasks

Enter END command to process and return to Detail Control, ENTER to continue

```

**Step 4** Select a type by placing an **S** next to it. Only one object type can be selected at a time.

**Step 5** Press **Enter** or **END** to return to the Object Detail Control panel.

The **Object Type** field contains the selected object type.

If you do not select an object type from the panel, the **Object Type** field on the Object Detail Control panel contains the field value prior to entering a question mark.

**Note:** When defining an object to CSM, once you specify the object type of Transient or Normal, you cannot change the specification. To change the specification, you must delete the object and create a new one.

**Step 6** Press **Enter** to continue to the 999next object definition panel, Command Specifications.

## Entering Command Specifications for an Object

The Command Specification panel enables you to specify an object start, stop, cancel and recovery commands (or EXECs). Post start and stop EXECs can also be specified.

Default start, stop and cancel commands are supplied on the panel and these defaults specify the most likely choices for starting, stopping and canceling an object.

The format of the default commands are as follows:

- **#S objectname** to start an object
- **#P objectname** to stop an object
- **#C objectname** to cancel an object

See Figure 2-26 for an example of the Command Specifications panel.

**Figure 2-26 Command Specifications Panel**

```

BMC Software ---- Command Specifications for AAOCSM01 - Add ---- AutoOPERATOR
COMMAND ==> TGT --- KMZ1

Start Command/EXEC      ==> #S AAOCSM01_____
                        _____
User Post Start EXEC    ==> _____
                        _____
Stop Command/EXEC      ==> #P AAOCSM01_____
                        _____
User Post Stop EXEC     ==> _____
                        _____
Cancel Command/EXEC    ==> _____
                        _____
Recovery Command/EXEC  ==> _____

Start Command Time Out ==> 0002      Start Command Limit ==> 3____
Stop Command Time Out ==> 0002      Perform Count Reset ==> Y (Y/N)

Conditional Start      ==> N (Y/N)   Conditional Stop      ==> N (Y/N)

Enter END command to process and return to Detail Control, ENTER to continue
    
```

**Using Variables in the Object Definition Fields**

On the Command Specifications panel, you can specify both simple and compound variables in the Start, Stop, Post Start, Post Stop, Cancel and Recovery command fields. Some basic variables are supplied:

- **CSMDAY**: resolves to the current day of the week in three-character format
- **CSMMON**: resolves to the current month in three-character format
- **CSMDATE**: resolves to the current date in the format `yyyymmdd`
- **CSMGROUP**: resolves to the name of the group active on the local subsystem

- **CSMSYSNM**: resolves to the **SYSNAME** from the local subsystem's OS/390 system control blocks

Variables from the **SHARED** pool can also be used in the **Start, Post Start, Stop, Post Stop, Cancel** and **Recovery** command fields.

**Note:** The only variables that cannot be used for this facility are the variables prefixed with **IMF** that are supplied by **MAINVIEW AutoOPERATOR**.

## Example

An example of a start command containing a simple variable name follows:

```
#S AAOKMZSS , START=&STATYPE , VPOOL=&RESET
```

The variable is resolved when the CSM subsystem application processes the command and the variable is read from the **SHARED** variable pool. If the variable is not present in the **SHARED** pool, it resolves as the variable name (as in **REXX** language coding).

Using a compound variable is the same as a simple variable. For example, if an object has a start command defined as

```
#S AAOCSM09 , PARM=&AO . CSMDAY
```

the variable **AO.CSMDAY** is treated in the same manner as a **REXX** compound symbol: **AO** is the stem and **CSMDAY** is the tail.

On Monday, the variable resolves into shared variable '**AO.MON**'. It contains the value **STUFF**. The resulting command is as follows:

```
#S AAOCSM09 , PARM=STUFF
```

You can set the variable with a **Rule**, an **EXEC**, or an entry in the **BBIVARxx** member. The variable must already be present when CSM attempts to execute the defined command.

This function provides flexibility to specify **Start** and **Stop** commands that can change on a periodic basis. It also provides flexibility for object definitions in the shared repository whose **Started Task** name is the same across multiple partners but use significantly different parameters from system to system.

All commands entered in any of the command fields: **Start**, **Stop**, **Post Start**, **Post Stop**, **Cancel** and **Recovery** must be prefixed by one of the following control characters:

- period ( . )
- pound sign (#)
- forward slash (/)
- question mark (?)
- percent sign (%)

Single quotation marks can be used in commands. The quotation marks must be doubled, so that they are properly processed; for example:

```
#S ABCPDQ.PDQ,LIBRARY=' 'SYS1.PARMLIB' '
```

CSM running in the BBI-PAS strips out the extra quotation marks and submits the intended command to the system for execution. If an object used the previous example as its defined start command, it would be translated to the following command and submitted for execution:

```
#S ABCPDQ.PDQ,LIBRARY='SYS1.PARMLIB'
```

You can also use values defined in the static system symbol list. These values must be prefixed with two ampersand signs (&&); for example:

```
#S NET, , ,LIST=&&SUBAREA
```

The resolved command that is passed to OS/390 is

```
S NET, , ,LIST=&SUBAREA
```

When OS/390 processes this command, **&SUBAREA** will be substituted with a value defined in the static system symbol list.

If the command is not prefixed with a control character, MAINVIEW AutoOPERATOR tries to process the command as an EXEC using EXEC parameters.

## Completing the Additional Fields

You can also use the Command Specifications panel to specify

- Start and Stop Command Time Out values in minutes (refer to “Recovering Failed Objects” on page 2-17 for more information)
- Start Command Limit value (refer to “Specifying Start Command Limits” on page 2-18 for more information)

- whether to reset the Start Command count where a **Y** or **N** specification determines whether the object command count shared variable is reset each midnight to zero
- Conditional Start or Conditional Stop setting where a **Y** setting in either field will cause the Conditional Commands Specification panel to appear (refer to “Specifying Conditional Command Specifications for an Object” on page 2-61)

**Step 7** Press **Enter** to continue to the next object definition panel.

## Specifying Conditional Command Specifications for an Object

If you entered **Y** for either the Conditional Start or the Conditional Stop fields on the Command Specifications panel (refer to Figure 2-26 on page 2-58), the Conditional Commands Specification panel is displayed. Figure 2-27 shows the panel where you specify conditional Start commands.

**Figure 2-27** Conditional Command Specifications Panel: Start

```

BMC Software ----- Conditional Command Specification ----- MAINVIEW AutoOPERATOR
COMMAND =====>                                     Object --- MVS

After ____ minutes issue =====> _____
Start Command/EXEC #1                               _____

After ____ minutes issue =====> _____
Start Command/EXEC #2                               _____

After ____ minutes issue =====> _____
Start Command/EXEC #3                               _____

After ____ minutes issue =====> _____
Start Command/EXEC #4                               _____

After ____ minutes issue =====> _____
Start Command/EXEC #5                               _____

Enter END command to process and return to Detail Control, ENTER to continue

```

Figure 2-28 on page 2-62 shows an example of the Stop commands panel.

**Figure 2-28 Conditional Command Specifications Panel: Stop**

```

BMC Software ----- Conditional Command Specification ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                                    Object --- MVS

After ____ minutes issue ==> _____
Stop Command/EXEC #1      _____

After ____ minutes issue ==> _____
Stop Command/EXEC #2      _____

After ____ minutes issue ==> _____
Stop Command/EXEC #3      _____

After ____ minutes issue ==> _____
Stop Command/EXEC #4      _____

After ____ minutes issue ==> _____
Stop Command/EXEC #5      _____

Enter END command to process and return to Detail Control, ENTER to continue
    
```

Use a conditional Start or Stop command when you want a command issued after a specified period of minutes has elapsed after either the defined Start or the defined Stop command is issued or another conditional command has been issued for this object.

**Note:** The conditional command will not be issued if the object's Start or Stop event is triggered.

If the Start or Stop command and all the conditional Start or Stop commands are issued and the object Start or Stop event has not occurred, an ALERT (ACM720A) will be issued informing you that the command has timed out.

**Step 1** Enter the number of minutes after which a conditional Start or Stop command will be issued for an object.

**Step 2** Enter the conditional Start or Stop command.

You must enter each conditional command sequentially on a new line without skipping between commands.

Each command can be up to 126 bytes long and can contain variables. Any variables specified will be resolved from the local subsystem's SHARED pool.

When a conditional command is issued after the original command, CSM issues an escalated ALERT. An example of the ALERT (ACM789A) is

```
ACM789A Conditional command found for <object>.
        Unless a <start|stop> event fires for <object>
        the command will be issued in <xx> minutes.
```

**Step 3** Press **Enter** to continue to the next object definition panel, Event Specifications 1.

## Specifying Start and Stop Events: Event Specifications 1

Use the first Event Specification panel to specify text identifiers for Start and Stop events. On this panel you can specify the Text-ID and type for up to five Start events and five Stop events.

For each event specification, you can specify

- the type of job that will issue the event (JOB, STC or TSO)
- the name of the job that will issue the event

Figure 2-29 shows an example of the Event Specifications 1 panel:

**Figure 2-29** Event Specifications 1 Panel

```
BMC Software ----- Event Specifications 1 for AAOCSM01 - Add ----- AutoOPERATOR
COMMAND ==>                                     TGT --- AO6A

Event Identifiers / Type:                               Generate ==> YES

Start Events:

Text Id #1 ==> $HASP373_____ / MSG_   Job Type / Name ==> STC / AAOCSM01
Text Id #2 ==> _____ / _____ Job Type / Name ==> ___ / _____
Text Id #3 ==> _____ / _____ Job Type / Name ==> ___ / _____
Text Id #4 ==> _____ / _____ Job Type / Name ==> ___ / _____
Text Id #5 ==> _____ / _____ Job Type / Name ==> ___ / _____

Stop Events:

Text Id #1 ==> $HASP395_____ / MSG_   Job Type / Name ==> STC / AAOCSM01
Text Id #2 ==> _____ / _____ Job Type / Name ==> ___ / _____
Text Id #3 ==> _____ / _____ Job Type / Name ==> ___ / _____
Text Id #4 ==> _____ / _____ Job Type / Name ==> ___ / _____
Text Id #5 ==> _____ / _____ Job Type / Name ==> ___ / _____

Enter END command to process and return to Detail Control, ENTER to continue
```

1. Enter the Text-IDs and the event types for each of the Start or Stop events.

You must enter each Text-ID and event type sequentially on a new line without skipping between lines. The valid event types are MSG, CMD, ALRT, and JRNL.

2. Enter the type of job that will issue the event (JOB, STC or TSO).
3. Enter the name of the job that will issue the event. Wildcard characters + and \* can be used as they would be in a Rule.
4. Press **Enter** to continue to the next object definition panel, Event Specifications 2.

## Specifying Abnormal Termination Events: Event Specifications 2

Use the second Event Specification panel to specify identifiers for Abnormal Termination events. You can specify information for up to five Abnormal Termination events.

For each Abnormal Termination event, you can specify

- the type of job that issues the event (JOB, STC or TSO)
- the name of the job that issues the event

Figure 2-30 on page 2-65 shows an example of the Event Specifications 2 panel:

**Figure 2-30 Event Specifications 2 Panel**

```

BMC Software ----- Event Specifications 2 for AAOCSM01 - Add ----- AutoOPERATOR
COMMAND ==>                                     TGT --- KMZ1

Abnormal Termination Events:                               Generate ==> YES

Text Id      ==> IEF450I_____ Job Name ==> _____ Job Type   ==> ___
Text String  ==> ABEND=S0C7_____ Word/Op ==> WORD5 / EQ Event Type ==> ___

Text Id      ==> _____ Job Name ==> _____ Job Type   ==> ___
Text String  ==> _____ Word/Op ==> ___ / ___ Event Type ==> ___

Text Id      ==> _____ Job Name ==> _____ Job Type   ==> ___
Text String  ==> _____ Word/Op ==> ___ / ___ Event Type ==> ___

Text Id      ==> _____ Job Name ==> _____ Job Type   ==> ___
Text String  ==> _____ Word/Op ==> ___ / ___ Event Type ==> ___

Text Id      ==> _____ Job Name ==> _____ Job Type   ==> ___
Text String  ==> _____ Word/Op ==> ___ / ___ Event Type ==> ___

Enter END command to process and return to Detail Control, ENTER to continue
    
```

**Step 1** (optional) Specify the abend target string in the **Text String** fields to search for within the Text ID specified.

Perform this step when you want to limit recovery to a particular abend code for the message indicated.

Every message handled is broken into separate values (delimited by a space) and stored in WORDx variables. You must specify which word you expect to match the search string with. You also need to indicate the type of comparison operation that is to take place between the search string and the WORDx variable.

For example, you want to define an object where abnormal termination processing should occur only when the fifth word of the IEF450I message contains the string ABEND=S0C7. In this case, the Exceptions Specification panel would be filled in to look like Figure 2-30.

**Step 2** Enter the Text IDs and the event types for each of the Abnormal Termination events in the **Text Id** and **Event Type** fields.

You must enter each Text ID and event type sequentially on a new line without skipping between lines.

**Step 3** Enter the type of job (JOB, STC or TSO) that will issue the event in the **Job Type** field.

**Step 4** Enter the name of the job that will issue the event in the **Job Name** field. Wildcard characters + and \* can be used as they would in a Rule.

**Step 5** Press **Enter** to continue to the next object definition panel, User Notification.

## Specifying User Notification

Use the User Notification panel to

- send TSO messages
- send pages by using the Elan Workstation product option (if it is installed at your site)
- manage the ALERT queue
- notify up to 10 TSO users of error conditions in object processing

In addition, up to 10 different pages can be sent when the Elan Workstation product option is installed at your site.

**Note:** The message sent to the TSO users is a constant and cannot be tailored by the administrator.

Figure 2-31 on page 2-67 shows an example of the User Notification panel:

Figure 2-31 User Notification Panel

```

BMC Software --- User Notification for AAOCSM01 - code --- MAINVIEW AutoOPERATOR
COMMAND ===>                                     TGT --- KMZ1

TSO SEND ===> _____
Users      _____

Pager Notification

Contact # 1 ===> _____ Info ===> _____
Contact # 2 ===> _____ Info ===> _____
Contact # 3 ===> _____ Info ===> _____
Contact # 4 ===> _____ Info ===> _____
Contact # 5 ===> _____ Info ===> _____
Contact # 6 ===> _____ Info ===> _____
Contact # 7 ===> _____ Info ===> _____
Contact # 8 ===> _____ Info ===> _____
Contact # 9 ===> _____ Info ===> _____
Contact # 10 ===> _____ Info ===> _____

Object Alert Queue ===> MAIN_____ Alert Publish Mode ===> ADD_____

Enter END command to process and return to Detail Control, ENTER to continue

```

**Step 1** Enter any valid one- to seven-character TSO user IDs for each user to be notified of error conditions in object processing in the **TSO SEND Users** fields.

**Step 2** (Perform this step if you have the Elan Workstation option installed and operating at your site.) Enter

- the 1- to 32-character value in the **Contact** field specified for the NAME keyword of the IMFEXEC NOTIFY command (used to page operators when Elan Workstation is installed).
- the 1- to 12-character value in the **Info** field specified for the INFO keyword used in the IMFEXEC NOTIFY command (used to page operators when Elan Workstation is installed).

Refer to the IMFEXEC NOTIFY section in the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.

**Step 3** In the **Object Alert Queue** field, enter the name of the ALERT queue that will receive ALERTs for this object from CSM.

**Step 4** In the **Alert Publish Mode** field, specify whether an ALERT for this object is published and how it is published to connected PATROL EM workstations that have subscribed to receive ALERTs through the General Message Exchange (GME).

Possible values are as follows:

- REPLACE** An ALERT replacement for the ALERT's key and queue is sent to all MAINVIEW AutoOPERATOR PATROL Enterprise Manager Integration (MAPEMI) workstations that have subscribed to receive ALERTs from this MAINVIEW AutoOPERATOR. If an ALERT with that key and queue on a MAPEMI workstation already exists, it is deleted before writing the new ALERT with that key and queue.
- ADD** An ALERT add is sent to all workstations that have subscribed to receive ALERTs from this MAINVIEW AutoOPERATOR. If an ALERT with that key and queue on a MAPEMI workstation already exists, it is not deleted before writing the new ALERT with that key and queue.
- ADD is the default.
- NO** The ALERT is not written to the connected MAPEMI workstations even if they have subscribed to receive ALERTs.

**Step 5** Press **Enter** to continue to the next object definition panel, Object Groups.

## Associating an Object with a Group

Use the Object Groups panel to associate an object with a particular object group. Every CSM partner has its own object group. Any BBI-SS PAS running CSM can have only one group. When an object is a member of a group, CSM will start that object (if it is not already up) on a BBI-SS PAS when the group is active.

**Note:** The first time you use CSM, you must define a group before you can start defining objects. Refer to "Creating a Group" on page 2-51 for more information about creating a group. This section describes only the process of associating the object with one or more groups.

Figure 2-32 on page 2-69 shows an example of the Object Groups panel:

**Figure 2-32 Object Groups Panel**

```

BMC Software ----- Object Groups for AAOCSM01 - Add ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     SCROLL ==> CSR
                                                TGT --- KMZ1

Primary Commands: Locate      CMDSHOW      CANCEL

LC CMDS ----- (R)emove (L)ist      (M)ember M(o)veable

Enter END command to process and return to Detail Control, ENTER to continue

LC  Group      Status      Objects      Description
---  ---      -
___ KMZ1      Member      9      Group List for SYSD Again
___ KMZ2      Member      23     Group List for SJSE
___ KMZ6      Member      4      Group List for SYSD
___ KMZ7      Member      80     test group 7
***** Bottom of data *****
    
```

The **Status** field indicates the object’s association to all of the groups in the CSM-PLEX. The **Objects** field lists the number of object within each group.

- To associate an object with a group, enter the **(M)ember** line command next to the group name that you want to associate an object with.
- To dissolve the association between an object and its group, enter the **(R)emove** line command next to the group that the object is associated with.
- To see all the objects within a group, enter the **(L)ist** line command next to the group name. Figure 2-33 shows an example of all the objects in a group named KMZ2.

**Figure 2-33 Object Groups Panel**

```

BMC Software ----- Objects in Group List KMZ2 ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     SCROLL ==> CSR
                                                TGT --- KMZ1

Enter END or ENTER to continue

AAOCSM02
AAOCSM08
AAOCSM10
AAOCSM17
AAOCSM22
***** Bottom of data *****
    
```

- Press **Enter** to continue to the next Object Groups panel.

## Moving an Object between Groups

When an object is a member of a group, CSM starts that object (if it is not already UP) on a BBI-SS PAS when the group is active. Therefore, an object (and its schedule) can be defined once and then if it runs on multiple CSM partners, you can associate the object, using the (M)ember line command as described in “Associating an Object with a Group” on page 2-68, with a group on each of the partners. Each object can be a member of up to 20 CSM groups.

You can also define an object that is normally active on only one CSM partner and moveable to another BBI-SS PAS (in cases of emergency backups or if you need more system resources) with the M(o)veable line command. In this case, the object is defined as a member to one group where you want the object to be normally active and, as moveable to all groups where you would possibly move the object to. Once an object is defined as Moveable to any groups, this object can be a Member of only one group, and can be moveable to 19 groups.

**Step 1** To make an object moveable to another system, enter the M(o)veable line command next to the group name to which you want to be able to move the object.

This example shows object AAOCISM01 is a member of group KMZ1 but moveable to KMZ2 and KMZ6.

**Figure 2-34 Object Groups Panel: Creating a Moveable Object**

```

BMC Software ----- Object Groups for AAOCISM01 - Add ----- MAINVIEW AutoOPERATOR
COMMAND ==>>>                                     SCROLL ==>>> CSR
                                                    TGT --- KMZ1

Primary Commands: Locate      CMDSHOW      CANCEL

LC CMDS ----- (R)emove (L)ist      (M)ember  M(o)veable

Enter END command to process and return to Detail Control, ENTER to continue

LC  Group      Status      Objects  Description
---  -----  -
___  KMZ1      Member      9      Group List for SYSD Again
o___ KMZ2      Member      23     Group List for SJSE
o___ KMZ6      Member      4      Group List for SYSD
___  KMZ7      Member      80     test group 7
***** Bottom of data *****
    
```

**Note:** An object can be made moveable to as many as 20 groups. Once an object is denoted as moveable, it can be the member of only 1 group.

When an object is made moveable to a group, the object will not automatically start when that group is active until you instruct CSM to move the object to that group's control. For an example of moving an object, refer to "Moving an Object: Using the Moveable Attribute" on page 2-125.

**Step 2** Press **Enter** to continue to the next object definition panel, Object Requisites.

## Defining the Object's Parents

Figure 2-35 shows the Object Requisites panels for AAOCSM01 where the object's parents can be added.

**Figure 2-35 Object Requisites for AAOCSM01 Panel**

```

BMC Software ---- Object Requisites for AAOCSM01 - Add --- MAINVIEW AutoOPERATOR
COMMAND ==>                                     SCROLL ==> CSR
                                                TGT --- KMZ1

Primary Commands: ADD END CMDSHOW

LC CMDS ----- (E)dit Object, (D)elele Object

Enter END command to process and return to Detail Control, ENTER to continue

LC Parent Object   Group           Description
-----
_   MVS            KMZ1            THE BIG GIANT HEAD
***** Bottom of data *****
    
```

**Step 3** For this example, there are no parents of AAOCSM01, so to add a parent, enter the (A)DD primary command. The Add Object Requisites panel is displayed.

**Figure 2-36 Add Object Requisites for AAOCSM01 Panel**

```

BMC Software ---- Object Requisites for AAOCSM01 - Add --- MAINVIEW AutoOPERATOR
C | BMC Software           Add Object Requisites           AutoOPERATOR |
P | COMMAND ==> _____ TGT --- KMZ1 |
L | Requisite ==> _____ |
E | Group Name ==> _____ |
L | Press ENTER to continue processing |
  | Press END(PF3) / CANCEL to cancel request |
- | _____ |
* | _____*
    
```

If you wanted to add parents, you would specify the information in the Add Object Requisites panel.

**Step 4** Press **Enter** to continue to the next object definition panel, Client/Server Relationships.

## Defining the Object's Servers

Figure 2-37 shows the Client/Server Relationships for the AAOCSM01 panel.

**Figure 2-37 Client/Server Relationships for AAOCSM01 Panel**

```

BMC Software -- Client/Server Relationships for AAOCSM01 - Add --- AutoOPERATOR
COMMAND ==>                                SCROLL ==> CSR
                                                TGT --- KMZ1

Primary Commands: ADD END CMDSHOW

LC CMDS ----- (E)dit server, (D)elite server

Enter END command to process and return to Detail Control, ENTER to continue

LC  Server      Group      Description
                                Connect Command
                                Disconnect Command

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

**Step 5** For this example, there are no servers of AAOCSM01, so to add a server, enter the **(A)DD** primary command. The Add Client / Server Relationships panel is displayed.

**Figure 2-38 Add Client/Server Relationships for AAOCSM01 Panel**

```

BMC Software -- Client/Server Relationships for AAOCSM01 - Add --- AutoOPERATOR
C | BMC Software      Add Client / Server Relationships      AutoOPERATOR |
  | COMMAND ==> _____ TGT --- KMZ1 |
P |
  |
L |   Server          ==> _____
  |   Group Name     ==> _____
E |   Connect Command ==> _____
  |   Disconnect Command ==> _____
L |
  |
  | Press ENTER to continue processing
  | Press END(PF3) / CANCEL to cancel request
* -----*
    
```

If you wanted to add servers, you would specify the information in the Add Client / Server Relationships panel.

**Step 6** Press **Enter** to continue to the next object definition panel, Object Scheduling.

Figure 2-39 shows the Object Scheduling for AAOCSM01 panel.

**Figure 2-39 Object Scheduling for AAOCSM01 Panel**

```

BMC Software --- Object Scheduling for AAOCSM01 - Add --- MAINVIEW AutoOPERATOR
COMMAND ===>                                SCROLL ===> CSR
                                                TGT --- KMZ1

Primary Commands: ADD CANcel END CMDSHOW

LC CMDS ----- (E)dit Time, (D)elete Time

Enter END to process/return to Detail Control, ENTER to continue, or CANcel

LC Object      Day      Down From      Down To
  Group                HH:MM          HH:MM

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

For this example, a schedule is not created for object AAOCSM01. Refer to “Modifying an Object’s Routine Schedule” on page 2-93 for information about creating a schedule for an object.

**Step 7** Press **Enter** to return to the CSM Object Detail Control panel (Figure 2-40).

**Figure 2-40 Object Detail Control Panel**

```

BMC Software ----- Object Detail Control - Add ----- MAINVIEW AutoOPERATOR
COMMAND ===>                                TGT --- KMZ1

The following options are displayed in sequence, or may
be selected by entering the three-character code

  CMD - Command Specifications          EV1 - Event Specifications 1
  EV2 - Event Specifications 2         NOT - User Notification
  GRP - Object Groups                 REQ - Object Prerequisites
  CLS - Client/Server Relationships    SCH - Object Scheduling

Object Name      ===> AAOCSM01          Member of Group   ===> KMZ1
Object Type      ===> NORM_____      Type of object ( ? for list)

Verify Force Down  ===> N (Y/N)        Restart Only Control ===> N (Y/N)

Application information:
Author           ===> BAOKMZ1
Description      ===> Child of MVS_____

Last Modified by BAOKMZ1 on 01/30/2001 at 17:36

Enter END command to process and return, ENTER to continue, or CANcel
***** Bottom of data *****
    
```

**Step 8** Press **PF3/END** to return to the CSM Global Overview panel (Figure 2-41).

**Figure 2-41 CSM Global Overview Panel**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>>                                SCROLL ==>> CSR
                                           TGT --- KMZ1
Primary Commands: EXCeption  Group      SORT      SAVE      ADD      PROFile
                  CALendar   CANcel    UNsort    CONVERT   Build
                  Locate     CMDSHOW Xref      SAVERULE  SHUTSYS

LC CMDS ----- (S)tart   b(O)unce  rese(T)   depe(N)dents (A)dd
                  sto(P)    (EX)ec   (E)dit    c(L)ients  (D)elete
                  (CAN)cel (M)ove  (R)epeat (B)rowse   s(Y)nc

LC  Object  Group  System  Actual  Desired  Ind  Type
---  -----  -----  -----  -----  -----  ---  -----
___  AAOCSM01  KMZ1   SJSC    UP      UP      L   NORM
___  MVS
    
```

Object AAOCSM01 is added as the child of OS/390.

Repeat this procedure for all the new Normal objects that you want to define.

**Hint**

For the new objects to be added to the repository, you must issue the ADD primary command on the CSM panel.

After you issue the ADD command, the object is added. The SAVE command stores its definition in the repository. CSM begins controlling the object after the BUILD command is processed.

## Defining a Transient Object

To define a Transient object, follow the same instructions as for defining a Normal object (pages 2-55 through 2-74). The differences are as follows:

- Specify **TRAN** (Transient) for the **Object Type** field instead of **NORM** (Normal) on the Object Detail Control panel.
- Do not specify anything in the **Stop Command/EXEC** field on the Command Specifications panel.

**Note:** When defining an object to CSM, once you specify the object type of Transient or Normal, you cannot change the specification. To change the specification, you must delete the object and create a new one.

## Defining a Grouping Object

To define a Grouping object, follow the same instructions as for defining a Normal object (pages 2-55 through 2-74).

The only difference is that you do not specify a start command in the **Start Command/EXEC** field on the Command Specifications panel. The absence of the start command for an object is what CSM uses to differentiate between Grouping objects and Transient objects.

## Comparing CSM Object Types

Table 2-2 summarizes the similarities and differences for the three object types.

**Table 2-2 Comparison between Three CSM Object Types**

Object Type	Can Be Parents	Can Have Schedules	Can Be Clients or Servers	Requires a CSM Start Command	Requires a CSM Stop Command	Can Be Manually Started	Can Be Manually Stopped
Normal Objects	Y	Y	Y	Y	Y	Y	Y
Transient Objects	Y	Y		Y		Y	
Grouping Objects	Y	Y				Y	Y

## Using the Repeat Line Command to Create Objects

From the Global Overview panel, use the line command (R)repeat to duplicate an existing object’s definition to create a new object with the same information. You can use this line command when you want to define objects with similar attributes.

The only limitation is that no two objects can have the same name so you must create a new name for the new object.

To use the (R)repeat line command, follow this procedure:

- Step 1** From the Global Overview panel (Figure 2-42), enter the (R)repeat line command in the LC field next to the object that you want to duplicate.

**Figure 2-42 Repeating an Object**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>>                                     SCROLL ==>> CSR
                                                    TGT --- AO6A
Primary Commands: EXCeption  Group      SORT      SAVE      ADD      PROFile
                  CALendar  CANcel  UNsort    CONVERT   BUILD
                  Locate    CMDSHOW Xref      SAVERULE  SHUTSYS

LC CMDS ----- (S)tart    b(O)unce   rese(T)   depe(N)dents (A)dd
                  sto(P)    (EX)ec    (E)dit    c(L)ients  (D)etele
                  (CAN)cel  (M)ove    (R)peat  (B)rowse

LC  Object      Group      System      Actual      Desired      Ind      Type
---  ---
___  AAOCMS01    AO6A                UNKNOWN     UNKNOWN
___  AAOCMS02    AO6A                UNKNOWN     UNKNOWN
___  CSBEMSTR    AO63      SJSE        STARTING    UP           XE      NORM
___  MVS

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

```

The Repeat Object Definition panel is displayed (Figure 2-43 on page 2-77).

**Figure 2-43 Repeat Object Definition Panel**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
C | BMC Software          Repeat Object Definition          AutoOPERATOR
  | COMMAND ==>> _____ TGT --- AO6A                | A
P |
  |
  | Old Object Name ==>> AAOCMS01
L |
  | New Object Name ==>> _____
  |
L | Press ENTER to continue or END / CANCEL to abort
  |
R_ AAOCMS01 AO6A          UNKNOWN      UNKNOWN      NORM
   AAOCMS02 AO6A          UNKNOWN      UNKNOWN      NORM
   CSBEMSTR AO63      SJSE      STARTING      UP          XE      NORM
   MVS
***** Bottom of data *****

```

**Step 2** Enter a new unique name for the object in the **New Object Name** field.

**Figure 2-44 Add New Object Panel: Repeat Line Command Example 2**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
C | BMC Software          Repeat Object Definition          AutoOPERATOR
  | COMMAND ==>> _____ TGT --- AO6A                | A
P |
  |
  | Old Object Name ==>> AAOCMS01
L |
  | New Object Name ==>> AAOCMS03
  |
L | Press ENTER to continue or END / CANCEL to abort
  |
R_ AAOCMS02 AO6A          UNKNOWN      UNKNOWN      NORM
   AAOCMS02 AO6A          UNKNOWN      UNKNOWN      NORM
   CSBEMSTR AO63      SJSE      STARTING      UP          XE      NORM
   MVS
***** Bottom of data *****

```

- Step 3** Press **Enter** to process the changes on this panel. The Object Detail Control panel is displayed (Figure 2-45).

**Figure 2-45 Object Detail Control - Repeat**

```

BMC Software ----- Object Detail Control - Repeat ----- MAINVIEW AutoOPERATOR
COMMAND ===>                                     TGT --- KMZ1

The following options are displayed in sequence, or may
be selected by entering the three-character code

    CMD - Command Specifications           EV1 - Event Specifications 1
    EV2 - Event Specifications 2         NOT - User Notification
    GRP - Object Groups                  REQ - Object Prerequisites
    CLS - Client/Server Relationships    SCH - Object Scheduling

Object Name          ===> AAOCSM03          Member of Group      ===> KMZ1_____
Object Type          ===> NORM_____       Type of object ( ? for list)

Verify Force Down   ===> N (Y/N)          Restart Only Control ===> N (Y/N)

Application information:
Author              ===> BAOKMZ1
Description         ===> Child of MVS_____

Last Modified by BAOKMZ1 on 01/30/2001 at 17:36

Enter END command to process and return, ENTER to continue, or CANCEL

```

- Step 4** Follow the sequence of the panels by pressing **Enter** and fill in the information for new object AAOCSM03.
- Step 5** After all the information for object AAOCSM03 is completed, the Object Detail Control panel is redisplayed. To process the creation of object AAOCSM03, press **PF3/END**.

The Global Overview panel is redisplayed with the new object.

**Figure 2-46 AAOCSM03 Object Created**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     SCROLL ==> CSR
                                                TGT --- KMZ1
Primary Commands: EXCeption  Group    SORT    SAVE    ADD    PROFile
                  CALEndar  CANcel  UNsort  CONVERT  Build
                  Locate    CMDSHOW Xref    SAVERULE SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)  depe(N)dents  (A)dd
                  sto(P)   (EX)ec   (E)dit   c(L)ients   (D)elete
                  (CAN)cel (M)ove   (R)epeat (B)rowse   s(Y)nc

LC   Object  Group  System  Actual  Desired  Ind  Type
----  -----  -----  -----  -----  -----  ---  ----
___  AAOCSM01  AO6A                UNKNOWN  UNKNOWN                NORM
___  AAOCSM02  AO6A                UNKNOWN  UNKNOWN                NORM
___  AAOCSM03  AO6A                UNKNOWN  UNKNOWN                NORM
___  CSBEMSTR  AO63   SJSE    STARTING  UP                XE  NORM
___  MVS
***** Bottom of data *****

```

## Modifying Object Definitions

It might be necessary to modify an object’s relationships; for example, making changes, additions, or deletions to its list of parents. The following sections describe how to modify an object’s relationship to another object.

### Adding Parents

In this example, parent OS/390 has three child objects: AAOCSM08, AAOCSM09 and AAOCSM10. This process shows how to add AAOCSM10 to AAOCSM08’s list of parents.

To add parent AAOCSM10 to child object AAOCSM08, follow these steps:

**Step 1** Display the Global Overview panel (Figure 2-47).

**Figure 2-47 Global Overview Panel: Adding a Parent**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     SCROLL ==> CSR
                                                TGT --- KMZ1

Primary Commands: EXCeption  Group      SORT      SAVE      ADD      PROFile
                  CALendar   CANcel    UNsort    CONVERT   Build
                  Locate     CMDSHOW  Xref      SAVERULE  SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)   depe(N)dents (A)dd
                  sto(P)   (EX)ec   (E)dit    c(L)ients  (D)elete
                  (CAN)cel (M)ove  (R)epeat  (B)rowse   s(Y)nc

LC  Object  Group  System  Actual  Desired  Ind  Type
___  _____  _____  _____  _____  _____  _____  _____
___  AAOCSM08  AO6A      UNKNOW  UNKNOW  UNKNOW  UNKNOW  NORM
___  AAOCSM09  AO6A      UNKNOW  UNKNOW  UNKNOW  UNKNOW  NORM
___  AAOCSM10  AO6A      UNKNOW  UNKNOW  UNKNOW  UNKNOW  NORM
___  CSBEMSTR  AO63      SJSE    STARTING  UP      XE      NORM
___  MVS
***** Bottom of data *****
    
```

**Step 2** Enter the (E)dit line command next to AAOCSM08.

The Object Detail Control panel is displayed.

**Figure 2-48 Adding a Parent - Step 2**

```

BMC Software ----- Object Detail Control - Edit ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     TGT --- KMZ1

The following options are displayed in sequence, or may
be selected by entering the three-character code

  CMD - Command Specifications          EV1 - Event Specifications 1
  EV2 - Event Specifications 2          NOT - User Notification
  GRP - Object Groups                  REQ - Object Prerequisites
  CLS - Client/Server Relationships     SCH - Object Scheduling

Object Name          ==> AAOCSM08          Member of Group     ==> AO6A___
Object Type          ==> NORM___          Type of object ( ? for list)

Verify Force Down   ==> N (Y/N)          Restart Only Control ==> N (Y/N)

Application information:
Author              ==> BAOKMZ1
Description         ==> Child of MVS_____

Last Modified by BAOKMZ1 on 01/30/2001 at 17:36

Enter END command to process and return, ENTER to continue, or CANCEL
    
```

**Step 3** Enter the **REQ** primary command to skip the panels in the ring and display the Object Requisites panel (Figure 2-49).

**Figure 2-49 Adding a Parent - Step 3**

```

BMC Software --- Object Requisites for AAOCSM08 - Edit --- MAINVIEW AutoOPERATOR
COMMAND ===>                                SCROLL ===> CSR
                                                TGT --- KMZ1

Primary Commands: ADD END CMDSHOW

LC CMDS ----- (E)dit Object, (D)elete Object

Enter END command to process and return to Detail Control, ENTER to continue

LC Parent Object   Group           Description
-----
_   MVS            AO6A            Child of MVS
***** Bottom of data *****
    
```

**Step 4** Enter the **ADD** primary command to display the Add Object Requisites panel:

**Figure 2-50 Adding a Parent - Step 4**

```

BMC Software ---- Object Requisites for AAOCSM08 - Add --- MAINVIEW AutoOPERATOR
C | BMC Software           Add Object Requisites           AutoOPERATOR |
  | COMMAND ===> _____ TGT --- KMZ1 |
P | |
L | Requisite ===> _____ |
  | Group Name ===> _____ |
E | |
L | Press ENTER to continue processing |
  | Press END(PF3) / CAncel to cancel request |
_ | |
* |-----*
    
```

**Step 5** In the **Requisite** field, type the new object name, and in the **Group Name** field, type the group name of the parent to be added to CSM object AAOCSM0, as shown in Figure 2-51 on page 2-82.

**Figure 2-51 Adding a Parent - Step 5**

```

BMC Software ---- Object Requisites for AAOCSM08 - Add --- MAINVIEW AutoOPERATOR
C | BMC Software          Add Object Requisites          AutoOPERATOR |
  | COMMAND ==> _____ TGT --- KMZ1 |
P |
  |
L |   Requisite ==> AAOCSM10
  |   Group Name ==> AO6A
E |
  |
L |   Press ENTER to continue processing
  |   Press END(PF3) / CANcel to cancel request
- |
* -----*
    
```

**Step 6** Press **PF3/END** to process. The Object Detail Control panel is redisplayed.

**Step 7** Press **PF/3END** again. The Global Overview panel is redisplayed.

**Figure 2-52 Adding a Parent - Step 6**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>
                                     SCROLL ==> CSR
                                     TGT --- KMZ1

Primary Commands: EXCeption  Group    SORT    SAVE    ADD    PROFile
                  CALendar  CANcel  UNsort  CONVERT  Build
                  Locate    CMDSHOW Xref    SAVERULE SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)  depe(N)dents  (A)dd
                 sto(P)  (EX)ec   (E)dit   c(L)ients    (D)elete
                 (CAN)cel (M)ove   (R)epeat (B)rowse    s(Y)nc

LC  Object  Group  System  Actual  Desired  Ind  Type
---  ---    ---    ---    ---    ---    ---  ---
___  AAOCSM08  AO6A  SJSC
___  AAOCSM09  AO6A  SJSC
___  AAOCSM10  AO6A  SJSC
___  CSBEMSTR  AO63  SJSE    STARTING    UP    XE  NORM
___  MVS
***** Bottom of data *****
    
```

**Hint**

To see a list of all the parents of an object, use the Browse line command on the Global Overview panel for a specific object (Object Detail Control panel is displayed). Then enter the REQ command; Figure 2-53 on page 2-83 shows the result for object AAOCSM08.

**Figure 2-53 Adding a Parent - Step 7**

```

BMC Software --- Object Requisites for AAOCSM08 - Browse ---- MAINVIEW AutoOPERATOR
COMMAND ==>                                     SCROLL ==> CSR
                                                TGT --- KMZ1

Enter END command to process and return to Detail Control, ENTER to continue

  Parent Object   Group      Description
  AAOCSM10       AO6A      Child of MVS
  MVS            AO6A      The Operating System
*****
***** Bottom of data *****

```

You have now added AAOCSM10 as a parent to object AAOCSM08.

**Step 8** To save this change, issue the **SAVE** primary command on the Global Overview panel.

## Deleting Parents

In this example, parent OS/390 has three children: AAOCSM08, AAOCSM09 and AAOCSM10. AAOCSM10 is also a parent of AAOCSM08. For this procedure, you have decided to delete AAOCSM10 from the list of parents for AAOCSM08.

This process shows how to delete AAOCSM10 from AAOCSM08's list of parents.

To delete object AAOCSM10 from child object AAOCSM08, follow these steps:

**Step 1** Display the Global Overview panel (Figure 2-54).

**Figure 2-54 Global Overview Panel: Deleting a Parent**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>>                                     SCROLL ==>> CSR
                                                    TGT --- AO6A
Primary Commands: EXCeption  Group      SORT      SAVE      ADD      PROFile
                  CAleNDAR  CANcel    UNsort    CONVERT   Build
                  Locate    CMDSHOW  Xref      SAVERULE  SHUTSYS

LC CMDS ----- (S)tart  b(O)nce  rese(T)  depe(N)dents (A)dd
                  sto(P)   (EX)ec  (E)dit   c(L)ients  (D)elete
                  (CAN)cel (M)ove  (R)epeat (B)rowse  s(Y)nc

LC   Object  Group  System  Actual  Desired  Ind  Type
---  -----  -----  -----  -----  -----  ---  ---
___  AAOCSM08 AO6A   SJSC
___  AAOCSM09 AO6A   SJSC
___  AAOCSM10 AO6A   SJSC
___  CSBEMSTR AO63   SJSE    STARTING  UP        XE    NORM
___  MVS
***** Bottom of data *****
    
```

**Step 2** Enter the E(dit) line command next to AAOCSM08.

Figure 2-55 is displayed and shows the two parents of AAOCSM08; OS/390 and AAOCSM10.

**Figure 2-55 Deleting a Parent: Step 2**

```

BMC Software --- Object Requisites for AAOCSM08 - Edit --- MAINVIEW AutoOPERATOR
COMMAND ==>>                                     SCROLL ==>> CSR
                                                    TGT --- AO6A
Primary Commands: ADD END CMDSHOW

LC CMDS ----- (E)dit Object, (D)elete Object

Enter END command to process and return to Detail Control, ENTER to continue

   _____
_   MVS_____   AO6A_____   The Operating System
_   AAOCSM10   AO6A_____   Child of MVS
***** Bottom of data *****
    
```

**Step 3** Enter the D(elete) Object line command next to AAOCSM010.

The line for AAOCSM10 disappears from the panel (see Figure 2-56 on page 2-85).

**Figure 2-56 Deleting a Parent: Step 3**

```

BMC Software --- Object Requisites for AAOCSM08 - Edit --- MAINVIEW AutoOPERATOR
COMMAND ==>                                     SCROLL ==> CSR
                                                TGT --- AO6A

Primary Commands: ADD END CMDSHOW

LC CMDS ----- (E)dit Object, (D)elete Object

Enter END command to process and return to Detail Control, ENTER to continue

LC  Parent Object   Group           Description
--  _____   _____
_   MVS_____   AO6A_____   The Operating System
***** Bottom of data *****
    
```

**Step 4** Press **PF3/END**. The Object Detail Control panel is redisplayed.

**Step 5** Press **PF/3END** again. The Global Overview panel is redisplayed.

**Figure 2-57 Deleting a Parent: Step 5**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     SCROLL ==> CSR
                                                TGT --- AO6A

Primary Commands: EXCeption  Group      SORT      SAVE      ADD      PROFile
                  CALendar   CANcel    UNsort    CONVERT    BUILD
                  Locate     CMDSHOW   Xref      SAVERULE   SHUTSYS

LC CMDS ----- (S)tart   b(O)unce  rese(T)   depe(N)dents  (A)dd
                  sto(P)    (EX)ec   (E)dit    c(L)ients     (D)elete
                  (CAN)cel  (M)ove   (R)peat   (B)rowse      s(Y)nc

LC  Object   Group   System  Actual      Desired      Ind  Type
--  _____
___ AAOCSM08  AO6A   SJSC
___ AAOCSM09  AO6A   SJSC
___ AAOCSM10  AO6A   SJSC
___ CSBEMSTR  AO63   SJSE    STARTING    UP           XE   NORM
___ MVS
***** Bottom of data *****
    
```

**Step 6** To save this change, issue the **SAVE** primary command on the Global Overview panel.

## Adding Servers

Along with modifying the parent-child relationships for an object, it might be necessary to make additions or deletions to an object's list of servers.

A maximum of 45 servers can be entered for each object.

In this example, AAOCSM10 has no server objects assigned to it. AAOCSM09 functions in this system as a server. You have decided to add AAOCSM09 to AAOCSM10's list of servers.

To add server AAOCSM09 to client object AAOCSM10:

**Step 1** Display the Global Overview panel (Figure 2-58).

**Figure 2-58 Global Overview Panel: Adding a Server**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ===>                                     SCROLL ===> CSR
                                                    TGT --- AO6A
                                                    PROFILE
Primary Commands: EXCeption  Group      SORT      SAVE      ADD
                  CALendar   CANcel   UNsort    CONVERT   BUILD
                  Locate     CMDSHOW Xref      SAVERULE  SHUTSYS

LC CMDS ----- (S)tart   b(O)unce  rese(T)   depe(N)dents (A)dd
                  sto(P)    (EX)ec   (E)dit    c(L)ients  (D)elete
                  (CAN)cel (M)ove  (R)peat  (B)rowse   s(Y)nc

LC  Object  Group  System  Actual      Desired      Ind  Type
---  -----  -----  -----  -----  -----  ---  ---
___  AAOCSM08  AO6A   SJSC
___  AAOCSM09  AO6A   SJSC
___  AAOCSM10  AO6A   SJSC
___  CSBEMSTR  AO63   SJSE      STARTING    UP          XE  NORM
___  MVS

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

**Step 2** Enter the E(dit) line command next to AAOCSM10.

The Object Detail Control panel for AAOCSM10 is displayed.

**Figure 2-59 Adding a Server - Step 2**

```

BMC Software ----- Object Detail Control - Edit ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     TGT --- AO6A

The following options are displayed in sequence, or may
be selected by entering the three-character code

    CMD - Command Specifications           EV1 - Event Specifications 1
    EV2 - Event Specifications 2         NOT - User Notification
    GRP - Object Groups                  REQ - Object Prerequisites
    CLS - Client/Server Relationships     SCH - Object Scheduling

Object Name           ==> AAOCSM10       Member of Group      ==> AO6A_____
Object Type           ==> NORM_____    Type of object ( ? for list)

Verify Force Down    ==> N (Y/N)        Restart Only Control ==> N (Y/N)

Application information:
Author               ==> BAKMZ1
Description           ==> Child of MVS_____

Last Modified by BAKMZ1 on 01/30/2001 at 17:36

Enter END command to process and return, ENTER to continue, or CANCEL
    
```

**Step 3** Enter the primary command **CLS** to display the Client/Server Relationships panel for AAOCSM10 (Figure 2-60).

**Figure 2-60 Adding a Server - Step 3**

```

BMC Software -- Client/Server Relationships for AAOCSM10 - Edit -- AutoOPERATOR
COMMAND ==>                                     SCROLL ==> CSR
                                               TGT --- AO6A

Primary Commands: ADD END CMDSHOW

LC CMDS ----- (E)dit server, (D)elete server

Enter END command to process and return to Detail Control, ENTER to continue

LC   Server      Group      Description
                                Connect Command
                                Disconnect Command
***** Bottom of data *****
    
```

**Step 4** Enter the **ADD** primary command.

The Add Client / Server Relationships panel is displayed (see Figure 2-61 on page 2-88).

**Figure 2-61 Adding a Server - Step 4**

```

BMC Software ---- Client/Server Relationships for AAO63 - Add ---- AutoOPERATOR
C | BMC Software      Add Client / Server Relationships      AutoOPERATOR |
  | COMMAND ==>> _____ TGT --- AO6A |
P | |
L | |
  | Server            ==>> _____ |
  | Group Name       ==>> _____ |
E | |
  | Connect Command  ==>> _____ |
  | Disconnect Command ==>> _____ |
L | |
  | Press ENTER to continue processing
  | Press END(PF3) / CAnCel to cancel request
* -----*
    
```

**Step 5** Enter the Server name, the Group Name, and Connect and Disconnect commands for the new server.

Prefix the commands with a pound sign (#) if the commands are to be issued through OS/390.

The completed panel looks like Figure 2-62 on page 2-88.

**Figure 2-62 Adding a Server - Step 6**

```

BMC Software ---- Client/Server Relationships for AAO41 - Add ---- AutoOPERATOR
C | BMC Software      Add Client / Server Relationships      AutoOPERATOR |
  | COMMAND ==>> _____ TGT --- AO6A |
P | |
L | |
  | Server            ==>> AAOCSM09 |
  | Group Name       ==>> AO6A |
E | |
  | Connect Command  ==>> #F AAOCSM09,Connect AAOCSM10 _____ |
  | Disconnect Command ==>> #F AAOCSM09,Disconnect AAOCSM10 _____ |
L | |
  | Press ENTER to continue processing
  | Press END(PF3) / CAnCel to cancel request
* -----*
***** Bottom of data *****
    
```

**Step 6** Press **PF3/END**. The Object Detail Control panel is redisplayed.

**Step 7** Press **PF3/END** again. The Global Overview panel is redisplayed.

**Figure 2-63 Global Overview Panel: Server Added**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     SCROLL ==> CSR
                                                TGT --- AO6A
Primary Commands: EXCeption  Group      SORT      SAVE      ADD      PROFile
                  CALEndar   CANcel    UNsort    CONVERT   Build
                  Locate     CMDSHOW  Xref      SAVERULE  SHUTSYS

LC CMDS ----- (S)tart   b(O)unce  rese(T)   depe(N)de  (A)dd
                  sto(P)    (EX)ec   (E)dit    c(L)ients  (D)delete
                  (CAN)cel  (M)ove   (R)peat   (B)rowse   s(Y)nc

LC   Object   Group   System   Actual   Desired   Ind   Type
---   ---    ---    ---     ---     ---     ---   ---
___  AAOCSM08  AO6A   SJSC
___  AAOCSM09  AO6A   SJSC
___  AAOCSM10  AO6A   SJSC
___  CSBEMSTR  AO63   SJSE    STARTING  UP        XE    NORM
___  MVS
***** Bottom of data *****

```

**Step 8** To view the servers for object AAOCSM10, enter the line command **L** (cLients) next to AAOCSM10. Figure 2-64 is displayed.

**Figure 2-64 Clients of AAOCSM10 Panel**

```

BMC Software ----- Clients of AAOCSM10 ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     TGT --- AO6A

      AAOCSM09  Child of MVS
***** Bottom of data *****

```

**Step 9** Press **PF/3END** to return to the Global Overview panel.

**Step 10** To save this change, issue the **SAVE** primary command on the Global Overview panel.

## Deleting Servers

In this example, AAOCSM10 has the server object AAOCSM09 assigned to it. You have decided to remove AAOCSM09 from AAOCSM10's list of servers.

To delete server AAOCSM09 from object AAOCSM10, follow these steps:

- Step 1** Display the Global Overview panel (Figure 2-65).

**Figure 2-65 Global Overview Panel: Deleting a Server**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     SCROLL ==> CSR
                                                TGT --- AO6A
Primary Commands: EXCeption  Group      SORT      SAVE      ADD      PROFile
                  CALendar   CANcel    UNsort    CONVERT   Build
                  Locate     CMDSHOW  Xref      SAVERULE  SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)   depe(N)dents (A)dd
                  sto(P)    (EX)ec   (E)dit    c(L)ients  (D)elete
                  (CAN)cel (M)ove  (R)epeat  (B)rowse   s(Y)nc

LC  Object  Group  System  Actual  Desired  Ind  Type
___  _____  _____  _____  _____  _____  _____  _____
___  AAOCSM08  AO6A    SJSC
___  AAOCSM09  AO6A    SJSC
___  AAOCSM10  AO6A    SJSC
___  CSBEMSTR  AO63    SJSE    STARTING  UP        XE    NORM
___  MVS
***** Bottom of data *****
    
```

- Step 2** Enter the **E**(dit) line command next to AAOCSM09.

The Object Detail Control panel for AAOCSM09 is displayed (see Figure 2-66 on page 2-91).

**Figure 2-66 Deleting a Server - Step 2**

```

BMC Software ----- Object Detail Control - Edit ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     TGT --- AO6A

The following options are displayed in sequence, or may
be selected by entering the three-character code

    CMD - Command Specifications           EV1 - Event Specifications 1
    EV2 - Event Specifications 2         NOT - User Notification
    GRP - Object Groups                  REQ - Object Prerequisites
    CLS - Client/Server Relationships     SCH - Object Scheduling

Object Name           ==> AAOCSM09           Member of Group      ==> AO6A
Object Type           ==> NORM_____       Type of object ( ? for list)

Verify Force Down    ==> N (Y/N)           Restart Only Control ==> N (Y/N)

Application information:
Author               ==> BAOKMZ1
Description          ==> Child of MVS_____

Last Modified by BAOKMZ1 on 01/30/2001 at 17:36

Enter END command to process and return, ENTER to continue, or CANCEL

```

**Step 3** Enter the primary command CLS to display the Client/Server Relationships panel for AAOCSM09.

Figure 2-67 is displayed and shows that AAOCSM09 is the only server assigned to AAOCSM10.

**Figure 2-67 Deleting a Server - Step 3**

```

BMC Software -- Client/Server Relationships for AAOCSM09 - Edit -- AutoOPERATOR
COMMAND ==>                                     SCROLL ==> CSR
                                               TGT --- AO6A

Primary Commands: ADD END CMDSHOW

LC CMDS ----- (E)dit server, (D)elete server

Enter END command to process and return to Detail Control, ENTER to continue

LC   Server      Group      Description
-----
_   AAOCSM10     AO6A_____
                                     #F AAOCSM09,Connect AAOCSM10_____
                                     #F AAOCSM09,Disconnect AAOCSM10_____

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

```

**Step 4** Enter the D(elete Server) line command next to AAOCSM10.

The line for AAOCSM10 disappears from the panel (Figure 2-68 on page 2-92).

**Figure 2-68 Deleting a Server - Step 4**

```

BMC Software -- Client/Server Relationships for AAOCSM09 - Edit -- AutoOPERATOR
COMMAND ===>                                SCROLL ===> CSR
                                                TGT --- AO6A

Primary Commands: ADD END CMDSHOW

LC CMDS ----- (E)dit server, (D)delete server

Enter END command to process and return to Detail Control, ENTER to continue

LC  Server      Group      Description
                                Connect Command
                                Disconnect Command
***** Bottom of data *****
    
```

**Step 5** Press **PF3/END**. The Object Detail Control panel is redisplayed.

**Step 6** To save this change, issue the **SAVE** primary command on the Global Overview panel.

**Step 7** Press **PF/3END** again. The Global Overview panel is redisplayed.

**Figure 2-69 Global Overview Panel: Server Deleted**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ===>                                SCROLL ===> CSR
                                                TGT --- AO6A

Primary Commands: EXCeption  Group      SORT      SAVE      ADD      PROFile
                  CAleNDAR  CANcel    UNsort    CONVERT    Build
                  Locate     CMDSHOW   Xref      SAVERULE   SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)   depe(N)dents (A)dd
                  sto(P)    (EX)ec   (E)dit    c(L)ients   (D)delete
                  (CAN)cel  (M)ove   (R)epeat  (B)rowse    s(Y)nc

LC  Object      Group      System      Actual      Desired      Ind      Type
-----
___ AAOCSM08     AO6A      SJSC
___ AAOCSM09     AO6A      SJSC
___ AAOCSM10     AO6A      SJSC
___ CSBEMSTR     AO63      SJSE      STARTING    UP           XE      NORM
___ MVS
***** Bottom of data *****
    
```

**Step 8** To view the servers for object AAOCSM10, enter the **C** line command (cLients) next to AAOCSM10. Figure 2-70 on page 2-93 is displayed.

**Figure 2-70 Clients of AAOCSM10 Panel**

```

BMC Software      ----- Clients of AAOCSM10 ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                           TGT --- AO6A

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

```

- Step 9** Press **PF/3END** to return to the Global Overview panel.
- Step 10** To save this change, issue the **SAVE** primary command on the Global Overview panel.

## Modifying an Object's Routine Schedule

The following procedure describes how to schedule object AAOCSM07 to be unavailable (or **DOWN**) on:

- Wednesdays from 3:00 P.M. to 7:00 P.M.
- Fridays from 2:00 A.M. to 3:00 A.M.
- Fridays from 6:00 A.M. to 9:00 A.M.
- Sundays from 3:00 P.M. to 6:00 P.M.

Because an object cannot be available when its parent is unavailable, your schedule for the object also must reflect the down times of its parents.

### Determining the Schedule of the Object's Parent

When you want to schedule the unavailability of an object, begin by determining the unavailability of its parent. The dependent object must be down when the parent is down.

In this case, AAOCSM07 is a child of AAOCSM10.

To determine the schedule for AAOCSM10, follow these steps:

- Step 1** Enter the **B**(rowse) line command next to the object AAOCSM10 (Figure 2-71).

**Figure 2-71 Checking the Schedule of a Parent**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>>                                     SCROLL ==>> CSR
                                                    TGT --- AO6A
Primary Commands: EXCeption  Group      SORT      SAVE      ADD      PROFile
                  CALendar   CANcel   UNSort    CONVERT   BUild
                  Locate     CMDSHOW Xref      SAVERULE  SHUTSYS

LC CMDS ----- (S)tart   b(O)unce  rese(T)   depe(N)dents (A)dd
                  sto(P)    (EX)ec   (E)dit    c(L)ients   (D)elete
                  (CAN)cel  (M)ove  (R)epeat  (B)rowse   s(Y)nc

LC  Object  Group  System  Actual  Desired  Ind  Type
---  -----  -----  -----  -----  -----  ---  -----
___  AAOCSM07  AO6A   SJSC
___  AAOCSM08  AO6A   SJSC
___  AAOCSM09  AO6A   SJSC
B_  AAOCSM10  AO6A   SJSC
___  CSBEMSTR  AO63   SJSE    STARTING  UP        XE  NORM
___  MVS
***** Bottom of data *****

```

The Object Detail Control - Browse panel is displayed (Figure 2-72 on page 2-95).

**Figure 2-72 Checking the Schedule of a Parent: Step 1**

```

BMC Software ----- Object Detail Control - Browse ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                                    TGT --- AO6A

The following options are displayed in sequence, or may
be selected by entering the three-character code

    CMD - Command Specifications          EV1 - Event Specifications 1
    EV2 - Event Specifications 2         NOT - User Notification
    GRP - Object Groups                  REQ - Object Prerequisites
    CLS - Client/Server Relationships     SCH - Object Scheduling

Object Name          ==> AAOCSM10          Member of Group      ==> AO6A
Object Type          ==> NORM_____      Type of object ( ? for list)

Verify Force Down    ==> N (Y/N)          Restart Only Control ==> N (Y/N)

Application information:
Author              ==> BAOKMZ1
Description         ==> Child of MVS_____

Last Modified by BAOKM1M on 01/30/2001 at 17:36

Enter END command to process and return, ENTER to continue, or CAnCel
    
```

**Step 2** Enter the **SCH** primary command to display the Object Scheduling panel for AAOCSM10.

Figure 2-73 shows that AAOCSM10 is scheduled to be **DOWN** on Mondays from 3:00 P.M. to 5:00 P.M.

**Figure 2-73 Checking the Schedule of a Parent: Step 2**

```

BMC Software -- Object Scheduling for AAOCSM10 - Browse -- MAINVIEW AutoOPERATOR
COMMAND ==>                                                    SCROLL ==> CSR
                                                                TGT --- AO6A

Primary Commands: ADD CAnCel END CMDSHOW

LC CMDS ----- (E)dit Time, (D)elete Time

Enter END to process/return to Detail Control, ENTER to continue, or CAnCel

LC Object          Day          Down From          Down To
  Group              HH:MM              HH:MM
___ AAOCSM10      MONDAY              15:00              17:00
***** Bottom of data *****
    
```

**Step 3** Make a note of this time and press **PF3/END** to leave this panel and return to the Object Detail Control panel.

- Step 4** Press **PF3/END** again to return to the Global Overview panel.
- Step 5** Go to “Modifying the Object’s Schedule” for instructions about creating a schedule for AAOCSM07.

## Modifying the Object’s Schedule

Now you know that the parent of AAOCSM07 is DOWN on Mondays from 3:00 P.M. to 5:00 P.M. You also want to schedule AAOCSM07 to be DOWN at the following times:

- Wednesdays from 3:00 P.M. to 7:00 P.M.
- Fridays from 2:00 A.M. to 3:00 A.M.
- Fridays from 6:00 A.M. to 9:00 A.M.
- Sundays from 3:00 P.M. to 6:00 P.M.

To avoid scheduling conflicts with the parent (AAOCSM10), you must also include Monday 3:00 P.M. to 5:00 P.M. in this list. If you do not do this preplanning step, you will encounter a scheduling conflict. For information about scheduling conflicts, refer to “Dealing with Scheduling Conflicts at Modification” on page 2-104.

To modify AAOCSM07’s schedule, follow these steps:

- Step 1** From the CSM Global Overview panel, enter the **E(dit)** line command next to AAOCSM07.

The Object Detail Control panel for AAOCSM07 is displayed (Figure 2-74 on page 2-97).

**Figure 2-74 Modifying an Object Schedule - Step 1**

```

BMC Software ----- Object Detail Control - Edit ----- MAINVIEW AutoOPERATOR
COMMAND ==>>>                                     TGT --- AO6A

The following options are displayed in sequence, or may
be selected by entering the three-character code

    CMD - Command Specifications           EV1 - Event Specifications 1
    EV2 - Event Specifications 2         NOT - User Notification
    GRP - Object Groups                  REQ - Object Prerequisites
    CLS - Client/Server Relationships     SCH - Object Scheduling

Object Name           ==>> AAOCSM07           Member of Group      ==>> AO6A
Object Type           ==>> NORM_____       Type of object ( ? for list)

Verify Force Down    ==>> N (Y/N)           Restart Only Control ==>> N (Y/N)

Application information:
Author               ==>> BAOKMZ1
Description          ==>> Child of MVS_____

Last Modified by BAOKMZ1 on 01/30/2001 at 17:36

Enter END command to process and return, ENTER to continue, or CANCEL
    
```

**Step 2** Enter the primary command **SCH** to display the Object Scheduling panel for AAOCSM07 (Figure 2-75).

**Figure 2-75 Modifying an Object Schedule - Step 2**

```

BMC Software --- Object Scheduling for AAOCSM07 - Edit --- MAINVIEW AutoOPERATOR
COMMAND ==>>>                                     SCROLL ==>> CSR
                                                    TGT --- AO6A

Primary Commands: ADD CANCEL END CMDSHOW

LC CMDS ----- (E)dit Time, (D)elete Time

Enter END to process/return to Detail Control, ENTER to continue, or CANCEL

LC Object          Day          Down From          Down To
  Group                               HH:MM              HH:MM

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

**Step 3** Enter the schedule for AAOCSM07's DOWN times using the **ADD** primary command. Figure 2-76 on page 2-98 is displayed.

**Figure 2-76 Modifying an Object Schedule - Step 3**

```

BMC Software --- Object Scheduling for AAOCSSM07 - Edit --- MAINVIEW AutoOPERATOR
C | BMC Software          Add Object Scheduling          AutoOPERATOR
  | COMMAND ===> _____ TGT --- AO6A
P |                                     DATE --- 01/01/30
  |                                     TIME --- 19:27:12
L |
  | Day Selections: MON, TUE, WED, THU, FRI, SAT, SUN or WKD, WKN, DLY
E |
  | Group Name          ===> _____
L |
  | Day                 ===> _____
  | Down From (HH:MM) ===> __ : __
* | Down To   (HH:MM) ===> __ : __
  |
  | Press ENTER to continue processing
  | Press END(PF3) / CANCEL to cancel request
  |-----|
  
```

**Step 4** Use 24-hour clock format to enter all times. Repeat until all DOWN times are added.

Figure 2-77 on page 2-99 shows the Object scheduling panel with all the DOWN times filled in.

Note that the DOWN time for parent object AAOCSSM10 is also included. (Monday from 3:00 P.M. to 5:00 P.M.)

**Figure 2-77 Modifying an Object Schedule - Step 4**

```

BMC Software --- Object Scheduling for AAOCSM07 - Edit --- MAINVIEW AutoOPERATOR
COMMAND ==>                                     SCROLL ==> CSR
                                                TGT --- AO6A

Primary Commands: ADD CANCEL END CMDSHOW

LC CMDS ----- (E)dit Time, (D)elete Time

Enter END to process/return to Detail Control, ENTER to continue, or CANCEL

LC Object          Day          Down From          Down To
  Group                HH:MM             HH:MM
- AO6A_____ FRIDAY_____ 02:00             03:00
- AO6A_____ FRIDAY_____ 06:00             09:00
- AO6A_____ MONDAY_____ 15:00             17:00
- AO6A_____ SUNDAY_____ 15:00             19:00
- AO6A_____ WEDNESDAY 15:00             19:00
***** Bottom of data *****

```

AAOCSM07 will be DOWN:

- Fridays from 2:00 A.M. to 3:00 A.M.
- Fridays from 6:00 A.M. to 9:00 A.M.
- Mondays from 3:00 P.M. to 5:00 P.M. (because AAOCSM07's parent, AAOCSM10 is down at that time)
- Sundays from 3:00 P.M. to 6:00 P.M.
- Wednesdays from 3:00 P.M. to 7:00 P.M.

**Step 5** Press **PF3/END**. The Object Detail Control panel is redisplayed.

**Step 6** Press **PF/3END** again. The Global Overview panel is redisplayed (Figure 2-78 on page 2-100).

**Figure 2-78 Global Overview Panel: Schedule Modified**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>>                                     SCROLL ==>> CSR
                                                TGT --- AO6A
Primary Commands: EXCeption  Group      SORT      SAVE      ADD      PROFile
                  CALEndar   CANcel    UNSort    CONVERT   BUild
                  Locate     CMDSHOW  Xref      SAVERULE  SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)   depe(N)dents  (A)dd
                  sto(P)    (EX)ec   (E)dit    c(L)ients    (D)elete
                  (CAN)cel  (M)ove   (R)epeat  (B)rowse     s(Y)nc

LC   Object   Group   System   Actual       Desired       Ind   Type
---   -----   ---     ---     -----     -----     ---   ---
___  AAOCSM07  AO6A    SJSC
___  AAOCSM08  AO6A    SJSC
___  AAOCSM09  AO6A    SJSC
___  AAOCSM10  AO6A    SJSC
___  CSBEMSTR  AO63    SJSE     STARTING     UP            XE    NORM
___  MVS
***** Bottom of data *****

```

**Step 7** To save the changes, enter the SAVE primary command.

Changes take effect when you enter the Build primary command for the group.

## Modifying an Object's Schedule with CSM Schedule Tags

CSM includes three predefined schedules that you can use for scheduling objects:

- **WEEKDAY (WKD)**  
 specifies that the object should be DOWN at a specified time on Mondays through Fridays every week
- **WEEKEND (WKN)**  
 specifies that the object should be DOWN at a specified time on every Saturday and Sunday
- **DAILY (DLY)**  
 specifies that an object should be DOWN at a specified time every day of the week (Monday through Sunday) every week

These tags also can be used in conjunction with schedules for a specific day and time. CSM will take the object down on the specific day in addition to the time specified by any of these tags.

For example, suppose object AAOCSM09 should be DOWN every day from 2:00 A.M. to 3:00 A.M.

To modify AAOCSM09's schedule, follow these steps:

**Step 1** Enter the **E**(dit) line command next to AAOCSM09.

The Object Detail Control panel for AAOCSM09 is displayed.

**Figure 2-79 Using Object Schedule Tags**

```

BMC Software ----- Object Detail Control - Edit ----- MAINVIEW AutoOPERATOR
COMMAND ==>>>                                     TGT --- AO6A

The following options are displayed in sequence, or may
be selected by entering the three-character code

      CMD - Command Specifications           EV1 - Event Specifications 1
      EV2 - Event Specifications 2         NOT - User Notification
      GRP - Object Groups                  REQ - Object Prerequisites
      CLS - Client/Server Relationships    SCH - Object Scheduling

Object Name           ==>> AAOCSM09           Member of Group      ==>> AO6A
Object Type           ==>> NORM_____       Type of object ( ? for list)

Verify Force Down    ==>> N (Y/N)           Restart Only Control ==>> N (Y/N)

Application information:
Author               ==>> BAOKMZ1
Description          ==>> Child of MVS_____

Last Modified by BAOKMZ1 on 01/30/2001 at 17:36

Enter END command to process and return, ENTER to continue, or CANCEL

```

**Step 2** Enter the primary command **SCH** to display the Object Scheduling panel for AAOCSM09 (Figure 2-80 on page 2-102).

**Figure 2-80 Using Object Schedule Tags - Step 3**

```

BMC Software --- Object Scheduling for AAOCSM09 - Edit --- MAINVIEW AutoOPERATOR
COMMAND ==>>>                                SCROLL ==>> CSR
                                                TGT --- AO6A

Primary Commands: ADD CANCEL END CMDSHOW

LC CMDS ----- (E)dit Time, (D)elete Time

Enter END to process/return to Detail Control, ENTER to continue, or CANCEL

LC Object      Day      Down From      Down To
  Group                HH:MM          HH:MM

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

**Step 3** Enter the primary command **ADD**. Figure 2-81 is displayed.

**Figure 2-81 Modifying an Object Schedule - Step 4**

```

BMC Software --- Object Scheduling for AAOCSM09 - Edit --- MAINVIEW AutoOPERATOR
C | BMC Software          Add Object Scheduling          AutoOPERATOR |
  | COMMAND ==>>> _____ TGT --- AO6A |
P |                                     DATE --- 01/01/30 |
  |                                     TIME --- 19:27:12 |
L |                                     |
  | Day Selections: MON, TUE, WED, THU, FRI, SAT, SUN or WKD, WKN, DLY |
E |                                     |
  | Group Name           ==>>> _____ |
L |                                     |
  | Day                  ==>>> _____ |
  | Down From (HH:MM) ==>>> __ : __ |
* | Down To (HH:MM) ==>>> __ : __ | *
  |                                     |
  | Press ENTER to continue processing |
  | Press END(PF3) / CANCEL to cancel request |
  |-----|
    
```

**Step 4** Enter the word **DLY** to specify that this object is supposed to be **DOWN** every day (Monday through Sunday), and enter the times using 24-hour clock format (**02:00** and **03:00**), as shown in Figure 2-82 on page 2-103.

**Figure 2-82 Using Object Schedule Tags - Step 5**

```

BMC Software ----- Object Scheduling for AAOCSM09 - Edit ----- INVALID TIME
C | BMC Software          Add Object Scheduling          AutoOPERATOR |
  | COMMAND ===> _____ TGT --- AO6A              |
P |                                     DATE --- 01/01/30   |
  |                                     TIME --- 19:27:12   |
L |                                     |
  | Day Selections: MON, TUE, WED, THU, FRI, SAT, SUN or WKD, WKN, DLY |
E |                                     |
  | Group Name          ===> AO6A_____                |
L |                                     |
  | Day                 ===> DLY_____                |
  | Down From (HH:MM) ===> 02 : 00                    |
* | Down To   (HH:MM) ===> 03 : 00                    | *
  |                                     |
  | Press ENTER to continue processing                 |
  | Press END(PF3) / CANcel to cancel request         |
  |-----|
    
```

**Step 5** You can enter additional DOWN times by using the **ADD** primary command. Press **PF3/END** to redisplay the Object Scheduling panel.

**Step 6** Press **PF3/END**. The Object Detail Control panel is redisplayed.

**Step 7** Press **PF3/END** again. The Global Overview panel is redisplayed.

**Figure 2-83 Global Overview Panel: Schedule Modified**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ===>                                     SCROLL ===> CSR
                                                TGT --- AO6A
Primary Commands: EXception  Group      SORT   SAVE   ADD   PROFile
                  CALendar  CANcel    UNsort CONVERT BUild
                  Locate    CMDSHOW Xref   SAVERULE SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)  depe(N)dents  (A)dd
                  sto(P)   (EX)ec   (E)dit   c(L)ients    (D)elete
                  (CAN)cel (M)ove   (R)epeat (B)rowse    s(Y)nc

LC  Object  Group  System  Actual  Desired  Ind  Type
---  -----  ----  -----  -----  -----  ---  ----
___ AAOCSM07  AO6A  SJSC
___ AAOCSM08  AO6A  SJSC
___ AAOCSM09  AO6A  SJSC
___ AAOCSM10  AO6A  SJSC
___ CSBEMSTR  AO63  SJSE  STARTING  UP        XE  NORM
___ MVS
***** Bottom of data *****
    
```

**Step 8** To save the changes, enter the **SAVE** primary command.

**Step 9** Enter the primary command **BU** (BUild) on the **COMMAND** line.

The changes take effect when the **BUILD** command is issued for the group that this object is associated with. CSM issues an error message if it detects a schedule conflict. It is important to remember that it is not a conflict to have two identical schedule entries, each for a different group.

## Dealing with Scheduling Conflicts at Modification

If an object is scheduled to be UP when its parent is DOWN, CSM issues a short error message in the upper right corner when you try to exit from the scheduling panel.

**Step 1** For example, suppose you have completed the schedule panel for AAOCSM07, as shown in Figure 2-84, where the object will be DOWN on Wednesdays, Fridays, and Sundays.

**Figure 2-84 Schedule Conflicts: Example 1**

```

BMC Software --- Object Scheduling for AAOCSM07 - Edit --- MAINVIEW AutoOPERATOR
COMMAND ==>>>                                     SCROLL ==>>> CSR
                                                    TGT --- AO6A

Primary Commands: ADD CANcel END CMDSHOW

LC CMDS ----- (E)dit Time, (D)elete Time

Enter END to process/return to Detail Control, ENTER to continue, or CANcel

LC Object      Day      Down From      Down To
  Group                HH:MM          HH:MM
- AO6A_____ FRIDAY_____ 02:00          03:00
- AO6A_____ FRIDAY_____ 06:00          09:00
- AO6A_____ SUNDAY_____ 15:00          19:00
- AO6A_____ WEDNESDAY 15:00          19:00
***** Bottom of data *****
    
```

Now suppose you did not perform the task described in “Determining the Schedule of the Object’s Parent” on page 2-93 and you did not know that AAOCSM07’s parent (AAOCSM10) is scheduled to be DOWN on Mondays from 3:00 P.M. to 5:00 P.M.

**Step 2** When you press **PF3/END** to save this schedule, you will receive a short error message as shown in the upper right corner of Figure 2-85 on page 2-105.

**Figure 2-85 Schedule Conflicts: Example 2**

```

BMC Software ----- Object Scheduling for AAOCSM07 ---- Mismatch on Monday
COMMAND ==>                                     SCROLL ==> CSR
                                                TGT --- AO6A

Primary Commands: ADD CANCEL END CMDSHOW

LC CMDS ----- (E)dit Time, (D)elete Time

Enter END to process/return to Detail Control, ENTER to continue, or CANCEL

LC Object          Day          Down From          Down To
  Group                HH:MM            HH:MM
- AO6A_____ FRIDAY_____ 02:00            03:00
- AO6A_____ FRIDAY_____ 06:00            09:00
- AO6A_____ SUNDAY_____ 15:00            19:00
- AO6A_____ WEDNESDAY 15:00            19:00
***** Bottom of data *****

```

**Step 3** Press **PF1** to see the entire error message.

The conflict is described as shown in Figure 2-86.

**Figure 2-86 Schedule Conflicts: Example 3**

```

BMC Software ----- Object Scheduling for AAOCSM07 ---- Mismatch on Monday
COMMAND ==>                                     SCROLL ==> CSR
                                                TGT --- KMZ1

Primary Commands: ADD CANCEL END CMDSHOW

AAOCSM07 is up at 15:00; but AAOCSM10 is down
LC CMDS ----- (E)dit Time, (D)elete Time

Enter END to process/return to Detail Control, ENTER to continue, or CANCEL

LC Object          Day          Down From          Down To
  Group                HH:MM            HH:MM
- AO6A_____ FRIDAY_____ 02:00            03:00
- AO6A_____ FRIDAY_____ 06:00            09:00
- AO6A_____ SUNDAY_____ 15:00            19:00
- AO6A_____ WEDNESDAY 15:00            19:00
***** Bottom of data *****

```

**Step 4** To leave this panel, you must delete all the entries that you just entered for AAOCSM07.

**Step 5** Press **PF3/END** to return to the CSM panel.

**Step 6** To successfully enter the schedule for AAOCSM07, you must perform the procedure described in “Determining the Schedule of the Object’s Parent” on page 2-93, note all the times the parent AAOCSM10 is DOWN, and then perform the steps in “Modifying the Object’s Schedule” on page 2-96.

**Hint**  
 As a general practice, to avoid these types of conflicts, BMC Software recommends that you always schedule the children first and then create schedules for the parents.

## Overriding the Calendar for Special Situations

The following procedure describes how to override the Calendar to schedule object AAOCSM07 to be unavailable (or DOWN) on

03/01/97 from 06:00 P.M. to 12:00 A.M.

To access the Global Calendar Override panel, follow these steps:

**Step 1** Issue the **CAL(endar)** primary command from the CSM Global Overview panel.

The Global Calendar Override panel is displayed (Figure 2-87).

**Figure 2-87 Global Calendar Override Panel**

```

BMC Software ----- Global Calendar Override ----- MAINVIEW AutoOPERATOR
COMMAND ==>>>                                     TGT --- KMZ1

Primary Commands: ADD CANcel END CMDSHOW             DATE --- 01/30/2001
                                                       TIME --- 19:49

LC CMDS --- (I)nsert Time, (D)elete Time

Press END to process and return, CANcel to exit without changes.

LC Object      Group /      Calendar      Down      Reason for Down Time /
                SYSNAME      Date      From      To      Contact Name
                mm/dd/yy  HH:MM      HH:MM
-----
    
```

The Global Calendar Override functions under the control of a specific calendar date. The override is only for the date specified in this panel; therefore, the affected objects do not have to be restored to their normal schedules after the override takes place.

A Global Calendar Override entry is associated with a particular object within a group. The override takes effect after you press **PF3 (END)**.

**Step 2** Enter the following required information:

- Object's name (**Name** field)
- Group name
- Calendar date (**Calendar** field)
- Down times (**From** and **To** fields)

You also can tell the reason for the downtime and the contact name of the person who asked for it.

If you do not fill in the required information, CSM prompts you in the upper right corner when you press **PF3/END**.

Figure 2-88 shows an example of a Global Calendar Override panel that is filled in.

**Figure 2-88 Example of Global Calendar Override for Object AAOCSM07**

```

BMC Software ----- Global Calendar Override ----- MAINVIEW AutoOPERATOR
COMMAND ==>>>                                     TGT --- KMZ1

Primary Commands: ADD CANCEL END CMDSHOW             DATE --- 01/30/2001
                                                    TIME --- 19:49

LC CMDS --- (I)nsert Time, (D)elete Time

Press END to process and return, CANCEL to exit without changes.

LC Object      Group /      Calendar      Down      Reason for Down Time /
                SYSNAME      Date         From       To         Contact Name
                mm/dd/yy  HH:MM      HH:MM
_   AAOCSM07   A06A          01/12/01   18:00     24:00     Backup copies

*****BOTTOM OF DATA *****
    
```

**Step 3** Press **PF3/END** to save this Global Calendar Override.

# Enabling Cross-System Object Management

To perform cross-system object management, ensure that BBI-SS PAS to BBI-SS PAS communications is active between BBI-SS PASs where CSM is installed and operating.

The following procedure describes how to enable cross-system object monitoring and management.

To enable cross-system monitoring, you must edit the OS/390 object:

- Step 1** To edit the OS/390 object, enter E(dit) in the line command column next to OS/390 on the Global Overview panel (Figure 2-89).

**Figure 2-89 Editing the OS/390 Object to Enable Cross-System Management**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>>>                                     SCROLL ==>> CSR
                                                    TGT --- AO6A
Primary Commands: EXCeption  Group      SORT      SAVE      ADD      PROFile
                  CALendar   CANcel   UNsort   CONVERT   BUild
                  Locate     CMDSHOW Xref     SAVERULE  SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)  depe(N)dents  (A)dd
                  sto(P)    (EX)ec   (E)dit   c(L)ients    (D)elete
                  (CAN)cel  (M)ove  (R)epeat (B)rowse     s(Y)nc

LC  Object  Group  System  Actual  Desired  Ind  Type
---  -----  -----  -----  -----  -----  ---  ---
___  AAOCSM07  AO6A   SJSC   DOWN    UP        XE  NORM
___  AAOCSM09  AO6A   SJSC   STARTING  UP        XE  NORM
___  AAOCSM10  AO6A   SJSC   STARTING  UP        NXE  NORM
___  AO6B      AO6A   SJSC   UNKNOWN  UNKNOWN   XE  NORM
___  CSBEMSTR  AO63   SJSE   UP        UP        XE  NORM
_E_ OS/390
***** Bottom of data *****
    
```

The Object Detail Control - Edit panel for the OS/390 object is displayed (Figure 2-90 on page 2-109).

**Figure 2-90 Object Detail Control - Edit Panel for OS/390**

```

BMC Software ----- Object Detail Control - Edit ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                           TGT --- AO6A

The following options are displayed in sequence, or may
be selected by entering the three-character code

    CMD - Command Specifications          EV1 - Event Specifications 1
    EV2 - Event Specifications 2         NOT - User Notification

Object Name          ==> MVS              Enable PLEX          ==> Y_____
Object Type          ==> NORM            Type of object ( ? for list)

Verify Force Down   ==> N (Y/N)         Restart Only Control ==> N (Y/N)

Application information:
Author              ==> BMC
Description         ==> The Operating System_____

Last Modified by BMC      on 05/03/2000 at 00:00

Enter END command to process and return, ENTER to continue, or CANCEL

```

The Object Detail Control - Edit panel for the OS/390 panel includes the field **Enable PLEX** ==> **Y**, where the default setting is YES.

- Step 2** To enable cross-system management, ensure that Y is entered in this field.
- Step 3** Press **PF3/END**.
- Step 4** To complete enabling cross-system management, you must repeat this process on the other BBI-SS PAS targets.

## Performing Routine Object Tasks

The following section describes routine tasks you can perform on CSM objects from the Global Overview panel:

<b>Task</b>	<b>Pages</b>
starting objects	2-110
stopping objects	2-114
bouncing objects	2-116
resetting objects	2-119
canceling objects	2-122
moving objects	2-125
scheduling an EXEC associated with a CSM ALERT	2-128

### Starting an Object

The purpose of CSM is to help manage the availability of objects in your system, which is achieved by performing automatic starts and stops of objects, based on a defined schedule. Sometimes, however, it is necessary to start an object outside of its schedule.

The following procedure shows how to start an object from the Global Overview panel when the object's Actual and Desired states are DOWN.

**Step 1** Display the Global Overview panel (Figure 2-91).

**Figure 2-91 Global Overview Panel: Starting an Object**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     SCROLL ==> CSR
                                                TGT --- AO6A
Primary Commands: EXception  Group      SORT      SAVE      ADD      PROFile
                  CALendar  CANcel    UNSort    CONVERT   BUild
                  Locate    CMDSHOW Xref      SAVERULE  SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)  depe(N)dents (A)dd
                 sto(P)   (EX)ec   (E)dit   c(L)ients  (D)delete
                 (CAN)cel (M)ove   (R)peat  (B)rowse  s(Y)nc

LC  Object  Group  System  Actual      Desired      Ind  Type
---  -----  ----  -----  -----  -----  ---  ----
___  AAOCSM07 AO6A   SJSC    DOWN        DOWN         ___  NORM
___  AAOCSM09 AO6A   SJSC    STARTING    UP           XE   NORM
___  AAOCSM10 AO6A   SJSC    STARTING    UP           NXE  NORM
___  CSBEMSTR AO63   SJSE    STARTING    UP           XE   NORM
___  MVS
***** Bottom of data *****
  
```

**Step 2** Enter the S(start) line command next to AAOCSM07.

Figure 2-92 is displayed.

**Figure 2-92 Object Start Confirmation Panel**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
C | BMC Software          Confirm: Start for AAOCSM07          AutoOPERATOR |
  | COMMAND ==>                                     TGT --- AO6A          | A
P |
  |
  | Please confirm START for AAOCSM07   ? N (Y/N)
  | START all PARENTS of AAOCSM07      ? Y (Y/N)
  | START now, but STOP as scheduled    ? Y (Y/N)
  |
  |
  | Press ENTER to continue or END / CANcel to cancel request
  |
  | MVS
  | ***** Bottom of data *****
  
```

**Step 3** Reply to the start confirmation panel; the following list describes the replies that you can enter on this panel:

- To start AAOCSM07, enter **Y** for **Please confirm START for AAOCSM07**.

If you accept the default setting of **N**, AAOCSM07 will not start.

- To start all the parents of object AAOCSM07, accept the default setting of **Y** for **START all PARENTS of AAOCSM07**.

If you enter **N**, CSM attempts to start only AAOCSM07.

- To start the object and accept the predefined stop time for AAOCSM07, accept the default setting of **Y** for **START now, but STOP as scheduled**. CSM starts the object now.

After the object is started, CSM stops the object at the scheduled time. The object remains under the control of CSM and has an actual state of **UP** and a desired state of **UPEARLY**.

If you enter **N**, CSM starts the object, ignores the schedule, and does not trigger a shutdown at the scheduled time. The object will show an actual state of **UP** and a desired state of **UPFORCE**.

**Step 4** Press **Enter**.

Figure 2-93 on page 2-113 is displayed.

**Figure 2-93 Global Overview Panel: Object Started**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>>>                                     SCROLL ==>>> CSR
                                                    TGT --- AO6A
Primary Commands: EXception  Group      SORT      SAVE      ADD      PROFile
                  CALendar   CANcel   UNsort    CONVERT   BUild
                  Locate     CMDSHOW Xref      SAVERULE  SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)  depe(N)dents  (A)dd
                  sto(P)    (EX)ec   (E)dit   c(L)ients    (D)elete
                  (CAN)cel  (M)ove  (R)peat  (B)rowse     s(Y)nc

LC  Object  Group  System  Actual      Desired      Ind  Type
---  -----  -----  -----  -----  -----  ---  ---
___  AAOCSM07 AO6A   SJSC    STARTING    UPEARLY
___  AAOCSM09 AO6A   SJSC    STARTING    UP          XE   NORM
___  AAOCSM10 AO6A   SJSC    STARTING    UP          NXE  NORM
___  CSBEMSTR AO63   SJSE    STARTING    UP          XE   NORM
___  MVS
***** Bottom of data *****

```

The Global Overview now shows that the actual state of AAOCSM07 is STARTING and the desired state is UPEARLY. As soon as the object starts, the actual state will be UP.

**Hint**

On rare occasions, CSM issues ALERT ACM720A indicating that the start command has timed out for an object that has just started. A check of the SYSLOG shows that the object indeed started. The ALERT, however, remains. The object's state variable contains the value STARTING. Use the s(Y)nc line command to synchronize the object's state variable with the actual state of the object.

If the parent objects of the affected object are not started by CSM, it is usually due to the object's Start Rule not firing. Make sure the Start Rule has fired. If it has not, you must determine why and correct the problem for the next startup of this object.

## Stopping an Object

For many of the same reasons for unexpectedly starting an object, an object also might have to be stopped from the CSM Global Overview. This section explains how to stop an object.

**Step 1** Display the Global Overview panel (Figure 2-94).

**Figure 2-94 Global Overview Panel: Stopping an Object**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>>                                     SCROLL ==>> CSR
                                                    TGT --- AO6A
Primary Commands: EXception  Group      SORT      SAVE      ADD      PROFile
                  CALendar   CANcel    UNSort    CONVERT   BUild
                  Locate     CMDSHOW  Xref      SAVERULE  SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)  depe(N)dents  (A)dd
                  sto(P)    (EX)ec   (E)dit   c(L)ients  (D)etele
                  (CAN)cel  (M)ove   (R)epeat (B)rowse   s(Y)nc

LC  Object  Group  System  Actual  Desired  Ind  Type
---  -----  -----  -----  -----  -----  ---  -----
___  AAOCSM07  AO6A   SJSC   DOWN    UP        ___  NORM
___  AAOCSM09  AO6A   SJSC   UP       UP        XE  NORM
___  AAOCSM10  AO6A   SJSC   STARTING UP       NXE  NORM
___  CSBEMSTR  AO63   SJSE   STARTING UP       XE  NORM
___  MVS
***** Bottom of data *****
    
```

**Step 2** Enter the P (stoP) line command next to AAOCSM09.

Figure 2-95 on page 2-115 is displayed.

**Figure 2-95 Object Stop Confirmation Panel**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
C | BMC Software          Confirm: Stop for AAOCSM09          AutoOPERATOR |
  | COMMAND ===>                               TGT --- AO6A          | A
P |
  |
  | Please confirm STOP for AAOCSM09      ? N (Y/N)
L |
  | STOP all DEPENDENTS of AAOCSM09      ? Y (Y/N)
  |
  | STOP now, but START as scheduled      ? N (Y/N)
L |
  |
  | Press ENTER to continue or END / CANCEL to cancel request
  |
  | MVS
  | ***** Bottom of data *****

```

**Step 3** Reply to the stop confirmation panel; the following list describes the replies that you can enter on this panel:

- To stop AAOCSM09, enter **Y** for **Please confirm STOP for AAOCSM09**.

If you accept the default setting of **N**, the object is not stopped.

- To stop all dependents of object AAOCSM09, accept the default setting of **Y** for **STOP all DEPENDENTS of AAOCSM09**. CSM stops all dependent objects of AAOCSM09 before stopping AAOCSM09.

If you enter **N**, CSM stops only AAOCSM09.

- To stop the object and ignore the scheduled start time for AAOCSM09, accept the default setting of **N** for **STOP now, but START as scheduled**.

CSM stops the object now but does not restart at the scheduled restart time. The object is now outside of the control of CSM and has an actual state of **DOWN** and a desired state of **DOWNFORCE**.

If you enter **Y**, CSM stops the object, maintains control, and triggers restart at the scheduled time. The actual state of AAOCSM09 will be **DOWN** and the desired state will be **DOWNEARLY**.

**Step 4** Press **Enter**.

Figure 2-96 on page 2-116 is displayed.

**Figure 2-96 Global Overview Panel: Object Stopped**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>>>                                SCROLL ==>> CSR
                                           TGT --- AO6A
Primary Commands: EXCeption  Group      SORT      SAVE      ADD      PROFile
                  CAleNDAR  CANcel   UNsort    CONVERT   BUild
                  Locate    CMDSHOW Xref      SAVERULE  SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)  depe(N)dents (A)dd
                  sto(P)   (EX)ec   (E)dit   c(L)ients  (D)elete
                  (CAN)cel (M)ove  (R)peat  (B)rowse   s(Y)nc

-----
___ AAOC SM07 AO6A  SJSC   DOWN      UP          NORM
___ AAOC SM09 AO6A  SJSC   STOPPING  DOWNFORCE  XE   NORM
___ AAOC SM10 AO6A  SJSC   STARTING  UP          NXE  NORM
___ CSBEMSTR AO63  SJSE   STARTING  UP          XE   NORM
___ MVS
***** Bottom of data *****
    
```

The Global Overview now shows that the actual state of AAOC SM09 is STOPPING and the desired state is DOWNFORCE. As soon as the object stops, the actual state will be DOWN.

<p><b>Hint</b></p> <p>On rare occasions, CSM issues ALERT ACM725A, indicating that the stop command has timed out for an object that has just stopped. A check of the SYSLOG shows that the object indeed stopped. The ALERT, however, remains. The object's state variable contains the value STOPPING.</p> <p>If children objects of the affected object are not stopped by CSM, it is usually due to the object's Stop Rule not firing. Make sure the Stop Rule has fired. If it has not, you must determine why and correct the problem for the next shutdown of this object.</p>
---

## Bouncing an Object

Occasionally, an object might need to be recycled (stopped then immediately started). CSM simplifies this task with the Bounce function.

The Bounce line command causes a controlled, unscheduled termination of the object by triggering the object's stop command. During this stoppage, the object's desired state never changes from UP which causes CSM to restart the object immediately after the object's Stop Rule is fired.

The following procedure shows how to bounce an object (AAOCSM10):

**Step 1** Display the Global Overview panel (Figure 2-97).

**Figure 2-97 Global Overview Panel: Bouncing an Object**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     SCROLL ==> CSR
                                                TGT --- AO6A
Primary Commands: EXception  Group      SORT   SAVE    ADD      PROFILE
                  CALendar   CANcel   UNSort  CONVERT BUILD
                  Locate     CMDSHOW Xref    SAVERULE SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)  depe(N)dents  (A)dd
                  sto(P)   (EX)ec   (E)dit   c(L)ients    (D)elete
                  (CAN)cel (M)ove   (R)epeat (B)rowse     s(Y)nc

LC  Object  Group  System  Actual      Desired      Ind  Type
---  -----  -----  -----  -----  -----  ---  ---
___  AAOCSM07 AO6A   SJSC    DOWN        UP            XE  NORM
___  AAOCSM09 AO6A   SJSC    STOPPING    DOWNFORCE    XE  NORM
___  AAOCSM10 AO6A   SJSC    UP          UP            NXE  NORM
___  CSBEMSTR AO63   SJSE    STARTING    UP            XE  NORM
___  MVS
***** Bottom of data *****

```

**Step 2** Enter the **O** (bOunce) line command next to AAOCSM10.

Figure 2-98 on page 2-118 is displayed.

**Figure 2-98 Confirm Bounce Panel**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
C | BMC Software          Confirm: Bounce for AAOCSM10          AutoOPERATOR |
  | COMMAND ==>>>                                TGT --- A06A          | A
P |
  |
  | Please confirm Bounce for AAOCSM10    ? N (Y/N)
L |
  |
  |
L |
  |
  | Press ENTER to continue or END / CANCEL to cancel request
  |-----|
  | CSBEMSTR  A063      SJSE      STARTING      UP          XE      NORM
  | MVS
  | ***** Bottom of data *****

```

**Step 3** Reply to the bounce confirmation panel; the following list describes the replies that you can enter on this panel:

- To bounce AAOCSM10, enter **Y**.  
CSM bounces AAOCSM10.
- To cancel the bounce, accept the default setting of **N** for **Please confirm BOUNCE for AAOCSM10**.

The object bounce request is rejected.

**Step 4** Press **Enter**.

Figure 2-99 on page 2-119 is displayed.

**Figure 2-99 Global Overview Panel: Object Bounce**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>>                                     SCROLL ==>> CSR
                                                    TGT --- AO6A
Primary Commands: EXception  Group      SORT   SAVE   ADD     PROFile
                  CALendar  CANcel   UNSort  CONVERT BUILD
                  Locate    CMDSHOW Xref    SAVERULE SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)  depe(N)dents  (A)dd
                  sto(P)   (EX)ec   (E)dit   c(L)ients   (D)elete
                  (CAN)cel (M)ove   (R)peat  (B)rowse   s(Y)nc

LC  Object  Group  System  Actual  Desired  Ind  Type
---  -----  -----  -----  -----  -----  ---  -----
___  AAOCSM07  AO6A   SJSC   DOWN    UP        ___  NORM
___  AAOCSM09  AO6A   SJSC   STOPPING  DOWNFORCE  XE  NORM
___  AAOCSM10  AO6A   SJSC   DOWN    UP        NXE  NORM
___  CSBEMSTR  AO63   SJSE   STARTING  UP        XE  NORM
___  MVS
***** Bottom of data *****

```

The actual state of the object is DOWN. When the object restarts after being DOWN, the actual state will appear as UP.

<p><b>Hint</b></p> <p>If an ALERT Detail panel is active while a Bounce is being performed, you might see ALERT ACM755A issued by CSMDOWN on behalf of the object being bounced. The ALERT disappears a second or two later.</p> <p>This behavior is normal and expected. The ALERT is being issued because an object is coming down at an unexpected time. Remember that the Bounce command performs a controlled, unscheduled termination of the object.</p>
--

## Resetting an Object

Occasionally, objects need to be reset back to schedule control when the object is started or stopped at times other than scheduled times or when the object was removed from schedule control altogether.

At the same time, it might be useful to reset the object's command limit count to zero.

The following procedure describes how to reset an object to return it to the control of its schedule.

To reset object AAOCSM10, follow these steps:

**Step 1** Display the Global Overview panel (Figure 2-100).

**Figure 2-100 Global Overview Panel: Resetting an Object**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     SCROLL ==> CSR
                                                TGT --- AO6A
Primary Commands: EXception  Group      SORT    SAVE    ADD      PROFILE
                  CAleNDAR  CANcel   UNSort  CONVERT Build
                  Locate    CMDSHOW Xref    SAVERULE SHUTSYS

LC CMDS ----- (S)tart  b(O)nce  rese(T)  depe(N)dents  (A)dd
                  sto(P)  (EX)ec  (E)dit   c(L)ients    (D)elete
                  (CAN)cel (M)ove  (R)epeat (B)rowse    s(Y)nc

LC  Object  Group  System  Actual  Desired  Ind  Type
---  ---    ---    ---    ---    ---    ---  ---
___  AAOCSM07 AO6A   SJSC   DOWN    UP        XE   NORM
___  AAOCSM09 AO6A   SJSC   STOPPING DOWNFORCE XE   NORM
___  AAOCSM10 AO6A   SJSC   UP       UP        NXE  NORM
___  CSBEMSTR AO63   SJSE   STARTING UP        XE   NORM
___  MVS
***** Bottom of data *****
    
```

**Step 2** Enter the T (reseT) line command next to AAOCSM10.

Figure 2-101 on page 2-120 is displayed.

**Figure 2-101 Confirm Resetting an Object Panel**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
C | BMC Software      Confirm: Return AAOCSM10 to Schedule   AutoOPERATOR |
  | COMMAND ==>                                     TGT --- AO6A      | A
P |
  |
  | Please confirm RESET for AAOCSM10 ? N (Y/N)
L |
  | RESET command count for AAOCSM10 ? N (Y/N)
  |
  | Scope of RESET of AAOCSM10 ? S (Single/Parents/Children)
L |
  |
  | Press ENTER to continue or END / CANCEL to cancel request
T |
___ | CSBEMSTR AO63   SJSE   STARTING  UP        XE   NORM
___ | MVS
***** Bottom of data *****
    
```

**Step 3** Reply to the reset confirmation panel; the following list describes the replies that you can enter on this panel:

- To reset AAOCSM10, enter **Y** for **Please confirm RESET for AAOCSM10**. AAOCSM10 is reset back to schedule control.

If you accept the default setting of **N**, the object reset request is rejected.

- To prevent resetting the object's command count to zero, accept the default setting of **N** for **RESET command count for AAOCSM10**.

If you enter **Y**, CSM resets the command count to zero.

#### Hint

The object command count is incremented each time an object is started during the day, from 00:00 to 23:59.

If the command count reaches or exceeds the defined start command limit for the object, ALERT ACM750A is issued and the object start is interrupted.

The command count is reset to zero in one of three ways:

- The object stops and CSMDOWN executes on the behalf of the object.
- The time-initiated Rule ACMRC060, supplied by BMC Software, fires at 00:00:01 and schedules EXEC ACMM800.
- The command count is set to zero by way of the reset line command.

- For the **Scope of RESET of AAOCSM10** field, you can enter the following replies:

**S** CSM restores to schedule control and, if requested, resets the command count of only the selected object. This setting is the default.

**P** CSM restores the schedule control and, if requested, resets the command count of the selected object and its defined parents.

**C** CSM restores the schedule control and, if requested, resets the command count of the selected object and that object's children.

**Step 4** Press **Enter**.

Figure 2-102 on page 2-122 is displayed.

**Figure 2-102 Global Overview Panel: Object Reset**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>>>                                     SCROLL ==>> CSR
                                                    TGT --- AO6A
Primary Commands: EXCeption  Group      SORT      SAVE      ADD      PROFile
                  CALendar   CANcel    UNsort    CONVERT    BUILD
                  Locate     CMDSHOW  Xref      SAVERULE   SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)  depe(N)dents  (A)dd
                sto(P)    (EX)ec   (E)dit   c(L)ients    (D)elete
                (CAN)cel  (M)ove  (R)peat  (B)rowse     s(Y)nc

LC  Object  Group  System  Actual  Desired  Ind  Type
---  -----  -----  -----  -----  -----  ---  ---
___  AAOCSM07  AO6A   SJSC   DOWN   UP        ___  NORM
___  AAOCSM09  AO6A   SJSC   STOPPING  DOWNFORCE  XE  NORM
___  AAOCSM10  AO6A   SJSC   UP      UP        NXE  NORM
___  CSBEMSTR  AO63   SJSE   STARTING  UP        XE  NORM
___  MVS
***** Bottom of data *****
    
```

The object is returned to scheduled control. CSM either starts or stops the object as you defined it.

**Hint**

All the objects defined to CSM can be restored to schedule control with the use of a single line command. Issue the rese(T) line command for the OS/390 object.

By indicating a scope of C, CSM returns all of the children of OS/390 to schedule control. Every object in the CSM database is a child of OS/390 by default.

The command count for every object in the CSM database also can be reset to zero with one command by using the same procedure.

## Canceling an Object

In a critical situation, an object might be in a condition where it must be stopped quickly and a normal shutdown would take too long. The object can even be in a condition that makes a normal shutdown impossible. In either case, the object must be terminated quickly.

Issuing a cancel command for the object through CSM terminates the object quickly and instructs CSM not to restart the object immediately.

Note that the cancel command does not affect the parents or the children of the selected object. Only the selected object is canceled.

The following procedure describes how to cancel an object:

**Step 1** Display the Global Overview panel (Figure 2-103).

**Figure 2-103 Global Overview Panel: Canceling an Object**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     SCROLL ==> CSR
                                                TGT --- AO6A
Primary Commands: EXception  Group      SORT   SAVE    ADD      PROFile
                  CAleNDAR  CANcel  UNSort  CONVERT BUILD
                  Locate    CMDSHOW Xref   SAVERULE SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)  depe(N)dents  (A)dd
                sto(P)  (EX)ec  (E)dit  c(L)ients  (D)elete
                (CAN)cel (M)ove  (R)epeat (B)rowse  s(Y)nc

LC  Object  Group  System  Actual      Desired      Ind  Type
---  -----  ----  -----  -----  -----  ---  ----
___  AAOCSM07 AO6A   SJSC    DOWN        UP            XE  NORM
___  AAOCSM09 AO6A   SJSC    STOPPING    DOWNFORCE    XE  NORM
___  AAOCSM10 AO6A   SJSC    UP          UP            NXE  NORM
___  CSBEMSTR AO63   SJSE    STARTING    UP           XE  NORM
___  MVS
***** Bottom of data *****
    
```

**Step 2** Enter the C(ancel) line command next to AAOCSM10.

Figure 2-104 is displayed.

**Figure 2-104 Confirm Cancel Panel**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
C | BMC Software          Confirm: Cancel for AAOCSM10      AutoOPERATOR |
P | COMMAND ==>                                     TGT --- AO6A      | A
|
|
L | Please confirm Cancel for AAOCSM10      ? N (Y/N)
|
| WARNING: Be aware that by entering 'Y' in the field above CSM will
|          issue the CANcel command associated with this object
L |
|          The desired state for this object will be set to 'DOWNFORCE'
C |
| Press ENTER to continue or END / CANCEL to cancel request
- |
***** Bottom of data *****
    
```

**Step 3** Reply to the cancel confirmation panel:

- To cancel the object, enter **Y** for **Please confirm CANCEL for AAOCSM10**. The cancel for object AAOCSM10 is completed.
- If you choose not to continue with canceling AAOCSM10, accept the default of **N** and the object cancel request is rejected.

**Step 4** Press **Enter**.

Figure 2-105 is displayed.

**Figure 2-105 Global Overview Panel: Object Cancel**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>>                                     SCROLL ==>> CSR
                                                    TGT --- AO6A
Primary Commands: EXception  Group      SORT      SAVE      ADD      PROFile
                  CALendar   CANcel   UNsort    CONVERT   BUILD
                  Locate     CMDSHOW Xref      SAVERULE  SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)   depe(N)dents (A)dd
                  sto(P)    (EX)ec   (E)dit    c(L)ients  (D)elete
                  (CAN)cel (M)ove  (R)epeat  (B)rowse   s(Y)nc

LC  Object  Group  System  Actual  Desired  Ind  Type
---  -----  -----  -----  -----  -----  ---  ---
___  AAOCSM07 AO6A   SJSC   DOWN    UP        ___  NORM
___  AAOCSM09 AO6A   SJSC   STOPPING DOWNFORCE XE   NORM
___  AAOCSM10 AO6A   SJSC   DOWN    DOWNFORCE NXE  NORM
___  CSBEMSTR AO63   SJSE   STARTING UP        XE   NORM
___  MVS
***** Bottom of data *****
    
```

The object is canceled and the actual state shows DOWN and the desired state shows DOWNFORCE.

CSM does not attempt to start the object until you issue either a S(tart) or a (rese)T line command for the object.

## Moving an Object: Using the Moveable Attribute

Occasionally you might want to move an object from the local BBI-SS PAS to a remote BBI-SS PAS. Moving objects between systems is accomplished by making an object moveable (refer to “Moving an Object between Groups” on page 2-70 for more information).

For example, you might want to perform maintenance on one system, but you want to move an application running on that BBI-SS PAS to another BBI-SS PAS so it can complete running its jobs.

**Note:** An object can be moved from either the Global Overview panel or CSM Exceptions panel. This example shows moving an object from the Global Overview panel.

Moving an object requires that the following conditions be in effect:

- destination BBI-SS PAS has CSM installed and operating
- BBI-SS PAS to BBI-SS PAS communications between the two subsystems is active
- object is associated with the active group on the destination BBI-SS PAS or has been defined on the destination BBI-SS PAS if the two CSMs do not share a repository

**Note:** While an object can be in any state when it is moved, the actual state of the object on the moved to the destination is UP; however, if you are moving objects between CSMs that do not share a repository, the preferred state is UPFORCE.

When you want to move an object:

- Any object that is a candidate to be moved must be associated with the group running on the BBI-PAS that will control the object (usually on another OS/390 image).
- Objects can be moved from a group with the status of *inactive* or *unknown* to an active group if both groups are defined in the same repository. The object moved from the inactive or unknown group will not start in that group when it becomes active. It will remain in the group it was moved to until it is moved to another group.

**Note:** You cannot move an object to a BBI-SS PAS that is not defined or does not have CSM running.

- When defining the object to be moved:
  - The object must be defined to all BBI-SS PASs with exactly the same information except for the schedule.
  - On the BBI-SS PAS that an object is moved to, the object can have its own schedule definitions.
  - The object will not start on a BBI-SS PAS until it is moved there.
  - When an object is moveable, it can be a member of only one group but can be moveable to up to 19 groups.

Once the object is defined, you can move the object from one BBI-SS PAS to another. This example shows moving an object (CICSA) from SYSA to SYSB.

**Step 1** Display the Global Overview panel (Figure 2-106).

**Figure 2-106 Global Overview Panel: Moving an Object**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>>                                     SCROLL ==>> CSR
                                                    TGT --- AO6A

Primary Commands: EXception  Group      SORT      SAVE      ADD      PROFile
                  CALendar   CANcel    UNSort    CONVERT   Build
                  Locate     CMDSHOW  Xref      SAVERULE  SHUTSYS

LC CMDS ----- (S)tart  b(O)nce  rese(T)  depe(N)dents  (A)dd
                  sto(P)   (EX)ec  (E)dit   c(L)ients    (D)elete
                  (CAN)cel (M)ove  (R)peat  (B)rowse     s(Y)nc

LC  Object  Group  System  Actual      Desired      Ind  Type
---  -----  -----  -----  -----  -----  ---  ----
___  AAOCSM07 AO6A   SJSC    DOWN        UP            ___  NORM
___  AAOCSM09 AO6A   SJSC    STOPPING    DOWNFORCE    XE   NORM
___  AAOCSM10 AO6A   SJSC    UP          UP           NXE  NORM
___  CICSA    AA6A   SYSA    UP          UP           ___  NORM
___  CSBEMSTR AO63   SJSE    STARTING    UP           XE   NORM
___  MVS
***** Bottom of data *****
    
```

**Step 2** Enter the M(ove) line command next to the object CICSA.

Before moving an object from a group with a status of inactive or unknown, you should verify that the object is not active on any other group. For the object to be moved, the Move line command can be issued against any group with a status of inactive or unknown. CSM will unconditionally move the management of the object from the inactive or unknown group to the group that you specify as shown in Figure 2-112 on page 2-143.

**Note:** Use caution when moving an object from an inactive or unknown group because an object might be running multiple times in your systems. If you move an object under this condition, it could lead to unpredictable results for that object.

The Confirm Move Panel is displayed (Figure 2-107).

**Figure 2-107 Confirm Move Panel**

```
BMC Software          Confirm: Move for CICSА          MAINVIEW AutoOPERATOR
| COMMAND ==>>          TGT --- NS61 |
|
| Please confirm MOVE for CICSА ? Y    (Y/N)      |
| Group id to MOVE CICSА to    ? NS62          |
|
| Press ENTER to continue or END / CANCEL to cancel request
|-----|
```

**Step 3** To reply to the move confirmation panel:

- Enter **Y**
- Enter the SSID of the destination BBI-SS PAS. In this case, the destination BBI-SS PAS ID is NS62.

**Step 4** Press **Enter**.

The Global Overview panel is redisplayed (see Figure 2-108 on page 2-128).

**Figure 2-108 Global Overview Panel: Moving an Object**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>>>                                     SCROLL ==>>> CSR
                                                    TGT --- AO6A
Primary Commands: EXception  Group      SORT      SAVE      ADD      PROFile
                  CALendar   CANcel   UNSort   CONVERT   BUILD
                  Locate     CMDSHOW Xref     SAVERULE  SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)  depe(N)dents  (A)dd
                sto(P)    (EX)ec   (E)dit   c(L)ients  (D)elete
                (CAN)cel  (M)ove  (R)epeat  (B)rowse   s(Y)nc

LC  Object  Group  System  Actual  Desired  Ind  Type
---  -----  -----  -----  -----  -----  ---  ---
___  AAOCSM07  AO6A   SJSC   DOWN   UP        XE   NORM
___  AAOCSM09  AO6A   SJSC   STOPPING  DOWNFORCE  XE   NORM
___  AAOCSM10  AO6A   SJSC   UP      UP        NXE  NORM
___  CICSA     AA6A   SYSA   DOWN   DOWNFORCE  XE   NORM
___  CSBEMSTR  AO63   SJSE   STARTING  UP        XE   NORM
___  MVS
***** Bottom of data *****

```

The CICS object still appears on the CSM Global Overview panel. In addition, the actual state has been updated to DOWN and the desired state to DOWNFORCE.

## Scheduling an EXEC Associated with a CSM ALERT

The **Ind** column of the Global Overview shows the following information about an object:

- L** indicates that the object has clients
- N** indicates that the object has children (dependents)
- X** indicates that the object has an outstanding ALERT associated with it.

To view the ALERT and its contents, you must use the ALERT Management Facility

- E** indicates that the outstanding ALERT associated with the object has a follow-up EXEC

You can schedule the follow-up EXEC by entering the **X** (eXec) line command next to the object in the LC column.

Some CSM objects might have an outstanding ALERT that does not have an EXEC associated with it. In this case, the **Ind** column displays an X. When the ALERT associated with a CSM object is also associated with an EXEC, the **Ind** column displays *both* an X and an E.

Occasionally, you will want to schedule the EXEC that is associated with a CSM ALERT from the CSM Global Overview. The following steps describe this procedure.

To schedule an EXEC associated with a CSM ALERT:

**Step 1** Check the **Ind** column to ensure that the object has an X and an E displayed.

In this example, the only object with a CSM ALERT and EXEC associated with it is JESA.

**Step 2** Enter the **EX(ec)** line command in the **LC** column next to AAOCSM09 (Figure 2-109).

**Figure 2-109 Global Overview Panel: Scheduling an EXEC**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>>>                                     SCROLL ==>> CSR
                                                    TGT --- AO6A
Primary Commands: EXCeption  Group    SORT    SAVE    ADD    PRoFile
                  CALendar  CANcel  UNsort  CONVERT  BUild
                  Locate    CMDSHOW Xref    SAVERULE SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)  depe(N)dents  (A)dd
                  sto(P)  (EX)ec   (E)dit   c(L)ients    (D)elete
                  (CAN)cel (M)ove   (R)epeat (B)rowse     s(Y)nc

LC  Object  Group  System  Actual  Desired  Ind  Type
---  -----  -----  -----  -----  -----  ---  ---
___  AAOCSM07  AO6A   SJSC    DOWN    UP        ___  NORM
EX_  AAOCSM09  AO6A   SJSC    STOPPING  DOWNFORCE  XE  NORM
___  AAOCSM10  AO6A   SJSC    UP        UP        NXE  NORM
___  CICSA     AA6A   SJSC    UP        UP        ___  NORM
___  CSBEMSTR  AO63   SJSE    STARTING  UP        XE  NORM
___  MVS
***** Bottom of data *****

```

**Step 3** Press **Enter**.

The EXEC is scheduled.

## Displaying and Modifying Object States

You can view both actual and desired states of all objects defined to CSM either

- with the Global Overview panel
- with the CSM Exceptions panel

or

- with two EXECs, CSMACT and CSMINFO, that enable you to manually track and change the different states of an object between the normal UP or DOWN states

You can use these EXECs when you need to override an object's predefined schedule and still keep the object within CSM control. For more information about using the EXECs, refer to "Monitoring CSM Object States Manually" on page 2-133.

"What the Possible Desired States Are" and "What the Possible Actual States Are" on page 2-133 describe the possible desired and actual state a CSM object can have.

### What the Possible Desired States Are

Table 2-3 on page 2-131 shows the possible desired states that an object can have. An object's desired state is the state of the object according to how you scheduled the object, or its desired state can be the result of a command that you issued against the object. You can issue commands (such as start, stop, reset) against an object to change the desired state either through the CSM dialogs or with the EXEC CSMACT.

Use this table to figure out how an object arrived at its desired state:

- The first column lists possible desired states.
- The second column provides a brief description of the desired state.
- The third column shows what command was issued against the object.
- The fourth column shows the original state of the object before the command was issued against the object.

For example, from the first row of the table, you can see that if you execute any of the following commands against an object that is scheduled to be UP

- bounce
- reset
- resetc
- up

the resulting desired state of the object would be UP.

**Table 2-3 Possible Desired States an Object Can Have (Part 1 of 2)**

Resulting Desired State	Description	Command That Was Issued	Original Desired Object State
UP	An object that CSM has started according to its schedule.	BOUNCE	UP
		RESET	Any state of an object that is scheduled to be UP
		RESETC	Any state of an object that is scheduled to be UP
		UP	UP
UPEARLY	Intermediate state of UP: usually the result of bringing the object UP before its scheduled UP time.	BOUNCE	UPEARLY
		UPEARLY*	Any state
UPFORCE	Intermediate state of UP: result of bringing the object UP outside of the object's schedule, before its scheduled UP time.	BOUNCE	UPFORCE
		UP	Any state
UPLOGICAL	Result of bringing a grouping object UP according to its scheduled UP time.	BOUNCE	Uplogical
		UPLOGICAL	Any state
UPLOGICALE	Intermediate state of UP for grouping objects: result of bringing the object UP before its scheduled UP time.	BOUNCE	UPLOGICALE
		UPLOGICALE	Any state
DOWN	An object that CSM has stopped according to its scheduled DOWN time.	RESET	Any state of an object that is scheduled to be DOWN
		RESETC	Any state of an object that is scheduled to be DOWN
		DOWN	DOWN
DOWNEARLY	Intermediate state of DOWN: usually the result of bringing the object DOWN before its scheduled DOWN time.	DOWNEARLY*	Any state

**Table 2-3 Possible Desired States an Object Can Have (Part 2 of 2)**

<b>Resulting Desired State</b>	<b>Description</b>	<b>Command That Was Issued</b>	<b>Original Desired Object State</b>
DOWNFORCE	Intermediate state of DOWN: result of bringing the object DOWN outside of the object's schedule, before its scheduled DOWN time.	DOWN	Any state except DOWN
		CANCEL	Any state
DOWNLOGICAL	Result of bringing a grouping object DOWN according to its scheduled DOWN time.	DOWNLOGICAL	Any state
DOWNLOGICALE	Intermediate state of DOWN for grouping objects: result of bringing the object DOWN before its scheduled DOWN time.	DOWNLOGICALE	Any state
COMPLETE	State for Transient objects when they have been successfully started by CSM according to their schedule.	BOUNCE	COMPLETE
		UPLOGICAL	Any state
		UPLOGICALE	Any state
MOVED	Object control transferred from one group to another.	MOVE	Any State
<p><b>Note:</b> The commands UPEARLY and DOWNEARLY cannot be issued on objects that have been defined as Restart Only. When an object is defined as Restart Only, CSM waits for the scheduling product to start the object.</p>			

## What the Possible Actual States Are

Table 2-4 shows the possible actual (or current) states that an object can have. The actual state of an object is the current state of the object. You can use this table to figure out how an object arrived at its actual state:

- The first column lists the name of the actual state.
- The second column describes the state.

**Table 2-4 Possible Actual States**

Actual State Name	Description
UP	Object has started successfully
STARTING	Object is in an intermediate state of UP: the result of starting the object but CSM has not yet received the start message for the object. If an object's actual state stays in STARTING, the object has probably encountered an error.
UPMANUAL	Object has been brought UP (from a desired state of DOWN) outside of CSM control.
DOWN	Object has been stopped successfully.
CANCELED	Cancel command was issued but CSM has not yet received the stop message for the object.
ABENDED	Object has abended. This state indicates that an abnormal termination event has occurred for an object and the object is supposed to be UP and no longer is UP. The object must be RESE(T) to restore it to CSM control.
COMPLETE	Object has been successfully started by CSM according to its schedule.

## Monitoring CSM Object States Manually

CSM provides EXECs that you can use to manually monitor the state of CSM objects:

- **CSMINFO**  
enables you to see the actual and desired states of objects
- **CSMACT**  
enables you to start or stop CSM objects without using the CSM dialog panels

## Using the CSMINFO EXEC to Display CSM Object States Manually

The CSMINFO EXEC displays the actual and desired states of all CSM objects through a series of ALERTs. These ALERTs are also published to PATROL Enterprise Manager (PATROL EM). But the alerts are usually not useful to the PATROL EM users. Therefore, by using the PUBLISH parameter, you can specify whether the ALERTs should be published. The complete syntax of the CSMINFO EXEC is

```
%CSMINFO [ PUBLISH( ) ]
```

If the PUBLISH parameter is omitted, the shared variable CSMINFO\_PUBLISH\_DEFAULT is used. If this shared variable is not set, the system default for PUBLISH is used. The system default is usually PUBLISH(NO), but it can be changed in AAOPRM00.

The shared variable CSMINFO\_PUBLISH\_DEFAULT is used to establish the default for the CSMINFO PUBLISH, and it can be set in the UBBPARAM(BBIVAR).

### Example 1

To see the actual and desired states of the objects defined to CSM, enter

```
%CSMINFO PUBLISH(ADD)
```

at the command prompt on the ALERT Detail panel or from any MAINVIEW AutoOPERATOR application to display the current (actual) and desired states of all objects defined to CSM.

### Example 2

To see the state of a particular object, enter **%CSMINFO *object\_name* PUBLISH(ADD)**

For more information on PUBLISH parameters, see the *MAINVIEW AutoOPERATOR Advanced Automation Guide*.

## Using the CSMACT EXEC to Modify Object States Manually

Under ordinary circumstances, CSM should control the objects according to their schedules. The CSMACT EXEC enables you to change an object's state without using the CSM dialog panels. For example, you might use the CSMACT EXEC to modify an object if TSO/E and ISPF are not active.

When you use CSMACT, the object is still under CSM control and CSM can still track the state of the object. CSMACT can be scheduled on the **COMMAND** line of any MAINVIEW AutoOPERATOR application. The complete syntax of the CSMACT EXEC is

```
%CSMACT object_name command [DEP|NODEP] [REQ|NOREQ] ssid
```

where

Parameter Value	Description
object_name	is the name of the object  This parameter is required.
command	specifies the command that is to be operated on the object  This parameter is required. Valid commands you can use with CSMACT are listed in "CSMACT Commands" on page 2-136.
DEP	applies the command to the object and all its children
NODEP	applies the command to the object only and not its children
REQ	applies the command against the object's parents
NOREQ	does not apply the command to the object's parents
ssid	is the 4-character BBI subsystem identifier of the BBI-SS PAS where you want action to occur

### Example 1

To start an object named NET3 and all its parents, at the command prompt, enter

```
%CSMACT NET3 UP REQ
```

The NET object and all the objects that are defined as its parents, both direct parents and indirect parents, will be brought UP.

**Example 2**

To stop an object (named NET5) and all its children, at the command prompt enter

**%CSMACT NET5 DOWN DEP**

The NET object and all the objects that are defined as its dependents will be brought DOWN.

**Example 3**

To move an object (named CICSA) to another BBI-SS PAS where CSM is operating, at the command prompt, enter

**%CSMACT CICSA MOVE SS02**

The CICSA object will be brought DOWN on the current subsystem and brought UP on subsystem SS02.

**CSMACT Commands**

The following list shows valid commands that you can use with the CSMACT EXEC to affect the state of an object. They are listed in alphabetical order.

Command	Definition
BOUNCE	<p>stops and immediately restarts an object if the original desired state for the object is UP</p> <p>When used on an object that is UP, UPEARLY, UPFORCE, UPLOGICAL, or UPLOGICALE, the resultant desired state is UP.</p> <p>When issued on an object that is COMPLETE, the resultant desired state is COMPLETE.</p>
CANCEL	<p>issues the object's cancel command</p> <p>When issued on an object in any state, the resultant desired state is DOWNFORCE.</p>
DOWN	<p>stops an object</p> <p>When issued on an object that is DOWN, the resultant desired state is DOWN.</p> <p>When issued on an object that is in any state except DOWN, the resultant desired state is DOWNFORCE.</p> <p>This command requires the DEP or NODEP parameter.</p>

Command	Definition
DOWNEARLY	<p>stops an object and continues under CSM control</p> <p>When issued on any object, the resultant desired state is DOWNEARLY.</p> <p>This command requires the DEP or NODEP parameter.</p>
DOWNLOGICAL	<p>stops a grouping object and removes it from CSM control</p> <p>When issued on an object that is in any state, the resultant desired state is DOWNLOGICAL.</p> <p>This command requires the DEP or NODEP parameter.</p>
DOWNLOGICALE	<p>stops a grouping object and continues under CSM control</p> <p>When issued on an object that is in any state, the resultant desired state is DOWNLOGICALE.</p> <p>This command requires the DEP or NODEP parameter.</p>
MOVE	<p>moves an object to another subsystem</p> <p>This command requires the four-character subsystem ID of the moved-to subsystem.</p> <p>When an object that has an actual state UP is moved, the resultant actual state on the BBI-SS PAS that it is moved to is UPFORCE, and is DOWN on the BBI-SS PAS that it is moved from.</p>
RESET	<p>sets an object's desired state based on its schedule and returns an object to CSM control</p> <p>When issued on an object that is UP, the resultant desired state is UP.</p> <p>When issued on an object that is DOWN, the resultant desired state is DOWN.</p> <p>You can use the DEP or REQ parameters but they are not required.</p>
RESETC	<p>resets an object and the command counter to zero</p> <p>When issued on an object that is UP, the resultant desired state is UP.</p> <p>When issued on an object that is DOWN, the resultant desired state is DOWN.</p> <p>You can use the DEP or REQ parameters but they are not required.</p>
UP	<p>starts an object</p> <p>When issued on an object that is UP, the resultant desired state is UP.</p> <p>When issued on an object that is in any state, the resultant desired state is UPFORCE.</p> <p>This command requires the REQ or NOREQ parameter.</p>

Command	Definition
UPEARLY	<p>starts an object and continues under CSM control</p> <p>When issued on an object that is in any state, the resultant desired state is UPEARLY.</p> <p>This command requires the REQ or NOREQ parameter.</p>
UPLOGICAL	<p>starts a grouping object</p> <p>When issued on an object that is in any state, the resultant desired state is UPLOGICAL or COMPLETE.</p> <p>This command requires the REQ or NOREQ parameter.</p>
UPLOGICALE	<p>starts a grouping object</p> <p>When issued on an object that is in any state, the resultant desired state is UPLOGICALE or COMPLETE.</p> <p>This command requires the REQ or NOREQ parameter.</p>

## Using the Command Line Interface

Use the CSM Command Line Interface to manually monitor the state of CSM objects. When an object group is defined in the local repository, you are asked to specify a command prefix. This unique value is used by CSM to determine which subsystem handles the command entered.

When you browse an object group definition, you will see an entry for the group's command prefix. For example, Figure 2-110 shows the settings for group KMZ1.

**Figure 2-110 Command Prefix Settings for Group KMZ1**

```

BMC Software ----- Group Detail Control - Edit ----- MAINVIEW AutoOPERATOR
COMMAND ==>>> TGT --- KMZ1

Group Name          ==>> KMZ1
Command Prefix      ==>> CSMKM_____

Rule information:
Ruleset Dataset     ==>> BAKMZ.AAO63.KMZ1.UBBPARM_____
Ruleset Id          ==>> G2      Rule Prefix ==>> ACM1      Suffix ==>> 0001

Object Statistics:
Audit               ==>> Y (Y/N)
Dataset Name        ==>> BAKMZ.KMZ1.OFFLOAD_____
Disposition         ==>> MOD

Application information:
Author              ==>> BAKMZ2
Description         ==>> Group List for SYSD Again_____

Last Built   by BAKMZ2 on 07/07/2000 at 12:48:52
Last Modified by BAKMZ2 on 06/23/2000 at 11:43:33

Enter END command to process and return or CANCEL to leave

```

Note that the command prefix specified for this group is CSMKM. Therefore, any commands to group KMZ1 through the interface must be prefixed with CSMKM.

In this example, where a CSM-enabled MAINVIEW AutoOPERATOR subsystem (version 6.1) is using a repository that contains group KMZ1, you can issue commands from an OS/390 console against the CSM objects managed by group KMZ1.

Commands to the interface have the following format:

**user\_specified\_prefix command objectname parm1 parm2**

where the following commands can be issued:

**Table 2-5 Command Line Interface Commands and Parameters**

Command	Purpose	Parameter 1	Parameter 2
STOP	Stop an object.	no entry  EARLY LOGICAL LOGICALE	no entry  DEP NODEP
START	Start an object.	no entry  EARLY LOGICAL LOGICALE	no entry  REQ NOREQ
CANCEL	Cancel execution of an object.	none	none
BOUNCE	Stop and then immediately start an object.	none	none
MOVE	Transfer monitoring and management from CSM enabled subsystem to another. Execution of the object will move from one OS/390 image to another, if necessary.	Either the ID of the group to transfer control to or the value LOCAL, for the local group	none
RESET	Reset object state to that specified in the object's schedule.	none	none
STATUS	Report the desired and actual state of one or more objects. If an object name is not specified then all objects in a particular group will be reported on.	none	none
ENABLE	Enable processing of a group for a local subsystem.	Group ID to enable	SSID to enable group on
DISABLE	Disable processing of a group for a local subsystem.	none	none

In the case of the START and STOP commands, the values in the Parameter 1 column are based on the input parameters to the CSMACT EXEC. When a user starts an object with the interface command

**user\_specified\_prefix START AAOCSM01 EARLY**

CSMACT is scheduled on the appropriate subsystem with the following parameters:

**AAOCSM01 UPEARLY NOREQ**

Some examples follow.

1. Object group KMZ1 manages AAOCSM01. The group's command prefix is CSMKM. To stop AAOCSM01 and its children:

**CSMKM STOP AAOCSM01 DEP**

2. To move object AAOCSM08 from group KMZ1 to group KMZ2. The group prefix for KMZ1 is CSMKM. The group prefix for KMZ2 is CSMK2.

**CSMKM MOVE AAOCSM08 KMZ2**

3. To bounce object AAOCSM09 and the group KMZ2 that monitors it:

**CSMK2 BOUNCE AAOCSM09**

4. To check the status of all of the objects monitored by group KMZ1:

**CSMKM STATUS**

5. To disable object group KMZ1 which is running on subsystem KMZ1:

**CSMKM DISABLE**

6. Object group KMZ1 manages AAOCSM22. The group's command prefix is CSMKM. To start only AAOCSM22 but not its parents:

**CSMKM STOP AAOCSM22 NOREQ**

**Note:** In addition to an OS/390 console, the Command Line Interface commands can be entered on a third party software that simulates an OS/390 console (for example, the IBM product SDSF).

#### **Security Consideration**

To issue the commands in Table 2-5 on page 2-140, you must have security access. If you do not have security access to issue these commands from the terminal session, you will not be able to issue these commands from the command line interface either. If you do not have access, you will receive the WTO:

```
ACM993E User <userid> is not authorized to issue <resource>  
command
```

# Describing CSM Panels

This section contains the CSM panels and descriptions for the line commands, primary commands and fields.

## CSM Global Overview Panel

Figure 2-111 shows an example of the Global Overview panel for CSM. Refer to “What the CSM Global Overview Panel Is” on page 2-44 for more information about this panel.

**Figure 2-111 Global Overview Panel**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>>                                     SCROLL ==>> CSR
                                                    TGT --- KMZ1
Primary Commands: EXCeption  Group      SORT    SAVE    ADD    PROFile
                  CAleNDAR  CANcel   UNsort  CONVERT BUILD
                  Locate    CMDSHOW Xref    SAVERULE SHUTSYS

LC CMDS ----- (S)tart   b(O)unce  rese(T)  depe(N)dents  (A)dd
                  sto(P)   (EX)ec   (E)dit   c(L)ients  (D)elete
                  (CAN)cel (M)ove   (R)epeat (B)rowse   s(Y)nc

LC  Object      Group      System      Actual      Desired      Ind      Type
---  ---
___  MVS
___  AAOCSM01  KMZ1      SJSJ      UP          UP          NORM
___  AAOCSM01  KMZ5      SJSJ      INACTIVE   INACTIVE   NORM
___  AAOCSM01  KMZ6      SJSJ      INACTIVE   INACTIVE   NORM
___  AAOCSM02  KMZ1      SJSJ      COMPLETE   COMPLETE   TRAN
___  AAOCSM02  KMZ2      SJSJ      INACTIVE   INACTIVE   TRAN
___  AAOCSM02  KMZ5      SJSJ      INACTIVE   INACTIVE   TRAN
___  AAOCSM02  KMZ6      SJSJ      INACTIVE   INACTIVE   TRAN
___  AAOCSM08  KMZ1      SJSJ      UP          UP          NORM
___  AAOCSM08  KMZ2      SJSJ      INACTIVE   INACTIVE   NORM
___  AAOCSM08  KMZ5      SJSJ      INACTIVE   INACTIVE   NORM
___  AAOCSM08  KMZ6      SJSJ      INACTIVE   INACTIVE   NORM
    
```

To view additional information about each object, scroll left by pressing **PF10/PF22** (see Figure 2-112 on page 2-143).

**Figure 2-112 Global Overview Panel: Scrolled LEFT**

```

BMC Software ----- Global Overview - Desc/Stats ----- MAINVIEW AutoOPERATOR
COMMAND ==>>>                                     Scroll ==>>> CSR
                                                    TGT --- KMZ1
Primary Commands: EXception  Group      SORT      SAVE      ADD      PROFile
                  CALendar  CANcel   UNsort   CONVERT  BUILD
                  Locate    CMDSHOW Xref     SAVERULE  SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)  depe(N)dents  (A)dd
                  sto(P)   (EX)ec   (E)dit   c(L)ients  (D)elete
                  (CAN)cel (M)ove   (R)peat  (B)rowse   s(Y)nc

LC  Object      Description                                     Id      Date      Time
---  ---
___  AAOCSM01    Child of MVS                                     BAOKMZ2 07/10/2000 15:58
___  AAOCSM01    Child of MVS                                     BAOKMZ2 07/10/2000 15:58
___  AAOCSM01    Child of MVS                                     BAOKMZ2 07/10/2000 15:58
___  AAOCSM02    Child of MVS, Also                             BAOKMZ2 07/10/2000 09:05
___  AAOCSM02    Child of MVS, Also                             BAOKMZ2 07/10/2000 09:05
___  AAOCSM02    Child of MVS, Also                             BAOKMZ2 07/10/2000 09:05
___  AAOCSM02    Child of MVS, Also                             BAOKMZ2 07/10/2000 09:05
___  AAOCSM02    Child of MVS, Also                             BAOKMZ2 07/10/2000 09:05
___  AAOCSM08    Child of MVS Again                             BAOKMZ2 06/21/2000 19:35
___  AAOCSM08    Child of MVS Again                             BAOKMZ2 06/21/2000 19:35
___  AAOCSM08    Child of MVS Again                             BAOKMZ2 06/21/2000 19:35
___  AAOCSM08    Child of MVS Again                             BAOKMZ2 06/21/2000 19:35

```

## Field Descriptions

Table 2-6 describes the fields for the CSM Global Overview panel.

**Table 2-6 Field Descriptions: CSM Global Overview Panel (Part 1 of 2)**

Field	Description
Object	contains the name of the object as specified in the object definition Objects are listed in alphabetical order.
Group	is the name identifying a group of objects managed by Continuous State Manager This name must be unique among the other members of the CSM-Plex to ensure proper object status reporting on the Global Overview display.
System	is the OS/390 SYSNAME where the group is enabled If the group is disabled, this column is blank. For objects that exist in another repository or a pre 6.1.0 system, this field contains the value REMOTE, which means remote or not version 6.1.
Actual	indicates the object's current status in the system This state should always match the desired state. When the actual and desired state do not match, CSM attempts to make them match. Refer to "Displaying and Modifying Object States" on page 2-130 for more information about object states.
Desired	indicates the object's desired state which is governed by its schedule This state should always match the actual state. When the actual and desired state do not match, CSM attempts to make them match. Refer to "Displaying and Modifying Object States" on page 2-130 for more information about object states.

**Table 2-6 Field Descriptions: CSM Global Overview Panel (Part 2 of 2)**

Field	Description
Ind	<p>indicates object relationships</p> <p>Possible values are as follows:</p> <p><b>L</b> indicates the object has clients</p> <p><b>N</b> indicates the object has children (dependents)</p> <p><b>X</b> indicates the object has an outstanding ALERT associated with it. To view the ALERT and its contents, you must use the ALERT Management Facility.</p> <p><b>E</b> indicates that the outstanding ALERT associated with the object has a follow-up EXEC</p> <p>You can schedule the follow-up EXEC by entering the e(X)ec line commands next to the object in the LC column.</p>
Type	shows the type of object: possible values are NORM (normal object) and TRAN (transient object)
Description	shows a description of the object as specified in the object definition
Id	shows the TSO ID of the last person who modified the object
Date	is the last date that the object was modified
Time	is the last time that the object was modified

### Primary Commands

Table 2-7 on page 144 describes the primary commands for the CSM Global Overview panel.

**Table 2-7 Primary Commands: CSM Global Overview Panel (Part 1 of 3)**

Command	Description
Exception   EXC	<p>causes the CSM Exceptions panel to be displayed</p> <p>The (MON)itor and (EXC)eption primary commands toggle the display between the CSM Global Overview panel and the CSM Exceptions panel.</p> <p>On the CSM Global Overview panel, the (EXC)eption command is shown. On the CSM Exceptions panel, the (MON)itor command is shown.</p>
Group	displays the Object Groups application, where you can manage CSM Groups
Sort   SORT	<p>sorts data in (up to) two columns of the CSM display in either ascending or descending order</p> <p>The default sort sequence is ascending alphabetical order based on the data in the Object column.</p>
Save   SAVE	<p>saves changes but does not activate them until CSM has been reinitialized or a BUILD command is issued</p> <p>SAVE <i>does not</i> make changes effective immediately. Issue the BUILD command for changes to be saved and activated.</p>

**Table 2-7 Primary Commands: CSM Global Overview Panel (Part 2 of 3)**

<b>Command</b>	<b>Description</b>
Add   ADD	<p>adds new objects to be controlled by CSM</p> <p>The added object automatically assumes that OS/390 is the parent. Enter ADD on the COMMAND line of the CSM panel when defining new objects.</p> <p>ADD takes effect after you issue the BUILD command.</p>
Calendar   CAL	invokes the Global Calendar Override panel from which you can define special schedules for one-time only situations
Cancel   CAN	<p>causes CSM to issue the (CAN)cel command for the selected object</p> <p>If the definition does not have a (CAN)cel command, the sto(P) command is issued. The (CAN)cel command does not affect the object's parents or children.</p> <p>Before canceling the object, CSM prompts you for verification:</p> <pre>Please confirm Cancel for object Cancel?? N (Y/N)</pre> <p>This command takes effect immediately.</p>
Unsort   UN	<p>causes CSM to return the display to the default sequence of ascending object names</p> <p>The UNSort has the same effect as specifying:</p> <p><b>COMMAND ==&gt;&gt; SORT OBJECT A</b></p> <p>The UNSort command refreshes the CSM display; it <b>does not</b> affect the database.</p>
CONVERT	converts CSM database from MAINVIEW AutoOPERATOR release 4.1 or 5.1 into a new group with objects in the CSM repository
Build   BU	<p>saves your parameter changes in the CSM database and automatically issues a command that tells CSM to reload the control parameters and make the changes effective immediately</p> <p>This command requires the GROUPNAME parameter. When you issue this command, a confirmation panel appears, requesting verification.</p> <p>BUILD makes changes effective immediately.</p>
Locate   L	<p>finds a specified object on the CSM display</p> <p><b>Format: L object name</b></p>
CMDSHOW	<p>enables you to control the display of the Primary and Line commands</p> <p>Issuing the command causes the application to be displayed without the Primary and Line commands.</p> <p>Issuing CMDSHOW again reverses the change and re- displays the Primary and Line commands.</p>

**Table 2-7 Primary Commands: CSM Global Overview Panel (Part 3 of 3)**

Command	Description
Xref   X	<p>invokes the CSM Diagnostic Cross-Reference facility to create a cross-reference of the CSM database</p> <p>The cross-reference report is written to a specified data set from which the report can be browsed or printed. To obtain a printed copy of the report:</p> <ol style="list-style-type: none"> <li>1. Change the default value for Browse from <b>Y</b> to <b>N</b>.</li> <li>2. Specify the SYSOUT class; for example: <ul style="list-style-type: none"> <li>- Data Set Name==&gt;'ABC.CSMXREF.RS31'</li> <li>- Browse report after generation (Y/N)? <b>N</b></li> <li>- SYSOUT class for printed report? <b>R</b></li> </ul> </li> <li>3. Press END to generate or print the cross-reference report.</li> </ol>
SAVERULE	<p>generates the CSM Ruleset that is defined for the specified group.</p> <p>The command requires the group name as a parameter.</p>
SHUTSYS	<p>shuts down all the objects in a group with a single command</p> <p>The command requires that a group name is specified and can have an object as an optional parameter. The default for object is OS/390; therefore, if an object is not specified, all objects from the group are stopped. If an object other than OS/390 is specified, the shutdown occurs for this object and all of its children (dependents).</p> <p><b>Format</b>            <b>SHUTSYS &lt;groupid&gt; &lt;object&gt;</b></p> <p><b>Example</b>            SHUTSYS PROD    - CSM shuts down all objects in group PROD SHUTSYS PROD TSO - CSM shuts down TSO and all of its children</p>
Exceptions   EXC	<p>when issued on the CSM Global Overview panel, causes the CSM Exceptions panel to be displayed</p> <p>The (EXC)eption and (MON)itor primary commands toggle the display between the CSM Global Overview panel and the CSM Exceptions panel.</p> <p>On the CSM Global Overview panel, the (EXC)eption command is shown. On the CSM Exceptions panel, the (MON)itor command is shown.</p>
Monitor   MON	<p>when issued on the CSM Exceptions display, causes the CSM Global Overview panel to be displayed</p> <p>The (EXC)eption and (MON)itor primary commands toggle the display between the CSM Global Overview panel and the CSM Exceptions panel.</p> <p>On the CSM Global Overview panel, the (EXC)eption command is shown. On the CSM Exceptions panel, the (MON)itor command is shown.</p>

## Line Commands

Table 2-8 describes the line commands for the CSM Global Overview panel.

**Table 2-8** Line Commands: CSM Global Overview Panel (Part 1 of 5)

Command	Description
Start   S	<p>starts an object</p> <p>CSM uses this command or the EXEC that is specified for this object.</p> <p>Before starting this object, CSM prompts you with three verification fields:</p> <pre>Please confirm START for object       ? N (Y/N) START all PARENTS of object       ? Y (Y/N) START now, but STOP as scheduled:       ? N (Y/N)</pre> <p>If an object must start without starting its parents, specify <b>N</b> in the second verification field.</p> <p>If TSO/E and ISPF are not active and an object must be started, use the CSM EXEC, CSMACT.</p> <p>If the Start or Stop command needs to be issued in response to an outstanding WTOR, you can use the EXEC CSMWTOR supplied by BMC Software. The CSMWTOR EXEC works only in a JES2 environment. This EXEC has two parameters: object name and reply. The syntax is as follows:</p> <pre>%CSMWTOR object_name reply_text</pre> <p>CSMWTOR can be defined in the Command Specifications panel as the Start or Stop command.</p> <p>This EXEC locates the outstanding reply from the object and responds to it with the reply text that you specified.</p> <p>This command takes effect immediately.</p>
Bounce   O	<p>causes CSM to issue a stop command for the selected object and then restart the object immediately if the object's schedule indicates that the object should be active</p> <p>All other control parameters for the object remain in effect. The b(O)unce command lets you recycle a task without modifying the normal control parameters for the object.</p> <p>The b(O)unce command affects only the selected object. It does not affect the object's parents or children.</p> <p>This command takes effect immediately.</p>

**Table 2-8 Line Commands: CSM Global Overview Panel (Part 2 of 5)**

Command	Description
Reset   T	<p>causes CSM to restore the object to its defined schedule</p> <p>You might use this command if you have used the (S)tart or sto(P) line commands to start and stop objects.</p> <p>Before resetting this object, CSM prompts for the following verifications:</p> <pre> Please confirm RESET for object       ? N (Y/N) RESET command count for object       ? N (Y/N) Scope of RESET of :pv.object:epv.       ? S (Single/Parent/Children) </pre> <p>The third verification field provides you with three options:</p> <p><b>S</b> returns object to its defined schedule  <b>P</b> returns object and its parents to their defined schedules  <b>C</b> returns object and its children to their defined schedules</p> <p>The rese(T) line command clears the status of an object in a manual state and returns it to CSM control.</p> <p>To rese(T) all objects to their normal schedules, reset OS/390 and specify <b>C</b> as the scope. Rese(T) takes effect immediately.</p>
Dependents   N	<p>displays the objects that are defined as being dependents (children) of the selected object</p> <p>In a parent-child relationship, the parent object must be active for the child object to be active. Therefore, if a parent object is taken down, all dependent objects are taken down first.</p> <p>If the selected object has no dependents defined, the Dependents of <i>object</i> panel contains no data.</p>
Add   A	<p>adds objects to the CSM database</p> <p>The added object automatically assumes the schedule of the object on which the (A)dd line command was performed and assumes the selected object as a required object.</p> <p>This command invokes the Object Detail Control panel.</p> <p>(A)dd takes effect after you issue the BUILD command.</p>

**Table 2-8 Line Commands: CSM Global Overview Panel (Part 3 of 5)**

Command	Description
Stop   P	<p>stops an object from the CSM application</p> <p>CSM uses this command or the EXEC that is specified for this object.</p> <p>Before stopping this object, CSM prompts for the following verifications:</p> <pre> Please confirm STOP for object ? N (Y/N) STOP all DEPENDENTS object ? Y (Y/N) STOP now, but START as scheduled: ? N (Y/N) </pre> <p>If a parent object must stop without stopping its children, specify <b>N</b> in the second verification field.</p> <p>If TSO/E and ISPF are not active and an object must be stopped, CSMACT should be used.</p> <p>If the Start or Stop command needs to be issued in response to an outstanding WTOR, you can use the EXEC CSMWTOR supplied by BMC Software. The CSMWTOR EXEC works only in a JES2 environment. This EXEC has two parameters: object name and reply. The syntax is as follows:</p> <pre>%CSMWTOR object_name reply_text</pre> <p>CSMWTOR can be defined in the Command Specifications panel as the Start or Stop command.</p> <p>This EXEC locates the outstanding reply from the object and responds to it with the reply text that you specified.</p> <p>This command takes effect immediately.</p>
Exec   EX	<p>specifies that CSM will schedule a predefined Alert follow-up EXEC for the object on the subsystem indicated in the target field</p> <p>The E(X)ec command takes effect immediately.</p>
Edit   E	<p>modifies the current control parameters for the selected object</p> <p>When you issue this command next to the OS/390 object, the Default Control Parameters panel is displayed.</p> <p>When you issue this command next to any other object, the Object Detail Control panel is displayed. (E)dit modifications take effect after you issue the BUILD command.</p>

**Table 2-8 Line Commands: CSM Global Overview Panel (Part 4 of 5)**

Command	Description
Clients   L	<p>shows the clients of the selected object</p> <p>In this relationship, an object defined as a server can be started and stopped independently of its clients. After the clients are disconnected from the server through commands, the server then stops. When the server is operational again, all defined clients are reconnected automatically to the server through commands (if the clients are scheduled to be up at that time).</p> <p>Clients continue to operate while the server is unavailable but might operate in a partially degraded state.</p> <p>If the selected object has no clients defined to CSM, the Clients of <i>object</i> panel contains no data.</p>
Delete   DE	<p>eliminates the object from CSM control The selected object must be managed through some other means after this command is issued.</p> <p>If you attempt to (DE)lete an object that has children, CSM does not allow deletion of the object. Do one of the following actions to delete a parent object:</p> <ul style="list-style-type: none"> <li>• Delete the child objects and then delete the parent object.</li> <li>• Redefine the parent-child dependency relationship to be a client-server relationship.</li> <li>• Define the child objects to another parent.</li> </ul> <p>(DE)lete modifications take effect after you issue the BUILD command. If you delete an object by mistake, issue the CANCEL primary command.</p>
Cancel   CAN	<p>causes CSM to issue the (CAN)cel command for the selected object</p> <p>If the definition does not have a (CAN)cel command, the sto(P) command is issued. This command does not affect the object's parents or children.</p> <p>Before canceling the object, CSM prompts you for verification:</p> <pre>Please confirm Cancel for object Cancel?? N (Y/N)</pre> <p>This command takes effect immediately.</p>
Move   M	<p>enables you to move CSM objects from the local BBI-SS PAS to a remote BBI-SS PAS where CSM is installed</p> <p>For more information about moving objects, refer to "Moving an Object: Using the Moveable Attribute" on page 2-125.</p>
Repeat   RE	<p>enables you to duplicate an existing object's definition to create a new object with the same information</p> <p>Use this command when you want to define objects with similar attributes. The only limitation is that no two objects can have the same name.</p> <p>Refer to "Using the Repeat Line Command to Create Objects" on page 2-76 for more information about using the Repeat line command.</p>

**Table 2-8 Line Commands: CSM Global Overview Panel (Part 5 of 5)**

Command	Description
Browse   B	<p>enables you to browse the control parameters for the selected object</p> <p>The control parameters cannot be modified with this line command. The (B)rowse command displays the Object Detail Control panel.</p>
Synchronize   Y	<p>checks for discrepancies between the current state and the physical state of an object</p> <p>For example, an object could change state without issuing the expected start/stop messages, so its current and physical states would become unsynchronized.</p> <p>When you issue this command, CSM checks every defined object to see if it is physically present on the system. After checking for the presence of an object, CSM updates the object's current state variable.</p> <p>This command <i>does not</i> change the status of objects that are in a manual state.</p>

## Defining and Modifying Object Panels

To define a new object to CSM, enter the A(dd) line command in the **LC** field next to the object that will be the new object's parent, or enter the **ADD** primary command on the Global Overview panel. The Object Detail Control panel is displayed.

This panel links together all the panels required for defining an object to CSM. The panels are as follows:

- Object Detail Control
- Command Specification
- Conditional Commands
- Event Specification 1
- Event Specification 2
- User Notification
- Object Groups
- Object Requisites
- Client/Server Relationships
- Object Scheduling

The following sections list each of these panels and their field, primary and line command descriptions.

## Object Detail Control Panel

The following describes the Object Detail Control panel.

**Figure 2-113 Object Detail Control Panel Description**

```

BMC Software ----- Object Detail Control - Add ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     TGT --- KMZ1

The following options are displayed in sequence, or may
be selected by entering the three-character code

CMD - Command Specifications           EV1 - Event Specifications 1
EV2 - Event Specifications 2           NOT - User Notification
GRP - Object Groups                    REQ - Object Prerequisites
CLS - Client/Server Relationships      SCH - Object Scheduling

Object Name          ==> AAOCSM01      Member of Group     ==> KMZ1____
Object Type          ==> NORM____      Type of object ( ? for list)

Verify Force Down   ==> N (Y/N)       Restart Only Control ==> N (Y/N)

Application information:
Author              ==> BAOKMZ1
Description         ==> Child of MVS_____

Last Modified by BAOKMZ2 on 01/30/2001 at 15:43

Enter END command to process and return, ENTER to continue, or CANCEL
    
```

### Field Descriptions

Table 2-9 describes the fields for the Object Detail Control panel.

**Table 2-9 Field Descriptions: Object Detail Control Panel (Part 1 of 2)**

Field	Description
Object Name	specifies the name of object. You can specify a name up to 8-alphanumeric characters representing any valid job name.
Member of Group	is the name identifying a group of objects managed by Continuous State Manager This name must be unique among the other members of the CSM-Plex to ensure proper object status reporting on the Global Overview display.  <b>Format:</b> 8-character, alphanumeric <b>Default:</b> None

**Table 2-9 Field Descriptions: Object Detail Control Panel (Part 2 of 2)**

Field	Description
Object Type	<p>is the type of the object, either Normal or Transient</p> <p>A Normal object has a schedule based on the time-of-day and the day-of-week. A Transient object is started once after the IPL of the system and then its state is not managed by CSM. You can enter a question mark (?) in this field to see a list of valid object types.</p> <p><b>Format:</b> NORM or TRAN. Default is NORM</p>
Verify Force Down	<p>prevents accidental stopping of the object by requiring command verification</p> <p><b>Format:</b> Y or N</p> <p><b>Y:</b> CSM automatically asks for the stop request to be verified. You must reply to ALERT ACM705A before CSM will stop the object.</p> <p><b>N:</b> CSM stops the object without confirmation.</p>
Restart Only Control	<p>specifies whether or not CSM waits for the scheduling product to start the object</p> <p>Most sites have a scheduling package that manages work based on a time schedule. If a scheduling package is already managing the schedule of an object and you do not want to move control of that object to CSM, you can specify the CSM manages that object only for restart control with this parameter.</p> <p>When <b>YES</b> is specified, CSM waits for the scheduling product to start the object. If the scheduling product does not start the object at the desired time, CSM issues ALERT messages, indicating that the object is scheduled to be started but it has not started.</p> <p>Once started by the scheduling product, CSM manages the availability of the object by automatically restarting the object if it ends out of schedule.</p> <p>Even though CSM is not responsible for starting the object, you must still specify the complete schedule for the object for this parameter to perform correctly.</p> <p><b>Format:</b> Y or N</p> <p><b>Y:</b> CSM manages only restart functions.</p> <p><b>N:</b> CSM starts, stops, and restarts the object.</p> <p>Objects that are defined as Restart Only can be manually started or stopped one of the following ways:</p> <ul style="list-style-type: none"> <li>• By using the OS/390 START or STOP command</li> <li>• By issuing the UP or DOWN command with the CSMACT EXEC</li> <li>• By changing the object's definition with the object definitions panels.</li> </ul>
Author	TSO ID of the person who modified the object.
Description	<p>Any free-form text you choose to describe this object.</p> <p>The description should be meaningful to the viewer. You can provide the normal system in which the object runs, what the object is, or some indication of the dependencies for the object.</p> <p><b>Format:</b> 1 to 40 alphanumeric characters of free-form text. The text should not contain quote marks.</p> <p><b>Example:</b> IRLM for IMSPROD1 on SYSB</p> <p>The description is not used for any object management function.</p>

## Commands

You can issue the following commands on the **COMMAND** line from the Object Detail Control panel or on any of the panels listed to display another panel.

**Table 2-10**      **Shortcut Commands: Object Detail Control Panel**

<b>Command</b>	<b>Displays This Panel</b>
CMD	Command Specifications
EV1	Event Specifications 1
EV2	Event Specifications 2
NOT	User Notification
GRP	Object Groups
REQ	Object Prerequisites
CLS	Client/Server Relationships
SCH	Object Scheduling

## Command Specifications Panel

The first panel displayed from the Object Detail Control panel is the Command Specifications panel.

**Figure 2-114 Command Specification Panel Description**

```

BMC Software ----- Command Specifications for object ----- MAINVIEW AutoOPERATOR
COMMAND =====> TGT --- NS61

Start Command/EXEC =====> #S AAO41_____
                                     _____
                                     _____
User Post Start EXEC =====> _____
                                     _____
Stop Command/EXEC =====> #P AAO41_____
                                     _____
                                     _____
User Post Stop EXEC =====> _____
Cancel Command/EXEC =====> _____
Recovery Command/EXEC =====> _____

Start Command Time Out =====> 0002          Start Command Limit =====> 0003
Stop Command Time Out =====> 0002          Start Limit Reset =====> Y (Y/N)

Conditional Start =====> N (Y/N)          Conditional Stop =====> N (Y/N)

Enter END command to process and return to Detail Control, ENTER to continue

```

**Field Descriptions**

Table 2-11 describes the fields for the Command Specifications panel.

**Table 2-11 Command Specification Panel Fields (Part 1 of 5)**

Field	Description
<p>Start Command/EXEC</p>	<p>starts this object</p> <p>If you specify a command, enter the same command used to start the object from the OS/390 console. Prefix the Start command with the appropriate command recognition character. For example, use a pound sign (#) as a prefix for OS/390 commands:</p> <p><b>#S JES2</b></p> <p>If the command has any embedded quotation marks, you must use two single quotation marks on each side of the command specified in the <b>Start Command/EXEC</b> field; for example:</p> <p><b>#S ACF,PARM="data"</b></p> <p>If you use an EXEC instead of a command, do not prefix the EXEC name with any command characters. Parameters can be passed to the EXEC; for example, to start IMS with a warm start, specify the EXEC as follows:</p> <p><b>STARTIMS PROD1 WARM</b></p> <p>When the EXEC STARTIMS is scheduled, it will be passed the two parameters PROD1 and WARM.</p> <p>The EXEC name is limited to eight characters and must be a valid PDS member name. The EXEC must reside in a library in the SYSPROC concatenation of the MAINVIEW AutoOPERATOR subsystem address space.</p>
<p><b>Note:</b> A Start Command/EXEC is not required for transient objects.</p>	
<p>User Post Start EXEC</p>	<p>name of an optional, user-specified EXEC that CSM will schedule when CSM receives an object's start message</p> <p>When CSM receives the object's start message, the parameters CSM passes to the User Post Start EXEC include</p> <ul style="list-style-type: none"> <li>• Post Start EXEC name</li> <li>• Object name</li> <li>• Object status (normal, forced, or abnormal up/down)</li> <li>• Start message text-ID</li> <li>• Remaining words in start message</li> </ul> <p>The User Post Start EXEC must reside in a library in the SYSPROC concatenation of the MAINVIEW AutoOPERATOR subsystem address space.</p>

Table 2-11 Command Specification Panel Fields (Part 2 of 5)

Field	Description
Stop Command/EXEC	<p>stops this object normally</p> <p>If you specify a command, enter the same command used to end the object from the OS/390 console.</p> <p>Prefix the Stop command with the appropriate command recognition character. For example, use a pound sign (#) as a prefix for OS/390 commands:</p> <p><b>#F CICSPAY,CEMT P SHUT</b></p> <p>If the command has any embedded quotation marks, you must use two single quotation marks on each side of the command specified in the Stop Command/EXEC field.</p> <p>If you use an EXEC instead of a command, do not prefix the EXEC name with any command characters. Parameters can be passed to the EXEC in the same way as with a startup EXEC.</p> <p>The EXEC name is limited to eight characters and must be a valid PDS member name. The Stop EXEC must reside in a library in the SYSPROC concatenation of the MAINVIEW AutoOPERATOR subsystem address space. The Cancel command/EXEC is issued automatically if the Stop command/EXEC times out. See below.</p>
<p><b>Note:</b> A Stop Command/EXEC is not required for transient objects.</p>	
User Post Stop EXEC	<p>name of an optional, user-specified EXEC that CSM will schedule when CSM receives an object's stop message</p> <p>The User Post Stop EXEC can be used to call batch work from the active scheduling product, update or create incident tracking system tickets, or trigger other events that should occur after the object has ended.</p> <p>The parameters passed to the User Post Stop EXEC are the same as the one you can use for the User Post Start EXEC. The User Post Stop EXEC must reside in a library in the SYSPROC concatenation of the MAINVIEW AutoOPERATOR subsystem address space.</p>
Cancel Command/EXEC	<p>used to cancel this object</p> <p>If you specify a command, enter the same command used to cancel the object from the OS/390 console.</p> <p>Prefix the Cancel command with the appropriate command recognition character. For example, use a pound sign (#) as a prefix for OS/390 commands:</p> <p><b>#Z NET,CANCEL</b></p> <p>If the command has any embedded quotation marks, you must use two single quotation marks on each side of the command specified in the <b>Cancel Command/EXEC</b> field.</p> <p>If you use an EXEC instead of a command, do not prefix the EXEC name with any command characters. Parameters can be passed to the EXEC.</p> <p>The Cancel EXEC must reside in a library in the SYSPROC concatenation of the MAINVIEW AutoOPERATOR subsystem address space. The Cancel command/EXEC is automatically issued if the Stop command/EXEC times out.</p>

**Table 2-11 Command Specification Panel Fields (Part 3 of 5)**

Field	Description
Recovery Command/EXEC	<p>name of an optional, user-specified EXEC that CSM will schedule when CSM receives an object's abnormal termination message</p> <p>If you specify a command and the command has any embedded quotation marks, you must use two single quotation marks on each side of the command specified in the <b>Recovery Command/EXEC</b> field.</p> <p>When CSM receives the object's abnormal termination message, the parameters CSM passes to the Recovery Command/EXEC include</p> <ul style="list-style-type: none"> <li>• Recovery EXEC name</li> <li>• Object name</li> <li>• Object status (which is ABENDDOWN)</li> <li>• Abnormal termination message text-ID</li> <li>• Remaining words in the message</li> </ul> <p>The Recovery EXEC must reside in a library in the SYSPROC concatenation of the MAINVIEW AutoOPERATOR subsystem address space.</p>
Start Command Time Out	<p>CSM provides automatic verification of Start commands.</p> <p>Rather than just changing the object's state when a Start command is issued, CSM waits for the confirmation message to indicate that the Start command was processed properly.</p> <p>The Start Command Time Out parameter indicates to CSM how long it should wait for this confirmation message before posting an ALERT to the ALERT Management Facility to report the failed command.</p> <p><b>Format:</b> <i>mmm</i>, where <i>m</i>=minutes  <b>Range:</b> 1 through 1440 minutes</p>
Stop Command Time Out	<p>CSM provides automatic verification of Stop commands.</p> <p>Rather than just changing the object's state when a Stop command is issued, CSM waits for the confirmation message to indicate that the Stop command was processed properly. The Stop Command Time Out parameter indicates to CSM how long it should wait for this confirmation message before posting an ALERT to the ALERT Management Facility to report the failed command.</p> <p><b>Format:</b> <i>mmm</i>, where <i>m</i>=minutes  <b>Range:</b> 1 through 1440 minutes</p> <p>If you define a Cancel command/EXEC for an object, CSM automatically issues this command when the Stop command/EXEC reaches the time out limit set in this field.</p>

**Table 2-11 Command Specification Panel Fields (Part 4 of 5)**

Field	Description
Start Command Limit	<p>specifies how many times during a 24-hour period that CSM will automatically issue the start commands for an object</p> <p>For example, you can specify that CSM will automatically attempt restarts for three abends of an object. If the restart fails, an ALERT is issued.</p> <p>The count of retries is reset when the object stops normally, at midnight every night, and when you issue the line command <b>Reset</b>.</p> <p>For example:</p> <ul style="list-style-type: none"> <li>• If you specify 0, CSM does not ever attempt to start the object.</li> <li>• If you specify 1, CSM attempts to start the object once with no retries.</li> <li>• If you specify 4, CSM attempts to start the object once and then tries three more times if the object terminates unexpectedly.</li> </ul> <p>If restarting fails at any attempt, an ALERT is issued.</p> <p>After unsuccessfully restarting the object the specified number of times, CSM generates an ALERT to the ALERT Management Facility that the number of retries for the specified object has been reached. An EXEC associated with the ALERT will start the object. No further restarts can be attempted until the start command counter is reset.</p> <p><b>Format:</b> xxxx for 4 positions  <b>Range:</b> 0 to 9999</p>
Start Limit Reset	<p>prevents an abnormally terminating object from endlessly restarting and terminating CSM; holds a Start Command Limit for the object</p> <p>When the object is started an internal command count for the object is compared to the Start Command Limit value. If the value is exceeded, the object is not restarted.</p> <p><b>Format: Y or N</b></p> <p><b>Y:</b> If the object is in an UP state or is a transient, its command count will be set to 1. Otherwise the command count will be set to 0. The count will be set each day at midnight.</p> <p><b>N:</b> Do not reset the command count for this object at midnight.</p>

**Table 2-11 Command Specification Panel Fields (Part 5 of 5)**

<b>Field</b>	<b>Description</b>
Conditional Start	<p>enables you to issue additional command on behalf of an object</p> <p>Use conditional commands when an object requires more than one command in order to be started or a user wants to issue an additional command if the object does not start within a prescribed period of time.</p> <p><b>Format: Y or N</b></p> <p><b>Y:</b> Allow user to specify Conditional Start commands.  <b>N:</b> Do not allow user to specify Conditional Start commands.</p>
Conditional Stop	<p>enables you to issue additional command on behalf of an object</p> <p>Use conditional commands when an object requires more than one command in order to be stopped or a user wants to issue an additional command if the object does not stop within a prescribed period of time.</p> <p><b>Format: Y or N</b></p> <p><b>Y:</b> Allow user to specify Conditional Stop commands.  <b>N:</b> Do not allow user to specify Conditional Stop commands</p>

## Conditional Command Specifications Panel

If you entered **Y** for either the **Conditional Start** or the **Conditional Stop** fields on the Command Specifications panel (refer to “Command Specifications Panel” on page 2-155), the Conditional Commands Specification panel is displayed. Figure 2-115 on page 2-161 shows the panel where you specify conditional Start commands.

**Figure 2-115 Conditional Command Specifications Panel: Start**

```

BMC Software ----- Conditional Command Specification ---- MAINVIEW AutoOPERATOR
COMMAND ==>>>                                         Object --- MVS

After ____ minutes issue ==>>> _____
Start Command/EXEC #1                               _____
                                                    _____

After ____ minutes issue ==>>> _____
Start Command/EXEC #2                               _____
                                                    _____

After ____ minutes issue ==>>> _____
Start Command/EXEC #3                               _____
                                                    _____

After ____ minutes issue ==>>> _____
Start Command/EXEC #4                               _____
                                                    _____

After ____ minutes issue ==>>> _____
Start Command/EXEC #5                               _____
                                                    _____

Enter END command to process and return to Detail Control, ENTER to continue
    
```

Figure 2-116 shows an example of the Stop commands panel.

**Figure 2-116 Conditional Command Specifications Panel: Stop**

```

BMC Software ----- Conditional Command Specification ----- MAINVIEW AutoOPERATOR
COMMAND ==>>>                                         Object --- MVS

After ____ minutes issue ==>>> _____
Stop  Command/EXEC #1                               _____
                                                    _____

After ____ minutes issue ==>>> _____
Stop  Command/EXEC #2                               _____
                                                    _____

After ____ minutes issue ==>>> _____
Stop  Command/EXEC #3                               _____
                                                    _____

After ____ minutes issue ==>>> _____
Stop  Command/EXEC #4                               _____
                                                    _____

After ____ minutes issue ==>>> _____
Stop  Command/EXEC #5                               _____
                                                    _____

Enter END command to process and return to Detail Control, ENTER to continue
    
```

Use a conditional Start or Stop command when you want a command issued after a specified period of minutes has elapsed either after the defined Start or Stop command is issued or after another conditional command has been issued for this object.

**Field Descriptions**

Table 2-12 describes the fields for the Conditional Commands Specifications panels.

**Table 2-12 Object Groups Panel Fields**

Field	Description
After ____ minutes	Enter a number of elapsed minutes before the Conditional Commands should be used.
Start Command/ EXEC	<p>The Conditional Command / EXEC is submitted for execution if the object start event has not been detected and the Conditional Command Wait Time value has been exhausted. For example, you can enter a command: #S IMSCNTL1</p> <p>If this is an OS/390 command, the command must be preceded by the OS/390 command recognition character (#). Always use the pound sign for OS/390 commands. Maximum number of characters for commands submitted to OS/390 is 126.</p> <p>You can also enter an EXEC name: EXECNAME. This is the name of the EXEC used to perform a series of commands for the start up of this task. The EXEC must be in a library included in the SYSPROC concatenation of the MAINVIEW AutoOPERATOR subsystem address space.</p>
Stop Command/ EXEC	<p>The Conditional Command / EXEC is submitted for execution if the object stop event has not been detected and the Conditional Command Wait Time value has been exhausted.</p> <p>For example, you can enter a command: #P IMSCNTL1</p> <p>If this is an OS/390 command, the command must be preceded by the OS/390 command recognition character (#). Always use the pound sign for OS/390 commands. Maximum number of characters for commands submitted to OS/390 is 126.</p> <p>You can also enter an EXEC name: EXECNAME. This is the name of the EXEC used to perform a series of commands for the shut down of this task. The EXEC must be in a library included in the SYSPROC concatenation of the MAINVIEW AutoOPERATOR subsystem address space.</p>

## Event Specifications 1 Panel

Pressing **Enter** from the Command Specifications panel displays the Event Specifications 1 panel.

**Figure 2-117 Event Specifications 1 Panel Example**

```

BMC Software ---- Event Specifications 1 for object ----- MAINVIEW AutoOPERATOR
COMMAND ==>
                                     TGT --- NS61

Event Identifiers / Type:
                                     Generate ==> YES

Start Events:

Text Id #1 ==> $HASP373_____ / MSG_   Job Type / Name ==> STC / _____
Text Id #2 ==> _____ / _____ Job Type / Name ==> ___ / _____
Text Id #3 ==> _____ / _____ Job Type / Name ==> ___ / _____
Text Id #4 ==> _____ / _____ Job Type / Name ==> ___ / _____
Text Id #5 ==> _____ / _____ Job Type / Name ==> ___ / _____

Stop Events:

Text Id #1 ==> $HASP395_____ / MSG_   Job Type / Name ==> STC / _____
Text Id #2 ==> _____ / _____ Job Type / Name ==> ___ / _____
Text Id #3 ==> _____ / _____ Job Type / Name ==> ___ / _____
Text Id #4 ==> _____ / _____ Job Type / Name ==> ___ / _____
Text Id #5 ==> _____ / _____ Job Type / Name ==> ___ / _____

Enter END command to process and return to Detail Control, ENTER to continue

```

### Field Descriptions

Table 2-13 on page 2-164 describes the fields for the Event Specifications 1 panel.

**Table 2-13 Event Specifications 1 Panel Fields (Part 1 of 2)**

Field	Description
Generate	<p>Specifies whether CSM automatically generates Start and Stop Rules for CSM objects.</p> <p>Possible values are as follows:  <b>Yes:</b> CSM will automatically generate Start and Stop Rules.  <b>No:</b> CSM will not automatically generate Start and Stop Rules.</p> <p>The Start and Stop Rules (and optionally, the Rules for an abnormal termination event) must be user-created. These Rules must be stored in a different BBPARM member than AAORULCM. Only the CSM-generated Rules are stored in BBPARM member AAORULCM.</p> <p>If you specify YES on the Event Specifications 1 panel, Rules will be generated for the Text IDs specified on the Event Specifications 2 panel. If you specify NO on the Event Specifications 1 panel, no Rules will be generated for the Text IDs specified on the Event Specifications 2 panel.</p> <p>The reverse is also true: If you specify YES on the Event Specifications 2 panel, Rules will be generated for the Text IDs specified on the Event Specifications 1 panel. If you specify NO on the Event Specifications 2 panel, no Rules will be generated for the Text IDs specified on the Event Specifications 1 panel.</p>
Start Events: Text-ID	<p>Specifies the ID of the message that indicates the object has been started.</p> <p>By default, this field contains the value specified on the default control parameters panel. The Text-ID is used to dynamically build a Rule in the Rule Set allocated for CSM.</p> <p>When this message is issued by an object, the Rules Processor schedules EXEC CSMUP to inform CSM of the object's state.</p> <p><b>Format:</b> 1 to 16 characters  <b>Example:</b> \$HASP373 or IEF403I</p>
Type	<p>Specifies the event type of the event that signals CSM that an object has initialized. A Rule is generated using the Start Text-ID and this event type.</p> <p>Must be one of the following event types: MSG, CMD, ALRT, JRNL</p> <p>For MSG events, if the Job Type /Name fields are left blank, the resulting Rule is generated with a Job Type of STC and a Job name equal to the object name.</p>
Job Type	<p>Specifies the job type of the text-ID that signals to CSM that an object has either started or stopped. The Job Type field, along with the event type, Job Name and Text-ID, is used to construct a Rule for this object.</p> <p>Must be one of the following job types: STC, JOB, TSO</p>

**Table 2-13 Event Specifications 1 Panel Fields (Part 2 of 2)**

Field	Description						
Job Name	<p>Specifies the job name of the event that signals CSM that an object has abnormally terminated. The Job Name field, along with the other fields on the Event Specifications #2 panel are used to construct event rules for this object.</p> <p>Must be a valid job name. Rules Processor wildcard characters are permitted.</p>						
Stop Events Text-ID	<p>Specifies the ID of the message that indicates the object has been stopped.</p> <p>By default, this field contains the value specified on the default control parameters panel. The Text-ID is used to dynamically build a Rule in the Rule Set allocated for CSM.</p> <p>When this message is issued by an object, the Rules Processor schedules EXEC CSMDOWN to inform CSM of the object's state.  <b>Format:</b> 1 to 16 characters  <b>Example:</b> \$HASP395 or IEF404I</p> <p>Normally, CSM monitors object status changes using the object's Start or Stop text-IDs. However, there are situations where monitoring these messages is not sufficient because an object might terminate without producing the expected messages.</p> <p>For example, the expected stop messages might not be produced if the OS/390 FORCE or SYSPROG EXEC commands are used to terminate an object. The End-of-Memory EXEC CSMEOM enables CSM to monitor an object stopping regardless of how the object stops.</p> <p>Because MAINVIEW AutoOPERATOR allows only one End-of-Memory EXEC in each MAINVIEW AutoOPERATOR subsystem, the CSM End-of-Memory EXEC is designed to be called from the EXEC, IMFEOM.</p> <p><b>Parameters passed to the EXEC</b></p> <p>Two parameters are passed to the CSM End-of-Memory EXEC:</p> <ul style="list-style-type: none"> <li>• The address space ID of the terminated address space</li> <li>• A character string which can have one of two values:</li> </ul> <table border="0" data-bbox="456 1325 1321 1419"> <thead> <tr> <th data-bbox="456 1325 646 1350">Parameter Value</th> <th data-bbox="667 1325 802 1350">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="456 1356 565 1381"><b>NORMAL</b></td> <td data-bbox="667 1356 1133 1381">Indicates normal address space termination</td> </tr> <tr> <td data-bbox="456 1388 602 1413"><b>ABNORMAL</b></td> <td data-bbox="667 1388 1321 1413">Indicates address space was terminated by passing it to RTM</td> </tr> </tbody> </table> <p style="margin-left: 40px;">This might happen when using the SYSPROG EXIT command or the OS/390 FORCE command. This is not an indication that the address space abended with a system or user abend code.</p> <p>If the installation already uses an IMFEOM EXEC, you can add a call to the CSMEOM EXEC by adding:</p> <p><b>IMFEXEC SELECT(CSMEOM IMFOASID EOMSTAT)</b> to the IMFEOM EXEC.</p> <p>If you have not used the IMFEOM EXEC before, use BBSAMP sample member SAMPEOM to implement the End-of-Memory EXEC. To use this EXEC, copy SAMPEOM to prefix.UBBPROC, and make sure to rename it to IMFEOM during the copy process.</p>	Parameter Value	Description	<b>NORMAL</b>	Indicates normal address space termination	<b>ABNORMAL</b>	Indicates address space was terminated by passing it to RTM
Parameter Value	Description						
<b>NORMAL</b>	Indicates normal address space termination						
<b>ABNORMAL</b>	Indicates address space was terminated by passing it to RTM						

## Event Specifications 2 Panel

Pressing **Enter** from the Event Specifications 1 panel displays the Event Specifications 2 panel.

**Figure 2-118 Event Specifications 2 Panel Example**

```

BMC Software ---- Event Specifications 2 for EV1 - Add --- MAINVIEW AutoOPERATOR
COMMAND ==>                                     TGT --- NS61

Abnormal Termination Events:                                Generate ==> YES

Text Id      ==> _____ Job Name ==> _____ Job Type   ==> ___
Text String  ==> _____ Word/Op  ==> ___ / ___ Event Type ==> ___

Text Id      ==> _____ Job Name ==> _____ Job Type   ==> ___
Text String  ==> _____ Word/Op  ==> ___ / ___ Event Type ==> ___

Text Id      ==> _____ Job Name ==> _____ Job Type   ==> ___
Text String  ==> _____ Word/Op  ==> ___ / ___ Event Type ==> ___

Text Id      ==> _____ Job Name ==> _____ Job Type   ==> ___
Text String  ==> _____ Word/Op  ==> ___ / ___ Event Type ==> ___

Text Id      ==> _____ Job Name ==> _____ Job Type   ==> ___
Text String  ==> _____ Word/Op  ==> ___ / ___ Event Type ==> ___

Enter END command to process and return to Detail Control, ENTER to continue
    
```

### Field Descriptions

Table 2-14 on page 167 describes the fields for the Event Specifications 2 panel.

**Table 2-14 Event Specifications 2 Panel Fields**

Field	Description
Abnormal Termination events Text-id	<p>Specifies the text-ID of the message that indicates the object has terminated abnormally (abended). This is also known as an abnormal termination event.</p> <p>If you specify an abnormal termination text-ID, when an abnormal termination event occurs, the Rules Processor will schedule the CSMABEND EXEC, which notifies CSM that the object has abended.</p> <p>If a recovery command or EXEC is defined for the object (refer to the description of Recovery Command/EXEC), the command or EXEC is issued and CSM also issues the ALERT ACM766A to notify you that the abnormal termination event has occurred.</p> <p><b>Format:</b> 1 to 16 characters <b>Example:</b> IEF450I</p>
Job Name	<p>Specifies the job name of the event that signals CSM that an object has abnormally terminated. The Job Name field, along with the other fields on the Event Specifications #2 panel are used to construct event rules for this object.</p> <p>Must be a valid job name. Rules Processor wildcard characters are permitted.</p>
Job Type	<p>Specifies the job type of the text-ID that signals to CSM that an object has either started or stopped. The Job Type field, along with the event type, Job Name and Text-ID, is used to construct a Rule for this object.</p> <p>Must be one of the following job types: STC, JOB, TSO</p>
Text String	<p>You might want CSM to handle object abnormal termination processing for particular abend codes displayed in the message identified by the Abend Text-ID field. To do this, you can use the optional Text String field.</p> <p>The 1- to 16 character value specified in this field is compared (using the OP field) with the actual value of the WORDx variable generated from the Text-ID. Abnormal termination handling does not take place unless the compare is successful.</p>
Word	<p>The Abend Text-ID field identifies the event which signifies abnormal termination for an object. The Rules Processor breaks the message down into a series of blank delimited values. Each value is stored in a WORDnn variable.</p> <p>The Abend String Position field identifies which WORDnn variable (i.e. WORD7, WORD12, WORD2, etc.) contains the value compare against the value in the Abend Search String to determine if abnormal termination processing is to take place.</p> <p>Valid values are numeric, up to 4 positions.</p>
Op	<p>The values specified in the Abend Search String and the Abend String Position fields identifies the two values to be compared to determine if abnormal termination processing is to take place for an object. The Abend Compare Operation Code indicates what kind of compare is to take place between the two values.</p> <p>Valid values must be one of the following operands: EQ, LT, GT, NE</p>
Event Type	<p>Specifies the event type of the event that signals CSM that an object has initialized. A Rule is generated using the Abend Text-ID and this event type.</p> <p>Must be one of the following event types: MSG, CMD, ALRT, JRNL</p>

## User Notification Panel

Pressing **Enter** from the Event Specifications 2 panel displays the User Notification panel.

**Figure 2-119 User Notification Panel Example**

```

BMC Software ----- User Notification for object ----- MAINVIEW AutoOPERATOR
COMMAND ==>                                     TGT --- NS61

TSO SEND ==> _____
Users      _____

Pager Notification

Contact # 1 ==> _____ Info ==> _____
Contact # 2 ==> _____ Info ==> _____
Contact # 3 ==> _____ Info ==> _____
Contact # 4 ==> _____ Info ==> _____
Contact # 5 ==> _____ Info ==> _____
Contact # 6 ==> _____ Info ==> _____
Contact # 7 ==> _____ Info ==> _____
Contact # 8 ==> _____ Info ==> _____
Contact # 9 ==> _____ Info ==> _____
Contact # 10 ==> _____ Info ==> _____

Object Alert Queue ==> MAIN_____ Alert Publish Mode ==> ADD_____

Enter END command to process and return to Detail Control, ENTER to continue
    
```

## Field Descriptions

Table 2-15 describes the fields for the User Notification panel.

**Table 2-15 User Notification Panel Fields**

Field	Description
TSO Send User ID	<p>CSM automatically sends a TSO message if an exceptional condition for the managed object exists.</p> <p>The TSO Send User ID parameter should contain the user ID of the responsible party for this object. The message sent to this TSO user ID is the same as the ALERT message generated for the exception condition.</p> <p>If the designated TSO user is not logged on to the system when the message is sent, the message will be dropped (The LOGON attribute is not used).</p> <p><b>Format:</b> userid01  <b>Range:</b> Any valid 1- to 7-character TSO user ID</p>
Contact	<p>Specify in the <b>Contact</b> field either a contact name defined to the Elan Workstation (if the option is installed) or a telephone number to be dialed if a problem is detected with this object. The value specified can be up to 32 characters.</p>
Info	<p>The <b>Information</b> field should contain the 1- to 12-character code that is to be sent to the name of the individual specified by the Contact field. This code should be something that is understood by the recipient of the automatic page.</p>
Object Alert Queue	<p>The name of the ALERT queue in the ALERT Management Facility that will receive ALERTs for this object from CSM.</p> <p><b>Format:</b> 1 to 8 alphanumeric characters  <b>Example:</b> MAIN or CSMOS/390</p>
Alert Publish Mode	<p>Specifies whether the ALERT will be published.</p>

## Object Groups Panel

Pressing **Enter** from the User Notification panel displays the Object Groups panel.

**Figure 2-120 Object Groups Panel Example**

```

BMC Software ----- Object Groups object ----- MAINVIEW AutoOPERATOR
COMMAND ==>>>                                     SCROLL ==>> CSR
                                                    TGT --- NS61

Primary Commands: Locate      CMDSHOW      CANCEL

LC CMDS ----- (R)emove (L)ist      (M)ember M(o)veable

Enter END command to process and return to Detail Control, ENTER to continue

LC   Group           Status      Objects  Description
--   -----
_   NS61             Member        6   Converted on: 01/30/2001 15:45
_   NS62             0
***** Bottom of data *****
    
```

### Field Descriptions

Table 2-16 describes the fields for the Object Groups panel.

**Table 2-16 Object Groups Panel Fields**

Field	Description
Group	Is the name identifying a group of objects managed by Continuous State Manager. This name must be unique among the other members of the CSM-Plex to ensure proper object status reporting on the Global Overview display.
Status	Shows the status of the object.
Objects	Shows the number of objects within the group.
Description	Contains a description about the group.

### Primary Commands

Table 2-17 on page 171 describes the primary commands for the CSM Object Groups panel.

**Table 2-17 Primary Commands: Object Groups Panel**

Command	Description
Locate	enables you to locate a specific CSM Group in the Object Groups display This command works similar to Locate command in ISPF display.
CMDSHOW	enables you to control the display of the Primary and Line commands Issuing the command causes the application to be displayed without the Primary and Line commands.  Issuing CMDSHOW again reverses the change and re- displays the Primary and Line commands.
CANCEL	enables you to cancel your changes

**Line Commands**

Table 2-18 describes the primary commands for the CSM Object Groups panel.

**Table 2-18 Line Commands: Object Groups Panel**

Command	Description
Remove   R	Remove the association of this object with the selected group. Each CSM object must be a member of at least one CSM group. Therefore, you cannot remove an object from a group if it belongs only to that group.
List   L	Display all objects, associated with the selected group.
Member   M	Makes the object a member of the selected group. When object is a member of a group, CSM will start the object according to the schedule on the BBI-SS PAS where the group is active.
Moveable   O	Adds the object as a moveable to the selected group. When object is Moveable to a group CSM will not start the object when this group is active until CSM Move is issued to move the object to that group.

## Object Requisites Panel

Pressing **Enter** from the Object Groups panel displays the Object Requisites panel where you can define the parents for an object.

**Figure 2-121 Object Requisites Panel Example**

```

BMC Software ----- Object Requisites for object ----- MAINVIEW AutoOPERATOR
COMMAND ==>>>                                     SCROLL ==>> CSR
                                                    TGT --- NS61

Primary Commands: ADD END CMDSHOW

LC CMDS ----- (E)dit Object, (D)elete Object

Enter END command to process and return to Detail Control, ENTER to continue

LC Parent Object   Group           Description
-----
_   MVS             NS61             The Operating System
***** Bottom of data *****
    
```

### Field Descriptions

Table 2-19 describes the fields for the Object Requisites panel.

**Table 2-19 Object Requisites Panel Fields**

Field	Description
Parent Object	Contains the name of the parent for an object. If that object is a member of more than one group, each group will be listed on this panel.
Group	Is the name identifying a group of objects managed by Continuous State Manager. This name must be unique among the other members of the CSM-Plex to ensure proper object status reporting on the Global Overview display.
Description	Contains a description about the object.

## Primary Commands

Table 2-20 describes the primary commands for the CSM Object Requisites panel.

**Table 2-20 Primary Commands: Object Requisites Panel**

Command	Description
ADD	Enter ADD to add a new parent for an object.  Displays the Add Object Requisites panel. On this panel, enter the object name of the parent and the group the object belongs to.
END	Enter END to return to the Object Detail Control panel.
CMDSHOW	enables you to control the display of the Primary and Line commands Issuing the command causes the application to be displayed without the Primary and Line commands.  Issuing CMDSHOW again reverses the change and re- displays the Primary and Line commands.

## Line Commands

Table 2-21 describes the primary commands for the CSM Object Requisites panel.

**Table 2-21 Line Commands: Object Requisites Panel**

Command	Description
Edit   E	Displays the Edit Object Requisites panel. On this panel, enter the object name of the parent and the group the object belongs to.
Delete   D	Removes a server object from the required objects list for a selected object.

## Client/Server Relationships Panel

Pressing **Enter** from the Object Requisites panel displays the Client/Server Relationships panel where you can define a client/server relationship for an object.

**Figure 2-122 Client/Server Relationships Panel Example**

```

BMC Software --- Client/Server Relationships for object -- MAINVIEW AutoOPERATOR
COMMAND ===>                                SCROLL ==> CSR
                                                TGT --- NS61

Primary Commands: ADD END CMDSHOW

LC CMDS ----- (E)dit server, (D)elele server

Enter END command to process and return to Detail Control, ENTER to continue

LC   Server      Group      Description
                                Connect Command
                                Disconnect Command

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

In a client-server relationship, clients and servers can be started and stopped independently of each other. CSM uses the Disconnect command you specify to disconnect the client from the server. When the server object is restarted, the client object is reconnected to the server using the Connect command you specify.

### Field Descriptions

Table 2-22 describes the fields for the Client/Server Relationships panel.

**Table 2-22 Client/Server Relationships Panel Fields (Part 1 of 2)**

Field	Description
Server	<p>Contains the name of the server for an object. If that object is a member of more than one group, each group will be listed on this panel.</p> <p>This is the same name that is started from the OS/390 console or submitted as a batch job. The server name must be an object under the control of CSM. This field is modifiable.</p>
Group	<p>Is the name identifying a group of objects managed by Continuous State Manager. This name must be unique among the other members of the CSM-Plex to ensure proper object status reporting on the Global Overview display.</p>
Description	<p>Contains a description about the object.</p>

**Table 2-22 Client/Server Relationships Panel Fields (Part 2 of 2)**

Field	Description
Connect Command	<p>A command or EXEC routine to connect the client object to the server object. This is usually, though not restricted to, an OS/390 Modify command or a VTAM Vary command.</p> <p>Prefix this command with the appropriate command recognition character, or CSM reads the input as an EXEC. Use a pound sign (#) for OS/390 commands.</p> <p>Sample Modify command:</p> <p><b>#F IMSPROD1,START</b></p> <p>Sample VTAM Vary command:</p> <p><b>#V NET,ACT,ID=IMSPROD1</b></p> <p>If an EXEC name is used, it must reside in the library concatenation of the SYSPROC DD name of the MAINVIEW AutoOPERATOR subsystem address space.</p>
Disconnect Command	<p>A command or EXEC to disconnect the client object from the server object before the server object is ended. This is usually, though not restricted to, an OS/390 Modify command or a VTAM Vary command.</p> <p>Prefix the Disconnect command with the appropriate command recognition character, or CSM reads the input as an EXEC. Use a pound sign (#) for OS/390 commands.</p> <p>Sample Modify command:</p> <p><b>#F IMSPROD1,STOP</b></p> <p>Sample VTAM Vary command:</p> <p><b>#V NET,INACT,ID=IMSPROD1</b></p> <p>If an EXEC name is used, it must reside in the library concatenation of the SYSPROC DD name of the MAINVIEW AutoOPERATOR subsystem address space.</p>

## Primary Commands

Table 2-23 describes the primary commands for the CSM Client/Server Relationships panel.

**Table 2-23 Primary Commands: Client/Server Relationships Panel**

Command	Description
ADD	<p>Enter ADD to add a new server for an object. Displays the Add Client/Server Relationships where you can add the</p> <p>Server name of the object                      Group Name of the object                      Connect command for the object                      Disconnect command for the object</p>
END	Enter END to return to the Object Detail Control panel.
CMDSHOW	<p>enables you to control the display of the Primary and Line commands                      Issuing the command causes the application to be displayed without the Primary and Line commands.</p> <p>Issuing CMDSHOW again reverses the change and re- displays the Primary and Line commands.</p>

## Line Commands

Table 2-24 describes the primary commands for the CSM Client/Server Relationships panel.

**Table 2-24 Line Commands: Client/Server Relationships Panel**

Command	Description
Edit   E	<p>Displays the Edit Client/Server Relationships panel. On this panel, edit the</p> <p>Server name of the object                      Group Name of the object                      Connect command for the object                      Disconnect command for the object</p>
Delete   D	Removes a server object from the required objects list for a selected object.

## Object Scheduling Panel

Pressing **Enter** from the Client/Server Relationships panel displays the Object Scheduling panel where you can create a schedule of DOWN times for an object.

**Figure 2-123 Object Scheduling Panel Example**

```

BMC Software ----- Object Scheduling for object ----- MAINVIEW AutoOPERATOR
COMMAND ==>>                                     SCROLL ==>> CSR
                                                    TGT --- NS61

Primary Commands: ADD CANcel END CMDSHOW

LC CMDS ----- (E)dit Time, (D)elete Time

Enter END to process/return to Detail Control, ENTER to continue, or CANcel

LC Object      Day      Down From      Down To
  Group                HH:MM          HH:MM

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

```

In a client-server relationship, clients and servers can be started and stopped independently of each other. CSM uses the Disconnect command you specify to disconnect the client from the server. When the server object is restarted, the client object is reconnected to the server using the Connect command you specify.

## Field Descriptions

Table 2-25 describes the fields for the Object Scheduling panel.

**Table 2-25 Object Scheduling Panel Fields**

Field	Description
Object/Group	The name for the group for which this schedule is created. The object must be associated with this group.
Day	The day when the object is scheduled to be DOWN.
Down From	<p>The time the object is scheduled to come down.</p> <p>CSM uses the 24-hour clock for managing down times. If a down time is to span midnight, two down time parameters must be specified. For example, if an IMS Control Region object is scheduled to be DOWN from 8:00 P.M. on Sunday night to 2:00 A.M. on Monday morning, the scheduled down time for that object would be specified as:</p> <pre>SUNDAY    20:00    24:00 MONDAY    00:00    02:00</pre> <p><b>Format:</b> <i>hh:mm</i>, where <i>hh</i>=hours and <i>mm</i>=minutes  <b>Range:</b> 00:00 to 24:00</p> <p>The 00:00 is the beginning of the day; the 24:00 is the end of the day.</p>
Down To	<p>The time the object starts after a scheduled down time. CSM uses the 24-hour clock for managing down times.</p> <p><b>Format:</b> <i>hh:mm</i>, where <i>hh</i>=hours and <i>mm</i>=minutes  <b>Range:</b> 00:00 to 24:00</p> <p>The 00:00 is the beginning of the day; the 24:00 is the end of the day.</p>

## Primary Commands

Table 2-26 describes the primary commands for the CSM Object Scheduling panel.

**Table 2-26 Primary Commands: Object Scheduling Panel**

Command	Description
ADD	<p>Use to add a scheduled DOWN time. Displays the Add Object Scheduling panel where you can add</p> <ul style="list-style-type: none"> <li>• name for the group for which this schedule is created. The object must be associated with this group.</li> <li>• days of the week the object is supposed to be down <ul style="list-style-type: none"> <li>-- To schedule an object to be down at a specific time on a specific day of the week, enter the day the down time is to occur.</li> <li>-- For ease of use, days have also been grouped by Weekday (Monday, Tuesday, Wednesday, Thursday and Friday), Weekend (Saturday and Sunday) and Daily (all the days of the week). All days can be abbreviated to the first two letters except for Weekday (WKD), Weekend (WKN) and Daily (DLY).</li> <li>-- Saturday, Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Weekday, Weekend, and Daily</li> </ul> </li> <li>• Down From and Down To times (in HH:MM format)</li> </ul>
END	Enter <b>END</b> to return to the Object Detail Control panel.
CMDSHOW	<p>enables you to control the display of the Primary and Line commands Issuing the command causes the application to be displayed without the Primary and Line commands.</p> <p>Issuing <b>CMDSHOW</b> again reverses the change and re- displays the Primary and Line commands.</p>

## Line Commands

Table 2-27 describes the primary commands for the CSM Object Scheduling panel.

**Table 2-27 Line Commands: Object Scheduling Panel**

Command	Description
Edit   E	<p>Displays the Edit Object Scheduling panel. On this panel, edit</p> <ul style="list-style-type: none"> <li>• name for the group for which this schedule is created. The object must be associated with this group.</li> <li>• days of the week the object is supposed to be down <ul style="list-style-type: none"> <li>-- To schedule an object to be down at a specific time on a specific day of the week, enter the day the down time is to occur.</li> <li>-- For ease of use, days have also been grouped by Weekday (Monday, Tuesday, Wednesday, Thursday and Friday), Weekend (Saturday and Sunday) and Daily (all the days of the week). All days can be abbreviated to the first two letters except for Weekday (WKD), Weekend (WKN) and Daily (DLY).</li> <li>-- Saturday, Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Weekday, Weekend, and Daily.</li> </ul> </li> <li>• Down From and Down To times (in HH:MM format)</li> </ul>
Delete   D	Use to delete a scheduled DOWN time.

## Global Calendar Override Panel

CSM enables the creation of calendar overrides for special situations in which objects can be brought DOWN outside the normal scheduled down time. Overrides also can be scheduled to extend the time an object is available.

The Global Calendar Override application does not check for conflicting schedules with dependents of the object for which the override is scheduled.

When an override is specified for an object, that object and all of its dependents are taken down on the specified date and time. This eliminates the need to specify the special down times for each object affected by the temporary schedule change.

To access the Global Calendar Override panel, issue the CALendar primary command from the CSM Global Overview panel (Figure 2-124 on page 2-181 is displayed).

**Figure 2-124 Global Calendar Override Panel Example**

```

BMC Software ----- Global Calendar Override ----- MAINVIEW AutoOPERATOR
COMMAND ==>>> TGT --- NS61

Primary Commands: ADD CANCEL END CMDSHOW DATE --- 01/31/20
TIME --- 17:25

LC CMDS --- (I)nsert Time, (D)elete Time

Press END to process and return, CANCEL to exit without changes.

LC Object      Group /      Calendar      Down      Reason for Down Time
                SYSNAME      Date      From      To      Contact Name
                mm/dd/yy  HH:MM      HH:MM

-----
***** Bottom of data *****
    
```

The Global Calendar Override functions under the control of a specific calendar date. The override is only for the date specified in this panel; therefore, the affected objects do not have to be restored to their normal schedules after the override takes place.

**Field Descriptions**

Table 2-28 describes the fields for the Global Calendar Override panel.

**Table 2-28 Global Calendar Override Panel (Part 1 of 2)**

Field	Description
Object	<p>Object name to which this special schedule override applies.</p> <p>Because of the parent-child dependencies that CSM manages, CSM automatically overrides the schedules of any objects that are defined as dependents of the selected object. CSM also automatically disconnects any client objects defined to the selected object before ending the object.</p> <p>When the object is returned to operation, the dependent objects are restarted and the client objects are connected to the server object automatically.</p> <p><b>Format: objname</b></p>
Group/SYSNAME	<p>Group and OS/390 SYSNAME where this override is supposed to occur.</p>
Calendar Date	<p>Specific date for which the special schedule applies.</p> <p>This is the only date the override is in effect. On all other days and dates, the normal schedules for the objects are in effect. If a special schedule is specified for an object, it is used in addition to the normal schedule for the object.</p> <p><b>Format: 8 characters mm/dd/yy, where mm=month, dd=day, yy=year</b></p>

Table 2-28 Global Calendar Override Panel (Part 2 of 2)

Field	Description
Down From	<p>Time CSM begins the process of stopping the selected object and its dependents.</p> <p>If it is necessary to have the system completely down by a specific time, you must allow time for the normal shutdown process to occur. If a down time is to span midnight of one day into another day, two down time parameters must be specified.</p> <p>For example, if an IMS Control Region object is scheduled to be DOWN from 8:00 P.M. on Sunday night to 2:00 A.M. on Monday morning, the scheduled down time for that object would be specified as:</p> <pre>SUNDAY    22:00    24:00 MONDAY    00:00    02:00</pre> <p><b>Format:</b> <i>hh:mm</i>, where <i>hh</i>=hours and <i>mm</i>=minutes  <b>Range:</b> 00:00 to 24:00</p> <p>The 00:00 is the beginning of the day; the 24:00 is the end of the day.</p>
Down To	<p>Time CSM attempts to start the object, except in the case of OS/390.</p> <p>If OS/390 is not active, then MAINVIEW AutoOPERATOR and CSM are not active. However, when MAINVIEW AutoOPERATOR and CSM are initialized, CSM automatically starts all the objects that should be active at that time.</p> <p><b>Format:</b> <i>hh:mm</i>, where <i>hh</i>=hours and <i>mm</i>=minutes  <b>Range:</b> 00:00 to 24:00</p> <p>The 00:00 value is the beginning of the day; the 24:00 value is the end of the day.</p>
Reason for Down Time	<p>Lets you document the reason for the special down time for the affected objects.</p> <p><b>Format:</b> 1 to 21 positions of free-form text</p>
Contact	<p>Group or individual responsible for scheduling the special down time for the affected objects.</p> <p><b>Format:</b> 1 to 16 positions of free-form text</p>

## Primary Command

Table 2-29 describes the primary commands for the Global Calendar Override panel.

**Table 2-29 Global Calendar Override Primary Commands**

Command	Description
ADD	<p>Add scheduled down times for the selected objects.</p> <p>Type <b>ADD</b> in the <b>COMMAND====&gt;</b> area of the display for an additional scheduled down time for this object. If the added schedule conflicts with any required objects, a message indicating the condition is displayed.</p> <p>ADD takes effect after you issue the BUILD command.</p>
CANcel	<p>Return to the screen without processing changes.</p> <p>Type <b>CANcel</b> in the <b>COMMAND====&gt;</b> area of the display to cancel any changes and return to the screen.</p>
END	<p>Process changes and return to the screen.</p> <p>Type <b>END</b> in the <b>COMMAND====&gt;</b> area of the display to process your changes and return to the screen.</p>
CMDSHOW	<p>enables you to control the display of the Primary and Line commands</p> <p>Issuing the command causes the application to be displayed without the Primary and Line commands.</p> <p>Issuing <b>CMDSHOW</b> again reverses the change and re- displays the Primary and Line commands.</p>

## Line Commands

Table 2-30 describes the line commands for the Global Calendar Override panel.

**Table 2-30 Global Calendar Override Line Commands**

Command	Description
Insert Time   I	<p>Add an additional object and special schedule to the Global Calendar Override panel.</p> <p>To insert a new schedule for an object, type the line command in the line command <b>(LC)</b> column and press <b>Enter</b>. Use the tab key to move the cursor to the name field and fill in the blanks in the display. Upon completing each field, press the tab key to position the cursor in the next field.</p> <p>(I)nsert modifications take effect after you issue the <b>BUILD</b> command.</p>
Delete Time   D	<p>Remove the special schedule for an object.</p> <p>To use this line command, use the tab key to position the cursor in the line command <b>(LC)</b> column for the desired object and type <b>D</b>. Press <b>Enter</b> to remove the special schedule for the selected object.</p> <p>If you delete a schedule by mistake, issue the <b>CANCEL</b> primary command.</p> <p>(D)elelete modifications take effect after you issue the <b>BUILD</b> command.</p>

## Object Groups Panel

Every object must belong to a group. Therefore, when you bring CSM up for the first time, the first task you must complete is to create a group before you can begin defining objects. A group name can be one- to eight-characters long, but it is recommended that you name the first group the SSID of the subsystem that you bring CSM up on.

Table 2-125 on page 185 shows an example of the Object Groups panel.

**Figure 2-125 Object Groups Panel Example**

```

BMC Software ----- Object Groups ----- MAINVIEW AutoOPERATOR
COMMAND ==>>                                     SCROLL ==>> CSR
                                                    TGT --- KMZ1

Primary Commands: ADD      Locate      CMDSHOW      BUILD

LC CMDS ----- (BU)ild (S)ync (E)dit E(n)able (D)isable (O)ffload
                  (A)dd (DE)lete (L)ist (R)repeat (B)rowse

LC  Group      System      SSID      Objects      Description
   _____  _____  _____  _____  _____

```

**Field Descriptions**

Table 2-31 describes the fields for the Object Groups panel.

**Table 2-31 Object Groups Panel Fields**

Field	Description
Group	is the name identifying a group of objects managed by Continuous State Manager  This name must be unique among the other members of the CSM-Plex to ensure proper object status reporting on the Global Overview display.
System	is the OS/390 SYSNAME where the group is enabled. If a group is disabled, this field will be blank.
SSID	is the subsystem ID (SSID) where the group is enabled  If a group is disabled, this field will be blank.
Objects	lists the objects in the group
Description	contains a description about the group

## Primary Commands

Table 2-32 describes the primary commands for the CSM Object Groups panel.

**Table 2-32 Primary Commands: Object Groups Panel**

Command	Description
ADD	<p>enables you to add a new Group to the current CSM repository</p> <p>CSM displays the Group Detail Control panel where you can specify all Group definitions (refer to Figure 2-126 on page 2-188).</p> <p>The ADD Primary command can be entered by itself, or with a Group name; for example: <b>ADD GROUP123</b></p> <p>To activate the newly created Group you need to issue Enable line command from the Object Groups display and specify on which CSM partner (PAS) the Group should be active.</p>
Locate	<p>enables you to locate a specific CSM Group in the Object Groups display</p> <p>This command works similar to Locate command in ISPF display.</p>
CMDSHOW	<p>enables you to control the display of the Primary and Line commands</p> <p>Issuing the command causes the application to be displayed without the Primary and Line commands.</p> <p>Issuing <b>CMDSHOW</b> again reverses the change and re- displays the Primary and Line commands.</p>
BUILD	<p>enables you to issue Build against CSM Group</p> <p>Build saves all changes made online and activates them immediately.</p> <p>The <b>BUILD</b> command requires parameter of Group Name.</p>

## Line Commands

Table 2-33 describes the primary commands for the CSM Object Groups panel.

**Table 2-33 Line Commands: Object Groups Panel**

Command	Description
Build   BU	enables you to issue Build against CSM Group Build saves all changes made online and activates them immediately.
Synchronize   Y	Tells CSM that there could be a discrepancy between the current state and the physical state of an object.  For example, an object could change state without issuing the expected start/stop messages, so its current and physical states would become unsynchronized.  When you issue this command, CSM checks every defined object to see if it is physically present on the system. After checking for the presence of an object, CSM updates the object's current state variable.  This command <b>does not</b> change the status of objects that are in a manual state.
Offload   O	offloads captured event statistics information to the statistics data set named in the group's definition
Edit   E	displays the Group Detail Control panel where you can edit the attributes of the group
Enable   N	enables a group
Disable   D	disables a group
Add   A	add new groups to be controlled by CSM
Delete   DE	eliminates the group from the control of CSM
List   L	display all objects, associated with the selected group
Repeat   R	enables you to duplicate an existing group's definition to create a new group with the same information  Use this when you want to define groups with similar attributes. The only limitation is that no two groups can have the same name.
Browse   B	enables you to browse the control parameters for the selected group.  The control parameters cannot be modified with this line command. The (B)rowse command displays the Object Groups Detail Control panel.

## Group Detail Control

Figure 2-126 shows an example of the Group Detail Control panel.

**Figure 2-126 Group Detail Control Example**

```

BMC Software ----- Group Detail Control - Add ----- MAINVIEW AutoOPERATOR
COMMAND ==>
Group Name ==> _____
Command Prefix ==> _____

Rule information:
Ruleset Dataset ==> _____
Ruleset Id ==> __ Rule Prefix ==> ____ Suffix ==> ____

Object Statistics:
Dataset Name ==> _____
Disposition ==> ____ (SHR/MOD)

Application information:
Author ==> BAKMZ1
Description ==> _____

Enter END command to process and return or CANCEL to leave

```

### Field Descriptions

Table 2-34 describes the fields for the Group Detail Control panel.

**Table 2-34 Group Detail Control Panel Fields (Part 1 of 2)**

Field	Description
Group name	The name identifying a group of objects managed by Continuous State Manager. This name must be unique among the other members of the CSM-Plex to ensure proper object status reporting on the Global Overview display.
Command Prefix	A 1- to 16-character identifier used by the CSM Command Line Interface to route commands to the appropriate CSM object group for processing. The prefix must be unique among other members of the local CSM repository.  Commands issued to the interface can enable users to monitor and manage objects in the group without the need for a terminal session display, knowledge of specialized execs and parameters or displaying groups of MAINVIEW AutoOPERATOR variables.

Table 2-34 Group Detail Control Panel Fields (Part 2 of 2)

Field	Description
Rule Dataset	<p>The Rules that indicate the Start, Stop and Recovery events for the objects managed by this group are stored in a parameter data set that is allocated to the MAINVIEW AutoOPERATOR BBI-PAS. In order for the CSM application to know where to write the Rules to be generated when either the BUILD or the SAVERULE command is issued for this group you must specify the Ruleset data set name in this field.</p> <p>For proper operation, the data set specified in this entry must be in the BBIPARM DD concatenation of the MAINVIEW AutoOPERATOR BBI-PAS.</p>
Ruleset I D	<p>The 2-character suffix of the Ruleset ID containing the Start, Stop and Recovery events for the objects managed by this group. The subsystem component of CSM will append this suffix onto the string AAORUL and checks to see if this Ruleset is active when CSM is managing this group.</p> <p>The Ruleset identified by this suffix must be present in one of the data sets in the BBIPARM concatenation in MAINVIEW AutoOPERATOR BBI-PAS.</p>
Rule Prefix	A 4-character string makes up the first four characters of the Rules generated by CSM for this group (see Ruleset id tutorial). As each Rule is generated by the CSM application the prefix will have the Rule Suffix appended to it and written into the Rule ID field.
Suffix	A 4-digit number that makes up the last four characters of the Rules generated by CSM for this group. As each Rule is generated by the CSM application the suffix is incremented by one, appended to the Rule Prefix and written into the Rule ID field.
Dataset Name	<p>If you want CSM to capture statistics on object Start, Stop and Recovery events, the name of a preallocated data set must be supplied in the <b>Statistics Dataset Name</b> field.</p> <p>When CSM captures event statistics information the data is written to the CSMSTATISTICS array in the local subsystem. EXEC ACMM800 off loads the data from the array daily at midnight to the data set named in the <b>Statistics Dataset Name</b> field.</p> <p><b>Note:</b> The CSM object statistics data set should be a fixed block sequential data set with the maximum record length of 80 bytes.</p>
Disposition	Used in conjunction with the <b>Dataset Name</b> field. Determines if the data is to be offloaded to the beginning of the data set (SHR) or after the data previously off-loaded (MOD) to the data set.
Author	Display-only field showing the TSO ID of the person who last modified the group.
Description	Brief description (30 characters or less) of the group.



---

## Chapter 3 JES2 Solution

The Job Entry Subsystem (JES) is critical to the processing flow of your MVS environment. You might have jobs that execute under the control of JES that are critical to your data center.

BMC Software provides the JES2 Monitoring Solution, which alerts data center personnel of any JES2-related problems.

### JES2 Monitoring Solution

The Job Entry Subsystem is a critical MVS subsystem. If JES2 is not functioning correctly, the throughput on the MVS system is adversely affected. The JES2 Monitoring Solution ensures that JES2 system problems are recognized immediately.

## Variables

For this solution to function properly in your environment, you must establish values for the following variables:

<b>SYSPROG</b>	TSO user ID of the primary system programmer who is to receive warning messages
<b>SYSBEEP</b>	information to be placed on pager
<b>MVSCALL</b>	name of support person for Elan to page. This name must be defined on the Elan Workstation
<b>SYSJES</b>	version, release, and modification level of JES in the form <i>SPn.n.n of</i>
<b>SYSMVS</b>	version, release, and modification level of MVS in the form of <i>SPn.n.n.</i>

See Appendix D, “MVS Solution Variables” for information about initializing variables and default settings.

## Invocation

This solution is invoked by Rules for the following JES2 messages:

<b>\$HASP050</b>	JES2 Resource Shortage
<b>\$HASP093</b>	nn% Spool Utilization (JES2 V2)
<b>\$HASP095</b>	JES2 Catastrophic Error Or Abend
<b>\$HASP646</b>	nn% Spool Utilization (JES2 V3)

## Processing Flow

When any of the following messages are received, an ALERT is issued with an alarm. If a value is specified for the SYSPROG variable, a message is sent to the system programmer's TSO user ID.

<b>\$HASP050</b>	JES2 Resource Shortage
<b>\$HASP093</b>	nn% Spool Utilization (with greater than 80% utilization)
<b>\$HASP095</b>	JES2 Catastrophic Error Or Abend
<b>\$HASP355</b>	Spool Volumes Are Full
<b>\$HASP646</b>	nn% Spool Utilization (with greater than 80% utilization)

For the \$HASP095 and \$HASP355 messages, an outboard pager call is also made if all the following conditions are met:

- the outboard component is installed
- a value is supplied for MVSCALL, specifying the person who is paged
- a value is supplied for SYSBEEP, specifying the information that is sent to the pager

For \$HASP355, the command to drain the initiators is issued.

## EXECs

The EXECs for the JES2 Monitoring Solutions are:

<b>\$HASP050</b>	Handles \$HASP050 message
<b>\$HASP093</b>	Handles \$HASP093 message
<b>\$HASP095</b>	Handles \$HASP095 message
<b>\$HASP355</b>	Handles \$HASP355 message
<b>\$HASP646</b>	Handles \$HASP646 message

## Rules

The Rules for the JES2 Monitoring Solution are:

**\$HASP050**      Text-ID=\$HASP050

**\$HASP093**      Text-ID=\$HASP093

**\$HASP095**      Text-ID=\$HASP095

**\$HASP355**      Text-ID=\$HASP355

**\$HASP646**      Text-ID=\$HASP646

The Rules are distributed **DISABLED** in RULESET AAORULM1. You must **ENABLE** these Rules to implement this solution.

---

# Chapter 4 Storage Subsystems Solutions

The DASD and tape subsystems can be just as important to your data center's performance as the processor. The commands and replies used to communicate with the storage subsystems are not always easy to use. Shared DASD environments present an even greater challenge.

BMC Software provides the following Storage Subsystems Solutions:

## **Shared DASD Control**

Allows devices to be varied online/offline to all shared systems with one command

## **TLMS**

Provides a console interface to TLMS

## **Storage Reply**

Provides automatic reply to storage WTORs

## **Shared DASD Control Solution**

You must issue commands on each system in the shared DASD configuration to change the status of a volume. This can also require switching consoles.

The Shared DASD Control Solution allows the status of a shared DASD volume to be changed using one command. The status change is automatically propagated to the other systems in the configuration.

## Variables

For this solution to function properly in your environment, you must establish values for the following variables:

<b>SYSN</b>	number of MVS systems in the shared DASD configuration
<b>SYS1-SYSn</b>	names of the primary AutoOPERATOR subsystems on each MVS system in the shared DASD configuration

See Appendix D, “MVS Solution Variables” for information about initializing the variables and default settings.

## Invocation

This solution is operator-invoked. Enter

**%XSYSVARY P1 P2 P3**

on the COMMAND line to change the status of a shared DASD volume, where

<b>P1</b>	Is the UCB address
<b>P2</b>	Is the desired status (for example, online or offline)
<b>P3</b>	Should be SHR for 3480s

## Processing Flow

When the operator requests that the status of a device be changed, an EXEC is scheduled to execute on each system in the shared DASD complex.

The scheduled EXEC issues either the **VARY DEV,ONLINE** or **VARY DEV,OFFLINE** command depending on the operator specification.

## EXECs

The EXECs for the Shared DASD Control Solution are:

<b>XSYSVARY</b>	Schedule EXECs on all systems
<b>MST001C</b>	Issue VARY commands

## Rules

There are no Rules for the Shared DASD Control Solution.

## LMS Solution

Some data centers dedicate a console to the TLMS INQR task so the operator need not search constantly for the outstanding reply number using the `D R,L` command.

This solution lets TLMS commands be entered from the MCS console without knowing the outstanding reply number.

## Variables

This solution does not require any values to be set for variables.

## Invocation

This solution is operator-invoked. Enter

```
%TLMS P1 P2
```

on the `COMMAND` line to issue a TLMS command, where:

**P1** Is the TLMS command; valid values are DV, DVA, DVH, DVL, DVM, and DVR

**P2** Is the volume serial number

## Processing Flow

The **RESOLVE REPLIES** command is used to determine the outstanding reply number for the CAT2291D Message ID.

When the outstanding reply number is found, the TLMS command is issued using the command and volume specified by the operator.

## EXECs

The EXEC for the TLMS Solution is:

**TLMS** Finds outstanding reply, issues command

## Rules

There are no Rules for the TLMS Solution.

## Storage Reply Solution

The DASD and tape storage subsystems issue WTORS that require operator intervention. If these are not replied to in a timely manner, allocation queues can back up and cause degradation of system throughput. This solution ensures that the storage WTORS are replied to correctly and immediately.

## Variables

This solution does not require any values to be set for variables.

## Invocation

This solution is invoked by Rules for the following messages:

<b>IEC701D</b>	M ddd, Volume To Be Labeled ser
<b>IEF238D</b>	Reply Device Name, Wait or Cancel
<b>IEF433D</b>	Wait Requested - Reply Hold or Nohold

## Processing Flow

The following table shows the replies generated automatically for each WTOR.

**Table 4-1 WTOR Generated Replies**

<b>WTOR</b>	<b>Description</b>	<b>Reply</b>
IEC701D	M ddd, Volume To Be Labeled ser	M
IEF238D	Reply Device Name, Wait or Cancel	Wait
IEF433D	Wait Requested - Reply Hold or Nohold	Nohold

## EXECs

The EXECs for the Storage Reply Solution are:

**IEC701D**      Handles IEC701D Message

**IEF238D**      Handles IEF238D Message

**IEF433D**      Handles IEF433D Message

## Rules

The Rules for the Storage Reply Solution are:

**IEC701D**      Text-ID=IEC701D

**IEF238D**      Text-ID=IEF238D

**IEF433D**      Text-ID=IEF433D

The Rules are distributed **DISABLED** in RULESET AAORULM1. You must **ENABLE** these Rules to implement this solution.



---

# Chapter 5 Monitoring Solutions

The following Monitoring Solutions, provided by BMC Software, are designed to assist data center personnel in monitoring events occurring in the system:

**Dump Data Sets** Manages dump data sets to keep a minimum number available

**WTO Buffers** Resolves WTO buffer shortages

**SMF Data Sets** Manages dumping and switching of SMF data sets

**LOGREC** Manages clearing of SYS1.LOGREC

**RMF Monitor** Monitors status of RMF

## Dump Data Sets Monitoring Solution

This solution helps the system automatically manage its system dump data sets. It attempts to keep a minimum number of dump data sets free at all times.

### Variables

For this solution to function properly in your environment, you must establish values for the following variables:

**DUMPCLER** Name of task to clear dump data sets

**DUMPMF** Number of dump data sets to keep clear

**DUMPNDS** Total number of dump data sets

See Appendix D, “MVS Solution Variables” for information about initializing the variables and default settings.

## Invocation

This solution is invoked by Rules for the following messages:

- IEA911E** Complete/Partial Dump on SYS1.DUMPnn
- IEA994A** All Dump Data Sets Are Full And No SVC Dumps Can Be Taken
- IEA994E** All Allocated SYS1.DUMP Data Sets Are Full

This solution can also be operator-invoked. Type:

```
%@DUMPCLRO
```

to clear the oldest dump data set.

## Processing Flow

The following processing checks for a value specification for the variable **DUMPCLER**. If a value is not specified, processing ends.

When the **IEA911E Complete/Partial Dump on SYS1.DUMPnn** message is received, the current number of free dump data sets is compared to the minimum number that should be kept free. If the number of free data sets is less than the minimum, a task to clear the oldest dump data set is started.

When either of the following messages is received, a task to clear the oldest dump data set is started:

- IEA994A** All Dump Data Sets Are Full And No SVC Dumps Can Be Taken
- IEA994E** All Allocated SYS1.DUMP Data Sets Are Full

## EXECs

The EXECs for the Dump Data Sets Monitoring Solution are as follows:

<b>@DMPCLRO</b>	Starts task to clear oldest dump data set
<b>IEA911E</b>	Compares minimum free with current free data sets
<b>IEA994A</b>	Invokes @DMPCLRO
<b>IEA994E</b>	Invokes @DMPCLRO

## Rules

The Rules for the Dump Data Sets Monitoring Solution are as follows:

<b>IEA911E</b>	Text-ID=IEA911E
<b>IEA994A</b>	Text-ID=IEA994A
<b>IEA994E</b>	Text-ID=IEA994E

The Rules are distributed DISABLED in RULESET AAORULM1. You must ENABLE these Rules to implement this solution.

# WTO Buffers Monitoring Solution

This solution lets the system automatically resolve WTO buffer shortages.

## Variables

For this solution to function properly in your environment, you must establish values for the following variables:

<b>SYSPROG</b>	TSO userid of primary system programmer to receive warning messages
<b>SYSMVS</b>	Version, release, and modification level of the MVS system in use

See Appendix D, “MVS Solution Variables” for information about initializing the variables and default settings.

In addition to setting variables, the parameter CMDCON must be specified in member BBISSP00 of the BBPARM data set. This is required because the K Q command used to clear buffers cannot be issued from a subsystem console. Specifying CMDCON causes all commands issued without response to be issued with the CMDCON console ID.

## Invocation

This solution is invoked by Rules for the following messages:

<b>IEA404A</b>	Severe WTO Buffer Shortage - 100% Full
<b>IEA405E</b>	Severe WTO Buffer Shortage - 80% Full

## Processing Flow

When either the IEA404A or IEA405E buffer shortage message is received, an ALERT is generated. If a value was specified for SYSPROG, a warning message is sent to the primary system programmer.

The **D C** command is issued to determine the number of buffers in use by each console. The buffers are deleted using one of the following commands:

<b>K Q,L=console id</b>	(clearing console buffers)
<b>WRITELOG</b>	(clearing syslog buffers)

After the shortage is relieved, the ALERT is deleted. If a value is specified for SYSPROG, a message is sent to notify the primary system programmer that the shortage has been relieved.

## EXECs

The EXECs for the WTO Buffers Monitoring Solution are as follows:

<b>IEA404A</b>	Generate ALERTs, invoke IEE249I/IEE889I
<b>IEA405E</b>	Generate ALERTs, invoke IEE249I/IEE889I
<b>IEE249I</b>	Clear buffers (pre-MVS SP4)
<b>IEE889I</b>	Clear buffers (SP4 and later)

## Rules

The Rules for the WTO Buffers Monitoring Solution are as follows:

**IEA404A**      Text-ID=IEA404A

**IEA405E**      Text-ID=IEA405E

The Rules are distributed DISABLED in RULESET AAORULM1. You must ENABLE these Rules to implement this solution.

## SMF Data Set Monitoring Solution

This solution manages dumping and switching a system to automatically determine which other system in the shared DASD configuration is preventing access to a shared device.

## Variables

For this solution to function properly in your environment, you must establish values for the following variables:

**SMFALT**      Suffix of alternate SMF parameters

**SMFCLEAR**    Task name to dump SMF data sets

See Appendix D, “MVS Solution Variables” for information about initializing the variables and default settings.

## Invocation

This solution is invoked by Rules for the following SMF messages:

<b>IEE361I</b>	SMF Data Lost - No Data Sets Available
<b>IEE362A</b>	SMF Enter Dump For SYS1.MANx On ser
<b>IEE364I</b>	SMF (Logical/Physical) Error On SYS1.MANx
<b>IEE366I</b>	No SMF Data Sets Available - Data Being Buffered
<b>IEE391I</b>	SMF Enter Dump for Data Set on VOLSER ser, DSN=dsname
<b>IEE392I</b>	SMF Enter Dump for Data Set on VOLSER ser, DSN=dsname
<b>IEE393I</b>	SMF(LOGICAL PHYSICAL) I/O Error on dsname

## Processing Flow

The processing flow for the SMF Data Set Monitoring Solution is described as follows:

When the IEE362A, IEE391A or IEE392I SMF Enter Dump For SYS1.MANx ON SER message is received, the task to dump the data set (SMFCLEAR) is started. If a value was not specified for the SMFCLEAR variable, no processing occurs.

When the IEE949I or IEE974I message (output from D SMF command) is received, the task to dump the data set (SMFCLEAR) is started if it is not already active. If a value was not specified for the SMFCLEAR variable, no processing occurs.

When the IEE361I, IEE364I, IEE393I or IEE366I messages are received, the primary system programmer is notified if a value was specified for the SYSPROG variable.

If a value was specified for variable SMFALT, the operator receives an ALERT requesting that the alternate SMF parameters be switched. If the operator confirms the switch, the T SMF command is issued. The primary system programmer is notified of the switch if a value was specified for the SYSPROG variable.

## EXECs

The EXECs for the SMF Data Set Monitoring Solution are as follows:

<b>IEE361I</b>	Handles IEE361I message
<b>IEE362A</b>	Handles IEE362A message
<b>IEE364I</b>	Handles IEE364I message
<b>IEE366I</b>	Handles IEE366I message
<b>IEE391I</b>	Handles IEE391I message
<b>IEE392I</b>	Handles IEE392I message
<b>IEE393I</b>	Handles IEE393I message
<b>IEE949I</b>	Handles IEE949I message
<b>IEE974I</b>	Handles IEE974I message
<b>MMN001C</b>	Switches to alternate SMF parameters
<b>MMN004C</b>	Switches to alternate SMF parameters

## Rules

The Rules for the SMF Data Set Monitoring Solution are as follows:

<b>IEE361I</b>	Text-ID=IEE361I
<b>IEE362A</b>	Text-ID=IEE362A
<b>IEE364I</b>	Text-ID=IEE364I
<b>IEE366I</b>	Text-ID=IEE365I
<b>IEE391I</b>	Text-ID=IEE391I
<b>IEE392I</b>	Text-ID=IEE392I
<b>IEE393I</b>	Text-ID=EE393I
<b>IEE949I</b>	Text-ID=IEE949I
<b>IEE974I</b>	Text-ID=IEE974I

The Rules are distributed **DISABLED** in RULESET AAORULM1. You must **ENABLE** these Rules to implement this solution.

## LOGREC Data Set Monitoring Solution

This solution lets the system automatically manage the SYS1.LOGREC data set.

### Variables

For this solution to function properly in your environment, you must establish values for the following variables:

<b>LOGREC</b>	Task name to clear SYS1.LOGREC data set
<b>SYSPROG</b>	TSO user ID of primary system programmer to receive warning messages
<b>SYSBEEP</b>	Information to be placed on pager
<b>MVSCALL</b>	Name of the support person for Elan to page; this name must be defined on the Elan Workstation

See Appendix D, “MVS Solution Variables” for information about initializing the variables and default settings.

### Invocation

This solution is invoked by a Rule for the following messages:

<b>IFB040I</b>	SYS1.LOGREC Area Is Full
<b>IFB060E</b>	SYS1.LOGREC Near Full
<b>IFB070I</b>	Logrec Cannot Be Accessed. Record Is Lost

## Processing Flow

When either the IFB040I or IFB060E LOGREC FULL message is received and a value was specified for variable LOGREC, the following actions are taken:

- The task to clear SYS1.LOGREC is started.
- An ALERT is issued.
- If a value was specified for variable SYSPROG, a message is sent to the system programmer.

When the IFB070I Logrec Cannot Be Accessed message is received, the following actions are taken:

- An ALERT is issued.
- If a value was specified for variable SYSPROG, a notification is sent to the system programmer.
- If a value was specified for variable MVSCALL and SYSBEEP, an outboard pager call is made.

## EXECs

The EXECs for the LOGREC Data Set Monitoring Solution are as follows:

<b>IFB040I</b>	Handles IFB040I message
<b>IFB060E</b>	Handles IFB060E message
<b>IFB070I</b>	Handles IFB070I message

## Rules

The Rules for the LOGREC Data Set Monitoring Solution are as follows:

<b>IFB040I</b>	Text-ID=IFB040I
<b>IFB060E</b>	Text-ID=IFB060E
<b>IFB070I</b>	Text-ID=IFB070I

The Rules are distributed DISABLED in RULESET AAORULM1. You must ENABLE these Rules to implement this solution.

# RMF Monitoring Solution

This solution issues an ALERT if RMF ends.

## Variables

This solution does not require any values to be set for variables.

## Invocation

This solution is invoked by a Rule for the following message:

**ERB102I**      sid: Terminated

## Processing Flow

When the ERB102I message is received, an ALERT is issued.

## EXECs

The EXEC for the RMF Monitoring Solution is

**ERB102I**      Handles ERB102I message

## Rules

The Rule for the RMF Monitoring Solution is

**ERB102I**      Text-ID=ERB102I

The Rule is distributed DISABLED in RULESET AAORULM1. You must ENABLE this Rule to implement this solution.

---

# Chapter 6 Access Methods Solution

The Access Methods Solution is designed to alert the data center personnel when failures occur during I/O processing.

BMC Software provides the VSAM Failures Solution to alert the operator when a failure occurs in specified address spaces.

## VSAM Failures Solution

This solution alerts the operator when VSAM I/O fails in a critical address space or job stream.

### Variables

For this solution to function properly in your environment, edit the IDC3009I and IDC3351I EXECs to set values for the jobs where VSAM failures are critical.

<b>SYSPROG</b>	TSO user ID of primary system programmer to receive warning messages
<b>SYSBEEP</b>	information to be placed on pager
<b>MVSCALL</b>	name of the support person for Elan to page; this name must be defined on the Elan Workstation

See Appendix D, “MVS Solution Variables” for information on these variables.

## Invocation

This solution is invoked by Rules for the following text-IDs:

<b>IDC3009I</b>	VSAM CATALOG RETURN CODE IS rc
<b>IDC3351I</b>	VSAM OPEN CLOSE I/O RETURN CODE IS rc

## Processing Flow

When the message for either IDC3009I or IDC3351I is received, a comparison is made against the originating job name and the user-specified job names; if a match is found, an ALERT is issued to inform the operator of the failure.

If values for the SYSPROG, SYSBEEP, and MVSCALL variables were established during MAINVIEW AutoOPERATOR initialization, the processing continues by sending a TSO message to SYSPROG and by paging the SYSBEEP number through the outboard processor.

## EXECs

The EXECs for the VSAM Failures Solution are:

<b>IDC3009I</b>	Issues an ALERT
<b>IDC3351I</b>	Issues an ALERT

## Rules

The Rules for the VSAM Failures Solution are:

<b>IDC3009I</b>	Text-ID=IDC3009I
<b>IDC3351I</b>	Text-ID=IDC3351I

The Rules are distributed DISABLED in RULESET AAORULM1. You must ENABLE these Rules to implement this solution.

---

# Chapter 7 Job Scheduling Solution

The Job Scheduling Solution can be used to assist data center personnel in scheduling and managing time-dependent tasks. An alternative solution would be to use the MAINVIEW Total Object Manager (TOM) product. For more information on TOM, refer to the *MAINVIEW Total Object Manager User Guide*.

## Introduction

This solution consists of time-initiated EXECs that are scheduled at PAS startup time. These EXECs must be modified to contain the commands to start the time dependant task.

## Variables

None.

## Invocation

AAORULM1 contains disabled Rule PM00101. When this Rule Set and Rule ID are enabled at PAS initialization, this solution is invoked.

The appropriate EXECs must be modified with the necessary commands to start the various tasks.

## Processing Flow

AAOPRMxx is updated to enable Rule Set AAORULM1 and PM00101 is enabled at a cold start. The PAS is restarted with either the Rule Set and Rule already enabled from a previous invocation, or a cold start is done with AAORULM1 listed in AAOPRMxx.

## EXECs

The following EXECs can be used for the Job Scheduling Solution:

EXECs	Origin/Function
PM00101	scheduled from the PM0010I Rule in Rule Set AAORULM1
MSU002C	scheduled from within PM0010I at PAS initialization
MSU005C	scheduled from within MSU002C and schedules the various EXECs that set timer EXECs
MJSU001C	starts daily events
MJSU002C	starts Sunday events
MJSU003C	starts Monday events
MJSU004C	starts Tuesday events
MJSU005C	starts Wednesday events
MJSU006C	starts Thursday events
MJSU007C	starts Friday events
MJSU008C	starts Saturday events

## Rules

The Rule for the Job Scheduling Solution is

**PM00101** This Rule is distributed disabled in Rule Set AAORULM1. You need to enable this Rule to implement this solution.

---

# Chapter 8    **CONTROLM Solution**

This solution provides a command interface to CONTROL-M.

The operator uses an MVS MODIFY command to communicate with the CONTROL-M job scheduling software. This solution provides a command interface to CONTROL-M that alleviates the need for the MODIFY command.

## **Variables**

This solution does not require you to set values for any variables.

## **Invocation**

This solution is operator-invoked. Enter

**%SCHEDCOM P1**

on the COMMAND line to issue a CONTROL-M command, where

**P1**                    CONTROLM command; valid values are CTMX004 and  
NEWDEST

---

## Processing Flow

The following command is issued using the input parameter as the CONTROL-M command:

**F CONTROLM,cmd**

## EXECs

The EXEC for the CONTROLM Solution is

**SCHEDCOM** Issues the MODIFY command

## Rules

The CONTROLM Solution has no Rules.

---

# Chapter 9 MVS Performance Management Solutions

The Performance Management Solutions are designed to assist data center personnel monitor and adjust system parameters that affect system performance.

BMC Software provides the following three solutions:

## **Load Balancing**

lets the system dynamically adjust the job initiator structure based on performance considerations

## **DASD Reserve Analysis**

provides immediate insight into contention problems in a shared DASD environment

## **Exception Monitoring**

extends the capabilities of the RESOLVE Advanced Early Warning (AEW) system

## **Load Balancing Solution**

As workload type and activity on your system changes, you might want to adjust your job initiator configuration. For example, you might want to take advantage of times when the system resources support additional initiators without adversely affecting system performance. Or you might want to limit access to the system at times when resources are constrained.

This solution lets the system automatically increase or decrease its workload (initiators) based on current system performance and job demand.

## Variables

For this solution to function properly in your environment, you must establish values for the following variables:

<b>LBGPAGE</b>	Paging rate at which an attempt is made to decrease workload
<b>LBGCPUH</b>	CPU utilization percentage at which an attempt is made to decrease workload
<b>LBGCPUL</b>	CPU utilization percentage at which an attempt is made to increase workload
<b>LBGIBEG</b>	Beginning initiator number to manage
<b>LBGIEND</b>	Ending initiator number to manage
<b>LBGIINC</b>	Number of initiators to start in an increase situation
<b>LBGIDEC</b>	Number of initiators to stop in an decrease situation
<b>LBGCLAS</b>	Class priority list from high to low

See Appendix D, “MVS Solution Variables” for information about initializing the variables and default settings.

## Invocation

This solution is operator-invoked. Type

**%LBSTART**

on the **COMMAND** line to begin load balancing on your system.

## Processing Flow

Every five minutes, this solution compares the current CPU utilization and paging rate to the threshold variables. If any threshold is crossed 3 times within 15 minutes, an action is taken.

If CPU utilization is below the low CPU threshold, an attempt is made to increase the workload by starting additional initiators.

Initiators (up to the maximum specified by variable `LBGIINC`) that have a status of either `DRAINED` or `HALTED` are started. Classes are assigned to the initiators based on the class priority list (variable `LBGCLAS`) and work waiting to execute.

If the CPU utilization is above the high CPU threshold or the page rate is above the page rate threshold, an attempt is made to decrease the workload by stopping initiators. Initiators (up to the maximum specified by variable `LBGIDEC`) that have a status of either active or `INACTIVE` are purged or drained.

## EXECs

Following are the EXECs for the Load Balancing Solution:

<b>LBSTART</b>	Initializes CPU and page threshold counters, calls MPE003C to build initiator table, schedules MPE004C to execute in five minutes
<b>MPE003C</b>	Retrieves initiator information from JES
<b>MPE004C</b>	Manages CPU and page threshold counters, schedules MPE005C to execute
<b>MPE005C</b>	Compares current CPU and page values against thresholds (if adjustment is necessary), and schedules MPE006C to execute or schedules MPE004C to execute in five minutes
<b>MPE006C</b>	Starts/stops initiators and schedules MPE004C to execute again in five minutes

## Rules

There are no Rules for the Load Balancing Solution.

# DASD Reserve Analysis Solution

In shared DASD environments, one or more systems can be prevented from accessing an entire DASD volume due to hardware reserves by another system. If a reserve remains in effect for an extended period of time, end-user response time can be severely affected.

This solution lets the system automatically determine which other system in the shared DASD configuration is preventing access to a shared device.

## Variables

For this solution to function properly in your environment, you must establish values for the following variables:

<b>SYSN</b>	Number of MVS systems in the shared DASD configuration
<b>SYS1-SYSn</b>	Names of the primary MAINVIEW AutoOPERATOR subsystem on each MVS system in the shared DASD configuration

See Appendix D, “MVS Solution Variables” for information about initializing the variables and default settings.

## Invocation

This solution is invoked by a Rule for the following IOS message:

```
IOS071I UCB, CHPID, JOBNAME, START PENDING
```

## Processing Flow

When the IOS071I message is received, a remote EXEC is scheduled on every other system in the shared DASD complex.

Each of the other systems use the `RESOLVE RESERVE` command to examine the DASD reserves it is currently holding. Any system that has a device reserved issues an operator `ALERT` back to the system that is being prevented access to the device.

## EXECs

The EXECs for the DASD Reserve Analysis Solution are

<b>IOS071I</b>	Schedules remote EXECs
<b>MPE001C</b>	Examines currently held reserves
<b>MPE002C</b>	Issues ALERT

## Rules

The Rule for the DASD Reserve Analysis Solution is

**IOS071I**      Text-ID=IOS071I

The Rule is distributed **DISABLED** in RULESET AAORULM1. You must **ENABLE** this Rule to implement this solution.

# Exception Monitoring Solution

If you use the BMC Software MAINVIEW SYSPROG Services product, you might want to extend its Advanced Early Warning (AEW) capabilities.

This solution lets RESOLVE AEW messages be posted to the MAINVIEW AutoOPERATOR ALERT application and, in some instances, provides information about the causes of an exception condition.

## Variables

For this solution to function properly in your environment, you must establish values for the following variables:

<b>MAXCCPU</b>	Maximum complex CPU utilization percentage
<b>MAXTCPU</b>	Maximum CPU utilization by TSO address spaces
<b>MAXBCSA</b>	Maximum CSA utilization percentage below the 16M line
<b>MAXACSA</b>	Maximum extended CSA utilization percentage
<b>MAXDEVU</b>	Maximum device utilization percentage

See Appendix D, “MVS Solution Variables” for information about initializing the variables and default settings.

## Invocation

The entire RESOLVE PLUS software is required to use the MVS Performance Monitoring Solution.

To invoke the Exception Monitoring Solution:

**Step 1** ENABLE the following Rules in the AAORULM1 Rule Set:

```
PWSCPU00 *WARNING* CPU 1 USAGE IS (x)%  
or  
PWSCPU01 *WARNING* CPU COMPLEX USAGE IS (x)%  
  
PWSCPU02 *WARNING* CPU USAGE IS (x)% FOR TSO  
  
PWSCSA01 *WARNING* CSA/ECSA USAGE IS (x)%; (x)K ARE FREE  
  
PWSDEV01 *WARNING* DVN VOLUME USAGE IS (x)%
```

**Step 2** To install these solutions

**2.A** Activate the AEW sampler.

```
COPY <PREFIX>.BBPARAM(PWSCPMZZ) TO  
<PREFIX>.UBBPARM,DISP=SHR
```

**2.B** Add the following statement to your BBI-SS JCL to activate the PWSCPM00 Rule.

```
//LIB DD DSN=<ORPREFIX>.UBBPARM,DISP=SHR
```

Restart your BBI subsystem after making those changes.

**Step 3** Set the variables for these solutions in <PREFIX>.UBBPROC member MSU002C.

Additional information is provided in MSU002C and PWSCPM00.

**Step 4** ENABLE the Rules in RULESET AAORULM1.

If you have only one CPU in the system in which you are running this BBI subsystem, ENABLE Rule PWSCPU00.

If there are multiple CPUs running, ENABLE Rule PWSCPU01.

Issue a RESOLVE CPU command to check the number of CPUs running in your system.

## Processing Flow

<b>Caution</b>
Do not ENABLE both PWSCPU00 and PWSCPU01. These two Rules schedule EXECs that use the same variables in EXEC MSU002C and are mutually exclusive.

If either of the PWSCPU messages is received three or more times within 10 minutes, the CPU value in the message is compared against the appropriate CPU threshold variable. If the threshold is exceeded, an ALERT is issued and an additional monitor (\$TOTCPU or \$TSOCPU, depending on which threshold is exceeded) is invoked.

The \$TOTCPU and \$TSOCPU EXECs issue the RESOLVE CPU command to determine who is the current major user of the CPU (\$TSOCPU limits the search to TSO users only). The EXECs then issue ALERTs to give the operator insight into which users are responsible for the exception condition.

When either the PWSCSA01 or PWSDEV01 message is received, the value in the message is compared against the appropriate threshold variable. If the threshold is exceeded, an ALERT is issued.

## EXECs

The EXECs for the Exception Monitoring Solution are

<b>PWSCPU00</b>	monitors MAXCCPU threshold
<b>PWSCPU01</b>	monitors MAXCCPU threshold
<b>PWSCPU02</b>	monitors MAXTCPU threshold
<b>PWSCSA01</b>	monitors MAXBCSA and MAXACSA thresholds
<b>PWSDEV01</b>	monitors MAXDEVU threshold

## Rules

The Rules for the Exception Monitoring Solution are

**PWSCPU00**    Text-ID=PWSCPU00

**PWSCPU01**    Text-ID=PWSCPU01

**PWSCPU02**    Text-ID=PWSCPU02

**PWSCSA01**    Text-ID=PWSCSA01

**PWSDEV01**    Text-ID=PWSDEV01

The Rules are distributed **DISABLED** in RULESET AAORULM1. You must **ENABLE** these Rules to implement this solution.

---

# Chapter 10 Error Recovery Solutions

Critical messages from CICS can be lost among the message traffic at the operator console. Also, replies to those messages and the follow-up actions taken might not be consistent.

The Error Recovery Solutions assist data center personnel when critical errors occur in the CICS environment.

BMC Software provides the following Error Recovery Solutions:

**Storage Violation**

Notifies a central operator of storage violations

**CICS Abnormal Termination**

Notifies a central operator of any CICS region abend

**VSAM Subtask Abnormal Termination**

Notifies a central operator of a VSAM subtask abend and automatically replies to the abnormal WTOR

**Temporary Storage Data Set Extension Failure**

Notifies a central operator and provides a follow-up display for investigating the problem

**Terminal Errors**

Resets terminal status after a terminal error occurs

# Storage Violation Solution

The Storage Violation Solution notifies an operator when a storage violation has occurred.

## Variables

For the Storage Violation Solution to function properly in your environment, you must establish values for the following variables:

<b>FOCALPT</b>	SSID of MAINVIEW AutoOPERATOR subsystem to receive all ALERTs
<b>ONLSYSN</b>	Job name of the target system
<b>ONLCNTN</b>	Subsystem ID of the controlling BBI-SS
<b>ONLALTN</b>	Color for ALERT messages (can be BLUE, WHITE, RED, or GREEN)
<b>ONLYTPN</b>	Type of target or BBI-SS (can be CICS, DB2, IMS, CICS AO, or IMS AO)
<b>ONLALMN</b>	Indicates whether an alarm sounds (can be Y or N)

See Appendix E, “CICS Solution Variables” for information on default settings and initializing the variables.

## Invocation

This solution is invoked by the Rule for the following text-ID:

**DFH0508** A STORAGE VIOLATION HAS OCCURRED

## Processing Flow

When the DFH0508 message is received, the ALERT

```
*STORAGE VIOLATION IN CICS cicsid, REPLY YES TO CANCEL
```

is sent to ask the operator to confirm cancellation of the CICS region that suffered the storage violation. If the operator replies yes, the region is cancelled and this message is sent:

```
*STORAGE VIOLATION IN CICS cicsid, BEING CANCELED NOW
```

To modify the solution to automatically cancel the region without asking for operator confirmation, edit BBPROC member DFH0508. When automatic cancellation is in effect, the ALERT BEING CANCELED NOW is sent.

## EXECs

The EXECs for the Storage Violation Solution are

**DFH0508**      Handles DFH0508 message

**CER001C**      Cancels the CICS system after a storage violation

## Rules

The Rules for the Storage Violation Solution are

**DFH0508**      Text-ID=DFH0508

**DFHSM010**    Test-ID=DFHSM010

The Rules are distributed DISABLED in RULESET AAORULC1. You must ENABLE these Rules to implement this solution. You must ENABLE DFHSM010 for CICS/ESA.

# CICS Abnormal Termination Solution

The CICS Abnormal Termination Solution notifies an operator when an abend in a CICS region has been detected.

## Variables

For the CICS Abnormal Termination Solution to function properly in your environment, you must establish values for the following variables:

<b>FOCALPT</b>	SSID of MAINVIEW AutoOPERATOR subsystem to receive all ALERTs
<b>ONLSYSN</b>	Job name of the target system
<b>ONLCNTN</b>	Subsystem ID of the controlling BBI-SS
<b>ONLALTN</b>	Color for ALERT messages (can be BLUE, WHITE, RED, or GREEN)
<b>ONLYTPN</b>	Type of target or BBI-SS (can be CICS, DB2, IMS, CICS AO, or IMS AO)
<b>ONLALMN</b>	Indicates whether an alarm sounds (can be Y or N)

See Appendix E, “CICS Solution Variables” for information on default settings and initializing the variables.

## Invocation

This solution is invoked by the Rule for the following text-ID:

**DFH0606** ABEND xxxx - xxxx HAS BEEN DETECTED

## Processing Flow

In response to the DFH0606 message, an ALERT is sent to the FOCAL POINT target to indicate a catastrophic abend ended a CICS system.

## EXECs

The EXEC for the CICS Abnormal Termination Solution is

**DFH0606**      Handles DFH0606 message

## Rules

The Rule for the CICS Abnormal Termination is

**DFH0606**      Text-ID=DFH0606

The Rule is distributed **DISABLED** in RULESET AAORULC1. You must **ENABLE** this Rule to implement this solution.

# VSAM Subtask Abnormal Termination Solution

The VSAM Subtask Abnormal Termination Solution automatically notifies an operator when a VSAM subtask abends.

## Variables

For the VSAM Subtask Abnormal Termination Solution to function properly in your environment, you must establish values for the following variables:

<b>FOCALPT</b>	SSID of MAINVIEW AutoOPERATOR subsystem to receive all ALERTs
<b>ONLSYSN</b>	Job name of the target system
<b>ONLCNTN</b>	Subsystem ID of the controlling BBI-SS
<b>ONLALTN</b>	Color for ALERT messages (can be BLUE, WHITE, RED, or GREEN)
<b>ONLYTPN</b>	Type of target or BBI-SS (can be CICS, DB2, IMS, CICS AO, or IMS AO)
<b>ONLALMN</b>	Indicates whether an alarm sounds (can be Y or N)

See Appendix E, “CICS Solution Variables” for information on default settings and initializing the variables.

## Invocation

This solution is invoked by the Rule for the following text-ID:

**DFH0901** VSAM SUBTASK ABEND - DO YOU WANT TO CONTINUE  
IN A DEGRADED MODE OR ABEND? REPLY GO OR  
CANCEL.

**Note:** This solution cannot be used for CICS/ESA.

## Processing Flow

When the DFH0901 WTOR is received, this solution replies GO to let CICS continue initialization and the following ALERT message is sent:

```
*VSAM SUBTASK ABEND, CICS CONTINUING IN DEGRADED MODE
```

To modify the solution to reply CANCEL instead of GO, edit BBPROC member DFH0901. When automatic cancel is in effect, the solution replies CANCEL to the DFH0901 WTOR and issues the ALERT

```
*VSAM SUBTASK ABEND, CICS BEING CANCELLED AUTOMATICALLY
```

## EXECs

The EXEC for the VSAM Subtask Abnormal Termination Solution is

**DFH0901** Handles DFH0901 message

## Rules

The Rule for the VSAM Subtask Abnormal Termination Solution is

**DFH0901** Text-ID=DFH0901

The Rule is distributed DISABLED in RULESET AAORULC1. You must ENABLE this Rule to implement this solution.

# CICS Temporary Storage Suspensions Solution (CICSTSS)

This solution detects and automatically corrects suspensions of CICS Temporary Storage processing caused when Auxiliary Temporary Storage data set is full.

When that is the case, CICS suspends processing for many transactions—not only transactions that are writing to temporary storage directly but many transactions (including system functions) that use temporary storage indirectly are also affected. For example, the EXEC CICS START command with the DATA parameter causes CICS to use temporary storage to hold the data being passed to the Started Task.

## Variables

This solution does not use any specific variable requiring special initialization. However, variables describing the CICS systems, as explained in Appendix E, “CICS Solution Variables”, should be set.

## Invocation

The EXECs in this solution are scheduled when the following message is received:

```
FT068*          NNNNNN TEMPORARY STORAGE SUSPENSIONS HAVE  
                  OCCURRED
```

## Processing Flow

All EXECs in this solution are message-driven. The EXECs create ALERT messages that may require an operator reply to initiate a follow-up EXEC, or may include extended help or follow-up commands. The IND column of the Alert Detail Display shows which options are available for each ALERT.

If an ALERT message has an E in the IND column, a follow-up EXEC is assigned. The message text (or extended help panel) explains what reply is requested. You should enter the message text in the LC column of the ALERT display to schedule the follow-on EXEC.

If an ALERT message has a H in the IND column, an Extended Help Panel is assigned. You should type the EXPAND command and press **Enter** (or use the EXPAND PFK, if assigned) after placing the cursor anywhere in the ALERT message text to display the help panel.

If an ALERT message has a C in the IND column, a follow-up Primary Command is assigned. You should type the TRANSFER command and press **Enter** (or use the TRANSFER PFK, if assigned) after placing the cursor anywhere in the ALERT message text to invoke the command. The follow-up command transfers you to a display with more information to help solve the problem.

## EXECs

The EXECs for the CICSTSS Solution are

- |                |  |
|----------------|--|
| <b>FT068</b>   | Starts follow-up of temporary storage suspensions by scheduling CER004C and setting a timer for CER003C.   |
| <b>CER003C</b> | Executes the CICS MANAGER Problem Display and looks for the FT068 message. If found, it compares the count in the message to the count saved in a variable. If the count has increased, the investigation EXECs are kicked off.  |
| <b>CER004C</b> | Executes the CICS MANAGER Task Display and purges any tasks waiting on temporary storage processing (ATSP).  |
| <b>CER005C</b> | Executes the CICS MANAGER Temporary Storage Unit Table (TSUT) display and purges some Auxiliary TSUT entries. It first purges any entries that start with CEBR, which typically contain transaction dump output. If the TSP suspension persists, on the next iteration, CER005C purges the largest TSUT entry. The largest entry is determined by examining the Data Length and PUTQ count fields. |
| <b>CER006C</b> | Handles the reply from the ALERT created by CER004C.   |

## Rules

There is one Rule for each of the messages listed in “Invocation” on page 10-7. The Rule-IDs equal the message-IDs.

The Rules are distributed DISABLED in RULESET AAORULC1. You must ENABLE the following Rules to implement this solution: CICSTART, CICSTERM, FT426W, FT425W, and FT435I.

# Temporary Storage (TS) Data Set Extension Failure Solution

The Temporary Storage (TS) Data Set Extension Failure Solution notifies a central operator and provides a follow-up display for investigating the problem.

## Variables

For the Temporary Storage Data Set Extension Failure Solution to function properly in your environment, you must establish values for the following variables:

<b>FOCALPT</b>	SSID of MAINVIEW AutoOPERATOR subsystem to receive all ALERTs
<b>ONLSYSN</b>	Job name of the target system
<b>ONLCNTN</b>	Subsystem ID of the controlling BBI-SS
<b>ONLALTN</b>	Color for ALERT messages (can be BLUE, WHITE, RED, or GREEN)
<b>ONLYTPN</b>	Type of target or BBI-SS (can be CICS, DB2, IMS, CICS AO, or IMS AO)
<b>ONLALMN</b>	Indicates whether an alarm sounds (can be Y or N)

See Appendix E, “CICS Solution Variables” for information on default settings and initializing the variables.

## Invocation

This solution is invoked by the Rule for the following text-ID:

<b>DFH1311</b>	TEMPORARY STORAGE DATA SET IS FULL AND CANNOT BE EXTENDED
----------------	---

## Processing Flow

In response to DFH1311, an ALERT is issued. In addition, the CICS MANAGER TEMPSTOR command is associated with the ALERT. The operator can use the command to map the temporary storage usage.

## EXECs

The EXEC for the Temporary Storage Data Set Extension Failure Solution is

**DFH1311**      Handles DFH1311 message

## Rules

The Rule for the Temporary Storage Data Set Extension Failure Solution is

**DFH1311**      Text-ID=DFH1311

The Rule is distributed **DISABLED** in RULESET AAORULC1. You must **ENABLE** this Rule to implement this solution.

# Terminal Errors Solution

The Terminal Errors Solution resets the terminal status after a terminal error occurs.

## Variables

For the Terminal Errors Solution to function properly in your environment, you must establish values for the following variables:

<b>FOCALPT</b>	SSID of MAINVIEW AutoOPERATOR subsystem to receive all ALERTs
<b>ONLSYSN</b>	Job name of the target system
<b>ONLCNTN</b>	Subsystem ID of the controlling BBI-SS
<b>ONLALTN</b>	Color for ALERT messages (can be BLUE, WHITE, RED, or GREEN)
<b>ONLYTPN</b>	Type of target or BBI-SS (can be CICS, DB2, IMS, CICS AO, or IMS AO)
<b>ONLALMN</b>	Indicates whether an alarm sounds (can be Y or N)

See Appendix E, “CICS Solution Variables” for information on default settings and initializing the variables.





---

# Chapter 11 CICS Performance Management Solutions

The CICS Performance Management Solutions help data center personnel achieve maximum availability and increased response time.

BMC Software provides the following Performance Management Solutions:

**File Degradation Analysis**

Automatically analyzes factors causing poor response time for a CICS file

**File Allocation - Deallocation**

Lets you allocate, enable, and open, or lets you close, disable, and deallocate files from a target CICS system

**VSAM Control Area Split Monitor**

Notifies the operator of VSAM Control Area split problems

**Transaction Response Time Monitor**

Notifies the operator when transactions are exceeding response time objectives

## File Degradation Analysis Solution

Most CICS response time problems are caused by I/O performance problems. High I/O service times, as reported by the CICS MANAGER Background Problem Service, are presented on various displays and logged to the BBI-SS Journal. Constant investigation of I/O problems can be very time consuming. Further, because most I/O contention problems are transient, factors causing a problem seem to disappear before an investigation is started.

This solution automatically analyzes the factors causing disk I/O performance problems that affect CICS response time.

**Note:** This solution requires RESOLVE services.

## Variables

For the File Degradation Analysis Solution to function properly in your environment, you must establish values for the following variables:

<b>FOCALPT</b>	SSID of MAINVIEW AutoOPERATOR subsystem to receive all ALERTs
<b>ONLSYSN</b>	Job name of the target system
<b>ONLCNTN</b>	Subsystem ID of the controlling BBI-SS
<b>ONLALTN</b>	Color for ALERT messages (can be BLUE, WHITE, RED, or GREEN)
<b>ONLYTPN</b>	Type of target or BBI-SS (can be CICS, DB2, IMS, CICS AO, or IMS AO)
<b>ONLALMN</b>	Indicates whether an alarm sounds (can be Y or N)

See Appendix E, “CICS Solution Variables” for information on default settings and initializing the variables.

## Invocation

This solution is invoked by a Rule for the following text-ID:

**FT095W** FILE filename EXCEEDED SERVICE TIME, n.nnn IS AVERAGE SERVICE TIME

## Processing Flow

When the FT095W message is received, contention for logical and physical resources in the system where the target CICS is executing is analyzed using CICS MANAGER and RESOLVE services. EXECs also are scheduled to look for any contention from other systems in the shared DASD environments. Results of the analysis are written in the MAINVIEW AutoOPERATOR Subsystem Log.

All results messages written to the MAINVIEW AutoOPERATOR Subsystem log are prefixed with a message-ID. The message-ID consists of a fixed literal (CAF#), followed by the EXEC sequence number (IMFEID); for example, CAF#1234. Use the message-ID to search for all messages from a given file analysis in the log, even when they arrive from different systems at different times and are intermixed with other messages in the SS Log.

The following messages are written to the MAINVIEW AutoOPERATOR Subsystem Log:

```
STARTING FILE DEGRADATION ANALYSIS FOLLOWING FT095W
FOR FILE filename IN CICS SYSTEM cicsid (smfid)
ON VOLUME volser ON UNIT ucb WITH nnnnn EXCPS.
```

```
DATASET dsname IS ALSO ALLOCATED TO JOB(S):
jobname1, jobname2, .... ON SYSTEM smfid.
```

```
NO OTHER JOBS ON SYSTEM smfid ARE USING DATASET dsname
IN CICS SYSTEM cicsid.
```

```
DEVICE ucb IS LESS THAN n% BUSY ON SYSTEM smfid.
```

```
DEVICE ucb WAS OBSERVED nnn% BUSY ON SYSTEM smfid
WITH AN I/O RATE OF nnn PER SECOND
AND AN AVERAGE WAIT TIME OF n.nnn MS.
```

```
nnn% OF THE ACTIVITY ON ucb IS FROM JOB jobname.
```

```
DEGRADATION ANALYSIS COMPLETE FOR FILE filename
ON SYSTEM smfid, RESULTS FROM OTHER SYSTEM(S)
WILL FOLLOW IN THE LOG WITH THE SAME MESSAGE ID.
```

## EXECs

The EXECs for the File Degradation Analysis Solution are

<b>FT095W</b>	As primary EXEC, schedules all secondary CPEnnnx EXECs that follow
<b>CPE001C</b>	Searches for the DD name corresponding to the filename when the file is an IMS database
<b>CPE002C</b>	Reports on other jobs allocated to the CICS file data set name
<b>CPE003C</b>	Reports on other jobs using the device where the CICS file is located
<b>CPE004C</b>	Gets the data set name, volser number, and UCB for the CICS filename using the RESOLVE TIOT command

## Rules

The Rule for the File Degradation Analysis Solution is

**FT095W**      Text-ID=FT095W

The Rule is distributed DISABLED in RULESET AAORULC1. In addition to this Rule, the following Rules must be ENABLED to implement this solution: CICSTART, CICSTERM, FT426W, FT425W, and FT435I.

## File Allocation - Deallocation Solution

The batch window (the interval between the time CICS is stopped and started again) might be too short to accommodate the necessary batch jobs to extract, update, and back up files used by CICS. If the window is too small, the operator must enter commands to close, disable, and free, and then later allocate, enable, and open files. This takes time, is tedious, and is prone to error. Syntax errors or a missed step in a complicated procedure can cause resources to be unavailable for use by CICS transactions.

This solution lets batch jobs allocate and deallocate files from a running CICS system. This lets files be freed from CICS, processed by other jobs, and returned to CICS, while CICS remains available for other users.

## Variables and JCL

For the File Allocation - Deallocation Solution to function properly in your environment, you must establish values for the following variables:

<b>FOCALPT</b>	SSID of MAINVIEW AutoOPERATOR subsystem to receive all ALERTs
<b>ONLSYSN</b>	Job name of the target system
<b>ONLCNTN</b>	Subsystem ID of the controlling BBI-SS
<b>ONLALTN</b>	Color for ALERT messages (can be BLUE, WHITE, RED, or GREEN)
<b>ONLTYPN</b>	Type of target or BBI-SS (can be CICS, DB2, IMS, CICS AO, or IMS AO)
<b>ONLALMN</b>	Indicates whether an alarm sounds (can be Y or N)

See Appendix E, “CICS Solution Variables” for information on default settings and initializing the variables.

## Invocation

Most aspects of this solution are invoked from batch JCL. Edit two sample JCL members distributed in the BBUSER data set to specify file names and CICS region names.

**CAOALLOC** Allocate, enable, and open a file in a CICS system from a batch job

**CAODEALO** Close, disable, and free a file in a CICS system from a batch job

The remaining aspects of this solution are invoked by Rules for the following text-IDs:

**FT401E** ALLOC FAILED RC=XX EC=XX FILE filename DSN dsname

**FT402I** ALLOC COMPLETE FOR FILE filename DSN dsname

**FT403E** DEALLOC FAILED NOT ALLOCATED FILE filename

**FT404I** DEALLOC COMPLETE FOR FILE filename

## Processing Flow

When the batch job CAOALLOC is run, the file you specified is allocated, enabled, and opened on the CICS system that you specified.

When the batch job CAODEALO is run, the file you specified is closed, disabled, and freed on the CICS system that you specified.

## EXECs

The EXECs for the File Allocation-Deallocation Solution are

<b>CPEALLOC</b>	Allocates, enables, and opens a file in a CICS system
<b>CPEDEALO</b>	Closes, disables, and deallocates a file in a CICS system
<b>FT401E</b>	Stores the result from IMFEXEC ALLOC in a shared variable
<b>FT402I</b>	Stores the result from IMFEXEC ALLOC in a shared variable
<b>FT403E</b>	Stores the result from IMFEXEC FREE in a shared variable
<b>FT404I</b>	Stores the result from IMFEXEC FREE in a shared variable

## Rules

The Rules for the File Allocation-Deallocation Solution are

<b>FT401E</b>	Text-ID=FT401E
<b>FT402I</b>	Text-ID=FT402I
<b>FT403E</b>	Text-ID=FT403E
<b>FT404I</b>	Text-ID=FT404I

The Rules are distributed DISABLED in RULESET AAORULC1. In addition to these Rules, you must ENABLE the following Rules: CICSTART, CICSTERM, FT426W, FT425W, and FT435I.

# VSAM Control Area Split Monitor Solution

VSAM Control Area splits can cause I/O response times to be extended. This solution assists the operator in monitoring which files are suffering Control Area splits.

## Variables and JCL

For the VSAM Control Area Split Monitor Solution to function properly in your environment, you must establish values for the following variables:

<b>FOCALPT</b>	SSID of MAINVIEW AutoOPERATOR subsystem to receive all ALERTs
<b>ONLSYSN</b>	Job name of the target system
<b>ONLCNTN</b>	Subsystem ID of the controlling BBI-SS
<b>ONLALTN</b>	Color for ALERT messages (can be BLUE, WHITE, RED, or GREEN)
<b>ONLYTPN</b>	Type of target or BBI-SS (can be CICS, DB2, IMS, CICS AO, or IMS AO)
<b>ONLALMN</b>	Indicates whether an alarm sounds (can be Y or N)

See Appendix E, “CICS Solution Variables” for information on default settings and initializing the variables.

## Invocation

This solution is invoked by a Rule for the following text-ID:

**FT092S** FILE filename HAS HAD nnn CONTROL AREA SPLITS

## Processing Flow

In response to the FT092S message, an ALERT is sent, telling the operator what to do about a VSAM file with several Control Area splits:

```
*FT092S FILE file SHOULD BE RE-DEFINED (nnn C.A. SPLITS)
```

In addition, the CICS MANAGER FILEXPND command is associated with the ALERT. The operator can use the command to display the status of the VSAM cluster.

## EXECs

The EXEC for the VSAM Control Area Split Monitor Solution is

**FT092S**            Handles FT092S message

## Rules

The Rule for the VSAM Control Area Split Monitor Solution is

**FT092S**            Text-ID=FT092S

The Rule is distributed DISABLED in RULESET AAORULC1. In addition to this Rule, you must implement the following Rules: CICSTART, CICSTERM, FT426W, FT425W, and FT435I.

# Transaction Response Time Monitor Solution

This solution ensures that operators are aware of transactions exceeding service level objectives.

## Variables and JCL

For the Transaction Response Time Monitor Solution to function properly in your environment, you must establish values for the following variables:

<b>FOCALPT</b>	SSID of MAINVIEW AutoOPERATOR subsystem to receive all ALERTs
<b>ONLSYSN</b>	Job name of the target system
<b>ONLCNTN</b>	Subsystem ID of the controlling BBI-SS
<b>ONLALTN</b>	Color for ALERT messages (can be BLUE, WHITE, RED, or GREEN)
<b>ONLTYPN</b>	Type of target or BBI-SS (can be CICS, DB2, IMS, CICS AO, or IMS AO)
<b>ONLALMN</b>	Indicates whether an alarm sounds (can be Y or N)

See Appendix E, “CICS Solution Variables” for information on default settings and initializing the variables.

## Invocation

This solution is invoked by a Rule for the following text-ID:

**FT094W**        TRAN tranid HAS EXCEEDED SERVICE LEVEL, nn.nnn IS  
                  AVG REPSONSE

## Processing Flow

In response to FT094W, the ALERT message is sent:

```
*SERVICE LEVELS EXCEEDED, ENTER ON THIS MSG TO SEE  
ACTIVITY
```

In addition, the CICS MANAGER MONITOR command is associated with the ALERT. The operator can use the command to display service level information in the CICS region.

## EXECs

The EXEC for the Transaction Response Time Monitor Solution is

**FT094W**        Handles FT094W message

## Rules

The Rule for the Transaction Response Time Monitor Solution is

**FT094W**        Text-ID=FT094W

The Rule is distributed DISABLED in RULESET AAORULC1. In addition to this Rule, you must implement the following Rules: CICSTART, CICSTERM, FT426W, FT425W, and FT435I.



---

# Chapter 12 VTAM/NCP Solution

Much of the work in data centers is performed interactively using online systems and terminal networks. However, the commands and replies used to communicate with the network software can be cumbersome to use.

BMC Software provides the TCAS Reply Solution, which provides automatic replies to TCAS WTORS.

## TCAS Reply Solution

When TCAS ends either normally or abnormally, WTORS are issued. If these WTORS are not replied to correctly and in a timely manner, the down time for TSO can be extended unnecessarily. This solution ensures that the TCAS WTORS are replied to correctly and immediately.

### Variables

This solution does not require any values to be set for variables.

### Invocation

This solution is invoked by Rules for the following TCAS messages:

<b>IKT001D</b>	nnn Users Active Reply U, SIC or FSTOP
<b>IKT010D</b>	nnn Users Active Reply SIC or FSTOP
<b>IKT012D</b>	TCAS termination in progress - specify U or DUMP

## Processing Flow

When either of the following WTORs is received, a reply of SIC is generated:

- IKT001D**      nnn Users Active Reply U, SIC or FSTOP  
(received when TCAS is started after abnormal termination)
- IKT010D**      nnn Users Active Reply SIC or FSTOP  
(received when TCAS is stopped)

When the IKT012D TCAS Termination In Progress - Specify U or DUMP WTOR is received, a reply of U is generated.

## EXECs

The EXECs for the TCAS Reply Solution are

- IKT001D**      Handle IKT001D message
- IKT010D**      Handle IKT010D message
- IKT012D**      Handle IKT012D message

## Rules

The Rules for the TCAS Reply Solution are

- IKT001D**      Text-ID=IKT001D
- IKT010D**      Text-ID=IKT010D
- IKT012D**      Text-ID=IKT012D

The Rules are distributed DISABLED in RULESET AAORULM1. You must ENABLE these Rules to implement this solution.

---

# Chapter 13 Initial Customization Steps

Before you can take advantage of the automation features of the DB2 Solutions, you must customize your automation environment to accommodate them. You can do this by implementing the solutions described in Chapter 14, “DB2 Global Operations Solutions”. You can then successfully use the other DB2 Solutions. Or you can follow these steps to customize your automation environment to use the DB2 Solutions:

## Customization Steps for DB2 Solutions

**Step 1** Customize EXECs.

Make these changes to the following EXECs:

EXEC	Change
DSU101C	Change the value specified for variable DB2 to the name of one of your DB2 subsystems.
DSU101C	Verify that the variables MSTR, DBM1, IRLM and DIST correctly identify the names of the Started Tasks for the DB2 you named.

**Note:** For the initial install, you can accept the default values for the rest of the variables in this EXEC.

**Step 2** Determine number of DB2 subsystems.

You can monitor up to nine different DB2 subsystems with the DB2 Solutions. However, changing the number of DB2 subsystems that your BBI subsystem (BBI-SS) monitors requires that you also change the settings of some DB2 variables (See Appendix F, “DB2 Solution Variables and Parameters”).

If your BBI-SS is monitoring more than one subsystem, follow these steps:

- 2.A** Set the value of `n` in `D_DB2NUMn` to the number of DB2 subsystems in `DSU001C`.
- 2.B** Create as many copies of `EXEC DSU101C` as the number of subsystems you plan to monitor and name the copies as follows:
- 2.C** `DSU101C, DSU102C...DSU10nC`
- 2.D** where `n` equals the value of `D_DB2NUM` in step one.
- 2.E** Set each copy's variable to identify a unique DB2 subsystem.
- 2.F** Verify that the following variables' values correctly identify the names of the Started Tasks for this DB2 subsystem:
  - MSTR
  - IRLM
  - DBM1
  - DIST

**Step 3** Update and activate Rules.

Rules are initially `DISABLED` with `MAINVIEW AutoOPERATOR Solutions`. You must enable the Rules that affect the solutions you plan to use. In addition, you must update certain parameters in `BBPARM` to reflect the solutions Rules you enabled.

To activate and update DB2 Solutions Rules, follow these steps:

- 3.A** Activate the Rules in each set of Rules for the DB2 Solutions by using a text editor to change `STATUS(DISABLED)` to `STATUS(ENABLED)`.
- 3.B** Enable the Rule for message `PM0010I` found in Rule set `AA0RUL00` of the `MAINVIEW AutoOPERATOR` product.

**3.C** Activate the following DB2 Solutions sets of Rules from the AAO RULE PROCESSOR status control panel:

- AAORULD1
- AAORULD2
- AAORULD3
- AAORULD5
- AAORULD6
- AAORULD8

**3.D** Update the RULESCAN and RULESET parameters in BBPARM member BBISSP00 to reflect the DB2 Solutions Rules and make these sets of Rules permanently active.

**Note:** A sample is provided in BBPARM member BBISSPD1.

**Step 4** Add monitors and EXECs.

**4.A** Add the monitors and time-initiated EXECs for the DB2 Solutions to your startup list, which is specified in BBPARM member BBIISP00 (as shown in the sample provided in BBPARM member BBISSPD1).

**4.B** Start the monitors and time-initiated EXECs from the AAO Time-Initiated EXEC panel by specifying:

```
BLK=DPE002B
```

and

```
BLK=DPE030B
```

**Step 5** Determine threshold levels.

When running DB2 applications, DB2 informational messages appear frequently on your console screen. For the DB2RNWY Solution (see “DB2 Runaway Query Control (DB2RNWY Solution)” on page 16-1), the frequency of a message's appearance is directly related to any threshold levels you set before using this solution:

**5.A** Browse hilevel.BBPARM(DMRBEX00) to review DB2RNWY threshold level parameters.

**5.B** Browse hilevel.BBPARM(DMRBEXD1) to review the BMC Software recommended settings for these parameters.

**5.C** Copy member DMRBEXD1 into member DMRBEX00 and set each threshold parameter in DMRBEXD1 to the value of your choice.

**Note:** If you have more than one DB2 subsystem, you must copy the thresholds for each DB2 TARGET statement.

**5.D** Put these changes into effect by issuing the command

```
.E P DMRBEX00
```

from any command line.

**Step 6** Modify your ALERT profile.

To display the DB2 ALERT Queue on the ALERT Overview application, you must modify your ALERT Profile.

Use the primary command `PROFILE` and set the name of the ALERT Queue to match what you used for the shared variable `D_ALERT_Q`.

---

# Chapter 14 DB2 Global Operations Solutions

The following DB2 Global Operations Solutions provide initialization routines as well as utilities and minimum alert management for DB2.

<b>DB2ALRT</b>	Issues major DB2 messages as MAINVIEW AutoOPERATOR ALERTs
<b>DB2INIT</b>	Provides information about the DB2 complex to other solutions through variables
<b>DMRALRT</b>	Issues major DMR messages as MAINVIEW AutoOPERATOR ALERTs
	<b>Note:</b> DMR refers to MAINVIEW for DB2.
<b>DMRUTIL</b>	Provides service routines that convert DMR services to local variables

DMR is a prerequisite for several DB2 Solutions as indicated in the individual solution description.

## DB2 Major Messages (DB2ALRT Solution)

During its operations, DB2 issues a number of messages with various severity levels; some of these messages require immediate attention.

The DB2ALRT Solution transforms these messages, listed under “Invocation” on page 14-7 into MAINVIEW AutoOPERATOR ALERTs, assigning a color to each message based on the severity level of the DB2 problem.

## Variables

This solution does not require setting any variables.

## Invocation

This solution is invoked by Rules for these major DB2 messages:

```
DSN3201I ABNORMAL EOT IN PROGRESS FOR USER= CON-ID= COR-
ID=
DSNB200I UPDATE VVDS FAILED
DSNB204I OPEN OF DATA SET FAILED. DSNAME = dsn
DSNB207I DYNAMIC ALLOCATION OF DATA SET FAILED. REASON =
rrr
        DSNAME = dsn
DSNB217I csect-name - ONLINE RECOVERY FOR AN INCONSISTENT
PAGE
        WAS UNSUCCESSFUL FOR DBNAME = dbn, SPACENAME =
spn,
        PAGE NUMBER = X'pno'
DSNB224I csect-name - BUFFER MANAGER I/O ERROR DURING
function
        DBNAME=dbn, ...
DSNB225I BUFFER MANAGER I/O ERROR
DSNB226I BUFFER MANAGER DETECTED INVALID PAGE
DSNB227I DFHSM RECALL FAILED
DSNB551I BSDS READ ERROR
DSNB552I BSDS WRITE ERROR
DSNB553I BSDS INSERT ERROR
DSNB601I BUFFER POOL nn FULL
DSNB602I UNABLE TO CREATE BUFFERPOOL
DSNB603I INSUFFICIENT STORAGE FOR BUFFERPOOL
EXPAND/CREATE
DSNB605I INSUFFICIENT VIRTUAL STORAGE FOR BUFFERPOOL
DSNB606I INSUFFICIENT STORAGE FOR HIPERPOOL
DSNB607I UNABLE TO CREATE HIPERPOOL - NO EXPANDED
DSNB608I UNABLE TO CREATE HIPERPOOL
DSNB609I VIRTUAL BUFFERPOOL IS ZERO - DEFAULT USED
DSNB610I UNABLE TO CREATE HIPERPOOL - NO ADMF
DSNB611I HIPERPOOL DELETED - ADMF INACTIVE
DSNC001I CICS UR INDOUBT RESOLUTION IS INCOMPLETE FOR
name
DSNC030E ERROR WRITING TO TRANSIENT DS
DSNC034I INDOUBT RESOLUTION FOR ur-id IS INCORRECT
DSNC035I INDOUBT RESOLUTION INCOMPLETE
DSNC036I INDOUBT RESOLUTION FOR ur-id IS INCONSISTENT
DSNC901I UNRECOVERABLE I/O ERROR IN DSNCCOM1
DSNI001I RESTART HAS BEEN DEFERRED
DSNI007I UNABLE TO DIRECT READ A LOG RECORD
```

```

DSNI010I BROKEN PAGE ACCESSED TYPE type NAME name MODNAME
        csect-name CONN-ID id CORR-ID id
DSNI012I PAGE LOGICALLY BROKEN TYPE type NAME name
MODNAME
        modname ERQUAL erqual
DSNJ004I ACTIVE LOG COPY n INACTIVE, LOG IN SINGLE MODE,
        ENDRBA=...
DSNJ008E nn OF mm ACTIVE LOGS ARE FULL sname NEEDS
ARCHIVE
        SCRATCH. REPLY YY WHEN DEVICE READY OR N TO
CANCEL
DSNJ013I TERMINAL ERROR ccc IN BUFFER rrr BEFORE ACTIVE
LOG WRITE
DSNJ014I TERMINAL ERROR ccc IN BUFFER rrr AFTER ACTIVE
LOG WRITE
DSNJ073I LOG ARCHIVE UNIT ALLOCATION FAILURE DETECTED,
        RETURN CODE = nnnn. ALLOCATION OR OFF-LOAD OF
ARCHIVE
        LOG DATA SET MAY FAIL
DSNJ100I csect-name ERROR OPENING BSDSn DSNAME=..., ERROR
STATUS=...
DSNJ102I LOG RBA CONTENT OF LOG DATA SET DSNAME=...
STARTRBA=..
        ENDRBA=.. DOES NOT AGREE WITH BSDS INFORMATION
DSNJ103I LOG ALLOCATION ERROR DSNAME=dsname, ERROR
STATUS=eeeeiiii
DSNJ104I csect-name RECEIVED ERROR STATUS nnn FROM macro-
name
        FOR DSNAME dsname
DSNJ105I csect-name LOG WRITE ERROR DSNAME=...,
LOGRBA=...,
        ERROR STATUS=ccccffss
DSNJ106I LOG READ ERROR DSNAME=..., LOGRBA=...,
        ERROR STATUS=ccccffss
DSNJ107I READ ERROR ON BSDS DSNAME=... ERROR STATUS=...
DSNJ108I WRITE ERROR ON BSDS DSNAME=... ERROR STATUS=...
DSNJ109I OUT OF SPACE IN BSDS DSNAME=...
DSNJ110E LAST COPYn ACTIVE LOG DATA SET IS nnn PERCENT
FULL
DSNJ111E OUT OF SPACE IN ACTIVE LOG DATA SETS
DSNJ115I OFFLOAD FAILED FOR ARCHIVE
DSNJ117I INITIALIZATION ERROR READING BSDS DSNAME=...,
        ERROR STATUS=...
DSNJ120I DUAL BSDS DATA SETS HAVE UNEQUAL TIME STAMPS,
BSDS1
        SYSTEM=..., UTILITY=..., BSDS2 SYSTEM=...,
UTILITY=...
DSNJ124I OFFLOAD OF ACTIVE LOG SUSPENDED FROM RBA xxxx TO
RBA
        yyyy DUE TO I/O ERROR
DSNJ126I BSDS ERROR FORCED SINGLE MODE
DSNJ150E LOG CAPTURE EXIT ABEND

```

```
DSNL007I DDF IS ABNORMALLY TERMINATING
DSNL033I DDF TERMINATION BECAUSE OF ABEND
DSNL400E INDOUBT THREAD HEURISTIC DAMAGE
DSNL401E INDOUBT THREAD REMOTE ABORT - HEURISTIC DAMAGE
DSNL402I INDOUBT THREAD REMOTE COMMIT - HEURISTIC DAMAGE
DSNL403I INDOUBT THREAD REMOTE ABORT - HEURISTIC DAMAGE
DSNL404E PROTOCOL ERROR
DSNL405I THREAD PLACED INDOUBT
DSNL406I THREAD MAY BE INDOUBT - COMM FAILURE
DSNL408I INDOUBT THREAD HEURISTIC DAMAGE - COORDINATOR
DSNL409I INDOUBT THREAD HEURISTIC DAMAGE - COORDINATOR
DSNL411E COLD START BY COORDINATOR - MANUAL RESOLUTION
DSNL412I PROTOCOL ERROR DURING SYNCHPOINT
DSNL413I PROTOCOL ERROR DURING SYNCHPOINT
DSNL414E PROTOCOL ERROR DURING INDOUBT
DSNL420I COLD START BY PARTICIPANT - POSSIBLE DAMAGE
DSNL421I SNA XLN PROTOCOL VIOLATION
DSNL500I CONVERSATION FAILED TO LOCATION locname ...
DSNL501I CNOS PROCESSING FAILED
DSNL502I SYSTEM CONVERSATION FAILED TO LOCATION=locname
...
DSNM002I IMS/V S xxxx DISCONNECTED FROM SUBSYSTEM yyyy
RC=rc
DSNM004I RESOLVE INDOUBT ENTRY(S) ARE OUTSTANDING FOR
SUBSYS xxxx
DSNM005I RESOLVE INDOUBT SYNCHRONIZATION PROBLEM WITH
SUBSYS xxxx
DSNP001I DSNPm m m m m - dsn IS WITHIN n KBYTES OF AVAILABLE
SPACE
RC=r CONNECTION-ID=id CORRELATION-ID=id
DSNP007I DSNPm m m m m m - EXTEND FAILED FOR dsn
RC=r CONNECTION-ID=id CORRELATION-ID=id
DSNP011I DSNPm m m m m m - MEDIA MANAGER SERVICES ERROR FOR dsn.
MMRC=C
DSMRC=r CONNECTION-ID=id CORRELATION-ID=id
DSNP012I DSNPm m m m m m - ERROR IN VSAM CATALOG LOCATE FUNCTION
FOR dsn
CTLGRC=r CTLGRSN=r CONNECTION-ID=id CORRELATION-
ID=id
DSNP015I IRLM MANUAL UNLOCK FAILED
DSNP028I HSM RECALL FAILED
DSNT377I PLAN plan-id1 WITH CORRELATION ID id1 CONNECTION
ID id2
IS IN CONFLICT WITH AN INDOUBT THREAD
DSNT500I csect-name RESOURCE UNAVAILABLE REASON r TYPE t
NAME n
DSNT501I csect-name RESOURCE UNAVAILABLE CORRELATION-ID
id1
CONNECTION-ID id2 REASON r TYPE t NAME n
DSNV086E DB2 ABNORMAL TERMINATION REASON=xxxxxxxx
```

```
DXR013E ABEND UNDER IRLM TCB/SRB
DXR016E BUCK PROCESS TIME EXCEEDS 30 SECONDS
DXR019E STORAGE POOL DESTROYED
DXR021E ABEND VTAM ERROR
DXR022E SUBTASK FAILURE
DXR023E INTERNAL OR MVS ERROR
DXR024E VTAM REQUEST UNSUCCESSFUL
DXR027A SESSION LOST
DXR031E ABEND UNDER PTB SRB
DXR050I OUT OF STORAGE DETECTED DURING DEADLOCK
```

## Processing Flow

When any message listed under “Invocation” on page 14-8 is received, an ALERT is issued. Color has been assigned to each ALERT based on the known severity of the attached message.

## EXECs

There is one EXEC for this solution:

**DPE992C**      Creates an ALERT for any message that invokes DPE992C

## Rules

There is one Rule for each of the messages listed under “Invocation” on page 14-8.

The Rules are distributed DISABLED in RULESET AAORULD8. You must ENABLE these Rules to implement this solution.

## DB2 Environment Set-Up (DB2INIT Solution)

Before you use the DB2 Solutions, you must provide some descriptive information about the DB2 subsystems you plan to use. This section describes how you provide this information to your automation environment.

**Note:** For additional information about customization steps affecting all solutions, see Chapter 13, “Initial Customization Steps.”

## Variables

To use DB2INIT, you must establish values for both the general DB2 variables and the DB2 subsystem variables as follows:

These are the general DB2 variables:

### **D\_ALERT\_TGT**

Target to receive ALERTS

### **D\_ALERT\_Q**

Name of the ALERT queue for all ALERTS generated by the solutions

### **D\_ALERT\_IDS**

TSO IDs to receive notification of ALERTS through TSO SEND commands

### **D\_ALERT\_SEND**

Indicates whether to issue the send command (Y/N)

### **D\_ALERT\_SENDOPT**

MVS SEND option; either LOGON or NOW

### **D\_DB2NUM**

Number of DB2s defined to this BBI-SS

Set these variables by modifying EXEC DSU001C. For a complete list of all the DB2 Solutions variables, see Appendix F, “DB2 Solution Variables and Parameters”.

For the DB2 subsystems variables, one set of the following variables should be defined for each DB2 system you intend to control through solutions:

<b>D_DB2n</b>	nth DB2 target name
<b>D_MSTRn</b>	nth DB2 MSTR address space name
<b>D_DBM1n</b>	nth DB2 DBM1 address space name
<b>D_IRLMn</b>	nth DB2 IRLM address space name
<b>D_DISTn</b>	nth DB2 DIST address space name

Set these variables by modifying EXEC DSU10 $n$ C, where  $n$  is the number of the associated DB2 subsystem. If you start more than one DB2 subsystem, you must duplicate the sample EXEC DSU101C provided. Other EXECs, up to DSU105C, are included with only the comments box.

**Note:** If you try to display the DB2 variables names and content by using the MAINVIEW AutoOPERATOR facilities, be aware that there might be several DB2 subsystems.

## Invocation

As seen in “Variables” on page 14-6, DB2INIT consists of EXEC DSU001C and one or several EXECs DSU10 $n$ C. You should ensure that these EXECs run as soon as possible after the host BBI-SS has been started.

This type of request usually is honored by having a Rule for PM0010I, which is the primary MAINVIEW AutoOPERATOR message. This Rule should initiate some kind of start-up EXEC. You need to add the necessary calls to EXECs found in “Variables” on page 14-6 to that start-up EXEC, such as

```
IMFEXEC SELECT EXEC(DSU001C) WAIT(Y)
```

## Processing Flow

Solution DB2INIT is executed only once—at BBI-SS start-up. If you need to run DB2INIT dynamically (such as the first time, or after a variable change), you can call the processing EXECs from any MAINVIEW AutoOPERATOR workstation.

## EXECs

The EXECs for the DB2INIT Solution are

<b>DSU001C</b>	Initializes shared variables for solutions
<b>DSU10<math>n</math>C <math>n</math>C</b>	Initializes shared variables for a particular DB2 subsystem where $n$ is the number of DB2 subsystems monitored by this BBI-SS

## Rules

There is no Rule for the DB2INIT Solution.

## DMR Major Messages (DMRALRT Solution)

MAINVIEW for DB2 (DMR) is one of the major components of MAINVIEW. It provides DB2-related performance information for the system programmer and the Database Administrator.

Several DMR components issue messages with various severity levels. DMR experts have sorted out these messages and selected a set of those requiring immediate attention. In fact, only the messages reported as *severe exceptions* have been retained, along with their associated *clearing* message.

The DB2 solutions were developed from a subset of those DMR messages—those considered to be candidates for automation.

The remaining messages are listed following “Invocation”. DMRALRT transforms the messages in this set into MAINVIEW AutoOPERATOR alerts, assigning a color to each message based on the severity level of the DMR problem.

**Note:** DMRALRT applies only when MAINVIEW for DB2 is installed.

### Variables

For this solution to function properly in your environment, you must establish values for the general DB2 variables and the DB2 subsystems variables defined in the DB2INIT solution.

### Invocation

This solution is invoked by Rules for these major DMR messages:

DZ1010S - INDOUBT THREAD

DZ1011I - EXCEPTION CLEARED: INDOUBT THREAD

DZ1040S - EDM POOL FULL FAILURES

DZ1041I - EXCEPTION CLEARED: EDM POOL FULL FAILURES

DZ1050S - BP(n) DM CRITICAL THRESHOLD REACHED

DZ1051I - EXCEPTION CLEARED: BP(n) DM CRITICAL THRESHOLD REACHED

DZ1060S - BP(n) IMMEDIATE WRITE THRESHOLD REACHED

DZ1061I - EXCEPTION CLEARED: BP(n) IMMEDIATE WRITE THRESHOLD REACHED

DZ1070S - BP(n) EXPANSION FAILURE, MAXPAGES REACHED

DZ1071I - EXCEPTION CLEARED: BP(n) EXPANSION FAILURE, MAXPAGES REACHED

DZ1080S - BP(n) EXPANSION FAILURE, VIRTUAL STORAGE SHORTAGE

DZ1081I - EXCEPTION CLEARED: BP(n) EXPANSION FAILURE, VIRT. ST. SHORTAGE

DZ1090S - FINAL ACTIVE LOG DATASET 75% FULL

DZ1091I - EXCEPTION CLEARED: FINAL ACTIVE LOG DATASET 75% FULL

DZ1100S - ACTIVE LOG REDUCED TO SINGLE MODE

DZ1101I - EXCEPTION CLEARED: ACTIVE LOG REDUCED TO SINGLE MODE

DZ1110S - BSDS REDUCED TO SINGLE MODE

DZ1111I - EXCEPTION CLEARED: BSDS REDUCED TO SINGLE MODE

DZ1120S - SOS CRITICAL

DZ1121I - EXCEPTION CLEARED: SOS CRITICAL

## Processing Flow

When any problem message is received, an ALERT is issued. Color has been assigned to each ALERT based on the known severity of the attached message. When any clearing message is received, the ALERT is cleared.

## EXECs

There is one EXEC for this solution:

**DPE994C** Issues or deletes the ALERT associated with any of the DMR messages found in "Invocation"

## Rules

There is one Rule for each of the messages given in “Invocation” on page 14-7. The Rule-IDs equal the message-IDs.

The Rules are distributed DISABLED in RULESET AAORULD3. You must ENABLE these Rules to implement this solution.

## DMR Utilities (DMRUTIL Solution)

MAINVIEW for DB2 (DMR) is one of the major components of MAINVIEW. It provides DB2-related performance information for the system programmer and the Database Administrator (DBA).

With the MAINVIEW architecture, MAINVIEW AutoOPERATOR and DMR can communicate internally. This provides great power by letting the solutions investigate what is happening within DB2. This communication is based on an internal exchange of the regular DMR screen image in reply to a received command.

To shorten development time, service routines are provided for the major DMR services, which let you get direct DB2 data from local variables. These routines are included in the DMRUTIL Solution.

**Note:** DMRUTIL applies only when MAINVIEW for DB2 is installed.

## Variables

This solution does not require any value to be set for variables.

## Invocation

The EXECs from this solution are called by the various DB2 Solutions. Therefore, any DMR-based solution requires DMRUTIL.

## EXECs

Following are the EXECs for the DB2UTIL Solution. Each utility EXEC is completed by an example on how to call it.

<b>DPE002C</b>	Support for BFRPL format in DB2 3.1
<b>DPE014C</b>	Start trace TYPE=D
<b>DPE900C</b>	DB2ST formatter — breaks DB2ST display into variables
<b>DPE900D</b>	Driver example for DPE900C and DPE901C
<b>DPE901C</b>	USERS formatter — breaks USERS display into variables
<b>DPE901D</b>	Driver example for DPE901C
<b>DPE902C</b>	Breaks DB2EX display into variables
<b>DPE902D</b>	Driver example for DPE902C
<b>DPE903C</b>	Breaks LOCKD display into variables
<b>DPE903D</b>	Driver example for DPE903C
<b>DPE905C</b>	BFRPL formatter — breaks BFRPL display into variables
<b>DPE905D</b>	Driver example for DPE905C
<b>DPE908C</b>	DBTS formatter — breaks DBTS display into variables
<b>DPE908D</b>	Driver example for DPE908C
<b>DPE908DR</b>	Driver for DPE908C (written in REXX)
<b>DPE909C</b>	IMFC call (IMAGE=NO)
<b>DPE917C</b>	Breaks BFRPL display into variables (DB2 release 2.3) and makes BFRPL variables compatible for DPE002C
<b>DPE919C</b>	Fixed column positions for DBTS display
<b>DBE920C</b>	DB2ST formatter
<b>DPE930C</b>	DB2ST formatter — breaks DB2ST display into variables (DB2 release 2.3)
<b>DPE931C</b>	USERS formatter — breaks USERS display into variables (DB2 release 3.1)

- |                |   |
|----------------|---|
| <b>DPE932C</b> | DLOGS formatter — breaks DLOGS display into variables (DB2 release 3.1) |
| <b>DPE933C</b> | DBTS formatter — breaks DBTS display into variables (DB2 release 3.1)   |
| <b>DPE935C</b> | BFRPL formatter — breaks BFRPL display into variables (DB2 release 3.1) |

## Rules

There are no Rules attached to this solution.

---

---

# Chapter 15 DB2 Resource Contention Analysis Solutions

The following DB2 Resource Contention Analysis Solutions concentrate on a few major resources that are critical to DB2 operations:

**DB2DLOK** Analyzes deadlock or timeout situations

**DB2TFUL** Provides an early warning of table spaces filling up

**DB2THRD** Resolves IMS and CICS queued for thread situations

## Deadlock/Timeout Analysis (DB2DLOK Solution)

A *timeout* occurs when the length of time a user has been waiting for a lock exceeds the IRLM timeout limit.

A *deadlock* occurs when two or more users have requested locks for two or more resources and each user has a lock that the other needs.

If these events occur frequently, consider redesigning the data structure or application.

For DB2DLOK, the PLAN that was waiting for a lock on a resource is called the *lock requestor*. The PLAN that held the lock is called the *lock owner*. The lock owner continues to run normally; the lock requestor receives a negative SQL return code.

**Note:** DB2DLOK requires MAINVIEW for DB2.

## Variables

To use DB2DLOK, you must establish values for the general DB2 variables and the DB2 subsystem variables defined in the DB2INIT Solution.

You should also define values for the following thresholds:

- MAX\_AGE** The age in minutes of the oldest event to be used for threshold comparison. For example, if MAX\_AGE = 30, then an event that occurred 31 minutes ago is not used for threshold checking. This limits the problem analysis to recent history. To include all events that are saved in the history table, set MAX\_AGE = 0. The default is MAX\_AGE = 30.
- MAX\_OWNER** Triggers an ALERT if a PLAN is a lock owner more than MAX\_OWNER times in MAX\_AGE minutes. The default is 2, so a PLAN has to be a lock owner 3 or more times to trigger the ALERT.
- MAX\_REQ** Triggers an ALERT if a PLAN is a lock requestor more than MAX\_REQ times in MAX\_AGE minutes. The default is 2, so a PLAN has to be a lock requestor 3 or more times to trigger the ALERT.
- MAX\_EITHER** Triggers an ALERT if a PLAN is involved in locking conflicts either as a lock owner or lock requestor more than MAX\_EITHER times in MAX\_AGE minutes. This threshold is checked only if the PLAN did not exceed the MAX\_OWNER or MAX\_REQ thresholds. The default is 3. For example, if a PLAN is a lockowner 2 times and a lock requestor 2 times, an ALERT is issued.

These variables should be set by reviewing EXECs DSU10nC, attached to initializing values pertaining to each specific DB2. These EXECs are included in the DB2INIT Solution.

## Invocation

This solution is invoked by Rules for the following messages:

**DSNT375I** PLAN **plan-id1** WITH CORRELATION-ID **id1** CONNECTION-ID **id2** LUW-ID **id3** IS DEADLOCKED WITH PLAN **plan-id2** WITH CORRELATION-ID **id4** CONNECTION-ID **id5** LUW-ID **id6**

**DSNT376I** PLAN **plan-id1** WITH CORRELATION-ID **id1** CONNECTION-ID **id2** LUW-ID **id3** IS TIMED OUT DUE TO A LOCK HELD BY PLAN **plan-id2** WITH CORRELATION-ID **id4** CONNECTION-ID **id5** LUW-ID **id6**

**DSNT501I** Csect-name RESOURCE UNAVAILABLE CORRELATION-ID **id1** CONNECTION-ID **id2** LUW-ID **id3** REASON **reason** TYPE **type** NAME **name**

## Processing Flow

The first part of this solution maintains a history table of deadlock and timeout events for each DB2 user. Each time a new event is added to the table, an analysis is done of the history table to see whether any PLAN has exceeded any threshold for contention.

### History Table

By running EXEC DPE019C, you can display the deadlock history table at any time. This wrap-around table shows all previously recorded deadlock and timeout events. Once the table is full, a new event takes the place of the oldest event in the table. DPE019C writes two reports to the journal.

**Report 1:** The first DPE019C report is a display of all the events saved in the deadlock table. It starts with the most current event and ends with the oldest event:

```

..DPE019C DEADLOCK/TIMEOUT HISTORY FOR DB2 FOLLOWS

DATE      TIME      TYPE      LOCK   PLAN      CONN-ID   CORR-ID
-----
95.031  11:11:11  (1)      OWNR   (2)       (3)       (4)
                        REQR   (5)       (6)       (7)
                        NAME   (8)

```

Numbered data items are:

- (1) Event type - either TIMEOUT or DEADLOCK
- (2) Plan name of lock owner
- (3) Connection-ID of lock owner
- (4) Correlation-ID of lock owner
- (5) Plan name of lock requestor
- (6) Connection-ID of lock requestor
- (7) Correlation-ID of lock requestor
- (8) Resource name in contention (from DSNT501I)

**Report 2:** The second report from DPE019C summarizes the events by PLAN and counts events above and below the MAX\_AGE threshold:

```
..DPE019C DEADLOCK/TIMEOUT SUMMARY FOR xxxx
..DPE019C MAX_AGE = nnn MINUTES
```

DB2	PLAN	LOCKOWNR < MAX_AGE	LOCKOWNR TOTAL	LOCKREQR < MAX_AGE	LOCKREQR TOTAL
1	2	3	4	5	6

- (1) target DB2 (whose subsys ID is xxxx)
- (2) DB2 PLAN
- (3) number of times the PLAN was a lock owner within the last &MAXAGE minutes
- (4) total number of times the PLAN was a lock owner
- (5) number of times the PLAN was a lock requestor within the last &MAXAGE minutes
- (6) total number of times the PLAN was a lock requestor

You can use this report to adjust your thresholds.

## ALERTs

If a threshold is exceeded, DB2DLOK creates an ALERT that details all the conflicting events for the plan in question and writes a message in the journal. You also can choose to notify a select group of TSO users.

From the ALERT, you can request a report of all deadlock events, request that a detail trace be started, or start a LOCKD data extractor EXEC. This is documented in the ALERT help panel.

Entering the EXT command in the ALERT RSP field invokes EXEC DPE015C. DPE015C examines the LOCKD display periodically and looks for lock contentions where this PLAN is a lock owner. When complete, a report of all contentions is written to the journal, followed by a summary by database and table space.

- The variable STOPTIME determines how long the EXEC runs (default of 10 minutes, set in DSU10nC)
- The variable INTVL determines how long the EXEC waits between LOCKD displays (default of 5 seconds, set in DSU10nC)

## Data Extractor

The LOCKD data extractor (EXEC DPE015C) writes two reports to the journal.

**Report 1:** The first report is a history report of all the events found:

```
DPE015C &DB2 LOCK EXTRACTION HISTORY FOR PLAN &PLAN
FOLLOWS.
DPE015C &DB2 &MAX EVENTS FOUND IN THE LAST &STOPTIME
MINUTES.
```

TIME	DB	TS	USERID	CNT	WAITER	TYPE
1	2	3	4	5	6	7

(1) HH:MM:SS that the locking conflict was observed

(2) Database in conflict

(3) Table space in conflict

(4) User ID owning the lock

(5) Count of lock waiters

(6) User ID of the first waiter

(7) Waiter type

For a further explanation of the above fields, consult the *MAINVIEW for DB2 User Guide* for the LOCKD service.

**Report 2:** The second report from DPE015C is a summary of the above events by DB/TS.

```
DPE015C &DB2: LOCK EXTRACTION REPORT BY DB/TS FOR PLAN
&PLAN
DPE015C OVER THE LAST &STOPTIME MINUTES FOLLOWS:
```

DATABASE	TBLSPACE	# EVENTS
-----	-----	-----
1	2	3

(1) Database held in conflict by PLAN &plan

(2) Table space held in conflict by PLAN &plan

(3) Number of times this DB/TS was held in conflict by PLAN &plan

## EXECs

The EXECs for the DB2 Deadlock/Timeout Solution are

<b>DPE010C</b>	Determines DB2 target and plans involved in deadlocks or timeouts
<b>DPE011C</b>	Performs deadlock history update
<b>DPE012C</b>	Updates resource name in history table, then checks against thresholds; issues ALERTS/MSGS and optional MVS SEND commands
<b>DPE013C</b>	Follow-up EXEC for ALERT
<b>DPE014C</b>	Starts trace
<b>DPE015C</b>	Analyzes LOCKD display
<b>DPE019C</b>	Reports deadlock history table to journal

## Rules

The Rules for the DB2 Deadlock/Timeout Analysis Solution are distributed in member AAORULD1:

**DSNT375I** Text-ID=DSNT375I, distributed DISABLED

**DSNT376I** Text-ID=DSNT376I, distributed DISABLED

**DSNT501X** Text-ID=DSNT501I, distributed DISABLED, applied to message DSNT501I only when the CSECT name is DSNILMCL.

The Rules are distributed DISABLED in RULESET AAORULD1. You must ENABLE these Rules to implement this solution.

## DB2 Table Space Filling Up (DB2TFUL Solution)

Many users access DB2 databases during the day, and some users might continue to add data until they run out of space.

When DB2 reports that all the space has been used up, it is too late to avoid an outage. DB2TFUL provides an early warning of a table space or a partition filling up.

DB2TFUL analyzes open table spaces and imposes user-specified thresholds on them. If a threshold is exceeded, an ALERT is issued.

Looking at many DB2 databases takes time and resources, so the attached analysis is run only every half-hour, through a timer request you set in MAINVIEW AutoOPERATOR.

**Note:** DB2TFUL requires MAINVIEW for DB2.

## Variables

To use DB2TFUL, you must establish values for the general DB2 variables and the DB2 subsystem variables defined in the DB2INIT Solution.

Additionally, you can define values for the following thresholds:

<b>MAX_EXTS</b>	The upper bound for the number of extents for a tablespace unless overridden by an exception. If zero, this threshold is ignored.
<b>MAX_UTIL</b>	The upper bound for the percent utilization for a tablespace unless overridden by an exception. If zero, this threshold is ignored.
<b>NUM</b>	Number of exceptions for this DB2. Must be equal to the highest suffix for variable NAME.
<b>NAME<math>n</math></b>	The name of the $n$ th exception. Must be specified as a tablespace. Can be generic.
<b>MAX_EXTS<math>n</math></b>	The extent threshold for the $n$ th exception name. If zero, this threshold is ignored.
<b>MAX_UTIL<math>n</math></b>	The utilization threshold for the $n$ th exception name. If zero, this threshold is ignored.

These variables should be set by reviewing EXEC DSU10 $n$ C, attached to initializing values pertaining to each specific DB2 subsystem. These EXECs are included in the DB2INIT Solution.

For example, suppose the following variables are set in EXEC DSU101C:

```
SET MAX_EXTS = 0
SET MAX_UTIL = 0

SET NUM      = 2

SET NAME1    = &STR(DSN*)
SET EXTS1    = 15
SET UTIL1    = 0

SET NAME2    = &STR(DSNDBO6_SYS*)
SET EXTS2    = 5
SET UTIL2    = 90
```

Setting both MAX\_EXTS and MAX\_UTIL to zero tells the solution to check only the exception thresholds.

The first exception states that any tablespace in a database starting with DSN that has more than 15 extents creates an ALERT.

The second exception states that an ALERT is generated for any tablespace starting with SYS in database DSNDB06 that has more than 5 extents or is over 90% utilized on the fifth extent.

## Invocation

Usually, DB2TFUL is initiated through a time-initiated request. A sample timer is provided in BBPARM member DPE030B. You also should create a call to this timer in your BBIISPx member. A sample entry is provided in BBPARM member BBIISPD1.

## Processing Flow

Using the timer facility described previously in “Invocation”, this solution is intended to run in the background and sample data periodically. However, both DPE030C and DPE031C can be executed from any COMMAND line of any MAINVIEW screen display by using the command character % (percent sign).

## EXECs

The EXECs for the DB2TFUL Solution are

- |                |   |
|----------------|---|
| <b>DPE030C</b> | Threshold analysis of the DBTS display for a particular DB2 |
| <b>DPE031C</b> | The EXEC that calls EXEC DPE030C for each defined DB2       |

## Rules

There are no Rules for this solution.

## DB2 Thread Control (DB2THRD Solution)

DB2 offers its users a number of access paths, referred to as *threads*. DSNZPARM specifies the maximum number of threads DB2 can have. When this limit is reached, DB2 queues each request until a thread is available. Once this queuing process is initiated, it causes delays to users.

For transaction-oriented systems such as IMS and CICS, waiting for a DB2 thread can degrade the throughput of the entire system and should be avoided.

In a data center using DB2, users often encounter a shortage of threads. In response to this situation, the DB2THRD Solution attempts to identify a TSO thread that can be made available for IMS or CICS to use.

**Note:** DB2THRD requires MAINVIEW for DB2.

### Variables

To use DB2THRD, you must establish values for the general DB2 variables and the DB2 subsystem variables defined in the DB2INIT Solution.

Additionally, you should define the following variable:

**Q\_VERIFY**      Indicates (Y/N) whether to issue an ALERT to verify the TSO cancel.

### Invocation

This solution is invoked by Rules for these major DMR messages:

```
DZ1020S - IMS TASK(S) QUEUED FOR THREAD
DZ1021I - EXCEPTION CLEARED: IMS TASK(S) QUEUED FOR
THREAD
DZ1030S - CICS TASK(S) QUEUED FOR THREAD
DZ1031I - EXCEPTION CLEARED: CICS TASK(S) QUEUED FOR
THREAD
```

## Processing Flow

If `Q_VERIFY = Y`, then an ALERT is created when the message DZ1020W or DZ1030W appears. This ALERT has an extended help panel (DPE040A) associated with it that can be modified by your site to give the operator-specific instructions for this situation. When the queued for thread condition clears, the solution intercepts messages DZ1021I and DZ1031I and deletes the ALERT.

If `Q_VERIFY = N`, then EXEC DPE041C is invoked to identify the TSO user with the shortest elapsed time. EXEC DPE040C then cancels that TSO user so that a thread can be freed up for CICS or TSO to use. A message is written to the journal identifying the canceled TSO user.

## EXECs

The EXECs for the DB2THRD Solution are

<b>DPE040C</b>	Analyzes message it has been scheduled from
<b>DPE041C</b>	Analyzes DMR USERS service output to locate the TSO user to be canceled

## Rules

There is one Rule for each of the messages listed under “Invocation” on page 15-9. The Rule-IDs equal the message-IDs.

The Rules are distributed DISABLED in RULESET AAORULD6. You must ENABLE these Rules to implement this solution.



---

# Chapter 16 DB2 Performance Management Solutions

The following DB2 Performance Management Solutions help data center personnel achieve maximum availability and increased response time:

**DB2RNWY** Reacts to DMR messages for CICS, IMS, TSO, Batch, CAF, and Utilities runaway queries

**Note:** DMR refers to MAINVIEW for DB2

**DB2RESP** Uses DMR services to identify potential causes of poor DB2 response time

DMR is a prerequisite for several DB2 Solutions as indicated in the individual solution description.

## DB2 Runaway Query Control (DB2RNWY Solution)

Runaway queries are one of the major problems affecting DB2 performance. MAINVIEW for DB2 dynamic and static SQL calls and warning capabilities made it possible to develop the DB2RNWY Solution, which provides selective cancel capabilities over such runaway queries.

This solution requires

- MAINVIEW for DB2
- MAINVIEW AutoOPERATOR for IMS and IMS RESOURCE ANALYZER (to cancel a runaway IMS query)

- MAINVIEW AutoOPERATOR for CICS (to cancel a runaway CICS query)

**Note:** DB2 provides limited protection from runaway queries with the Resource Limit Facility (RLF), available with DB2 Version 2.1. However, this applies only to dynamic SQL. Moreover, the result from RLF is a query that has stopped with an SQL error code (-905 expected).

## Variables

To use DB2RNWY, you must establish values for the global DB2 variables defined in the DB2INIT Solution.

## Invocation

This solution is invoked by Rules for the following messages:

```
DZ0610W RUNAWAY IMS <threshold> PST= TRAN= USER= J= TYPE=
CRGN=
DZ0611I EXCEPTION CLEARED: RUNAWAY IMS ...
DZ0620W RUNAWAY CICS TRANSACTION USER=u <threshold> J=
TRAN= TASK=
DZ0621I EXCEPTION CLEARED: RUNAWAY CICS TRANSACTION ...
DZ0630W RUNAWAY TSO QUERY USER= <threshold exceeded>
DZ0631I EXCEPTION CLEARED: RUNAWAY TSO QUERY ...
DZ0640W RUNAWAY BATCH USER= <threshold exceeded> J=
DZ0641I EXCEPTION CLEARED: RUNAWAY BATCH ...
DZ0650W RUNAWAY CAF, USER= <threshold exceeded> J=
DZ0651I EXCEPTION CLEARED: RUNAWAY CAF ...
DZ0660W RUNAWAY UTILITY FUNCTION, USER= <threshold
exceeded> J=
DZ0661I EXCEPTION CLEARED: RUNAWAY UTILITY ...
```

## Processing Flow

When MAINVIEW for DB2 detects a runaway query, as determined by thresholds set in BBPARM member DMRBEX00, a message is sent to the LOG and filtered to be echoed as an ALERT on the MAINVIEW centralized screen. A sample member named DMRBEXD1 is provided in the BBPARM data set.

Operators must confirm the cancel of a runaway query job by entering `CAN` in the response field for the `ALERT`; the `ALERT` is cleared when the operator cancels the runaway query job, or when the runaway condition ends.

If the runaway query is

- a job or TSO user, the cancel is done through the appropriate MVS cancel command
- an IMS region, the solution verifies that the region name, number, and trancode match those in the warning message and that the region status is `ACTV-USR` or `ACTV-DB2`

It then issues a `/STOP REG ABDUMP` command followed by a `/STOP REGION CANCEL` command.

- a CICS transaction, the solution issues a `KILL TASK xxxx FORCE`

**Note:** This solution does not cancel a runaway query automatically. It always requests operator agreement by the `ALERT`.

## EXECs

The EXECs for the DB2 Runaway Query Control Solution are

<b>DPE020C</b>	Create <code>ALERT</code> s for runaway query
<b>DPE021C</b>	Delete <code>ALERT</code> s for runaway query
<b>DPE022C</b>	Schedules MVS cancel EXEC <code>DPE024C</code>
<b>DPE023C</b>	Schedules IMS cancel EXEC <code>DPE025C</code>
<b>DPE024C</b>	Performs MVS cancel
<b>DPE025C</b>	Performs IMS cancel
<b>DPE026C</b>	Schedules CICS cancel EXEC <code>DPE027C</code>
<b>DPE027C</b>	Performs CICS cancel

## Rules

The Rules are distributed `DISABLED` in `RULESET AAORULD2`. You must `ENABLE` these Rules to implement this solution.

## DB2 Response Time (DB2RESP Solution)

Because transaction response time for all CICS, IMS, and TSO users relies on DB2's ability to reply to requests in the shortest possible time, it is important to obtain data about DB2 response time as early as possible.

This solution examines several MAINVIEW for DB2 (DMR) services for potential performance problems by comparing the observed values to user-defined thresholds and by reporting any observations that are above threshold levels.

DB2RESP is initiated by certain warning messages for elapsed and CPU time monitors or invoked directly by the operator.

The usefulness of this solution depends in large part on the thresholds chosen, so these must be chosen carefully. BMC Software has supplied what seem to be reasonable defaults with this solution, but you may want to make some modifications to fit your installation.

In the process of determining your installation thresholds, you should become aware of the DMR data items that are good performance indicators for your system.

This solution requires the DB2INIT and DMRUTIL Solutions.

### Variables

To use DB2RESP, you must establish values for the general DB2 variables and the DB2 subsystem variables defined in the DB2INIT Solution.

This solution makes extensive use of thresholds for many domains of DB2 performance. For each DB2 subsystem, up to 35 important figures are analyzed. You can change any of the default thresholds. They are available for each DB2 system, and are located in the DSU10nC EXECs.

For more information on setting these threshold parameters, see Appendix F, "DB2 Solution Variables and Parameters".

## Invocation

This solution is invoked by Rules for these DMR messages:

```
DW0120W (nn) hh:mm: AVG ELAPSED TIME(parm) = nnn (>ppp)
*****
DW0150W (nn) hh:mm: AVG ELAPSED IN DB2(parm) = nnn (>ppp)
*****
DW0180W (nn) hh:mm: AVERAGE CPU USED(parm) = nnn (>ppp)
*****
DW0190W (nn) hh:mm: AVERAGE CPU IN DB2(parm) = nnn (>ppp)
*****
```

where:

**parm**  
{ ALL|IMS|CICS|TSO }

To get these messages out of MAINVIEW for DB2, you must activate some monitors as listed in member DPE002B of the BBPARM data set. This is done in BBPARM member BBIISPxx using the TARGET statement, as shown in BBIISPD1 of BBPARM. For example, to set the DB2 monitors, specify

```
TARGET=DB2D, BLK=DPE002B, USRID=xxxxxxxx
```

You should adjust the WVAL values to values appropriate to your installation. Information about WVAL values can be found in the “Set Timer Request” chapter of the *MAINVIEW for DB2 User Guide*.

DB2RESP processes warning messages with an identifier of IMS, CICS, TSO, or ALL. These identifiers are required and cannot be modified.

## Processing Flow

Occurrence of the DMR messages listed under “Invocation” on page 16-5 invokes DB2RESP, or you can invoke DB2RESP from any MAINVIEW command line by entering

```
%DPE002C db2name
```

Here, db2name is the name of the target DB2.

Online Help is available by entering

```
%DPE002C ?
```

The positional parameters for DPE002C are

- |              |   |
|--------------|---|
| <b>PARM1</b> | DB2 TARGET (REQUIRED)                             |
| <b>PARM2</b> | OUTPUT DESTINATION (OPTIONAL)DEFAULT = DB2 TARGET |
| <b>PARM3</b> | CONNECTION TYPE (OPTIONAL)                        |

Values can be: ALL(default), IMS, CICS, or TSO.

## EXECs

The EXECs for the DB2RESP Solution are

- |                |   |
|----------------|---|
| <b>DPE002C</b> | Performs checking of thresholds for DB2                   |
| <b>DPE005C</b> | Drives DPE002C from DB2 monitor messages                  |
| <b>DPE007C</b> | Governor for solution-driven from monitor warning message |

## Rules

There is one Rule for each of the messages listed under “Invocation” on page 16-5. The Rule-IDs equal the message-IDs.

The Rules are distributed DISABLED in RULESET AAORULD5. You must ENABLE these Rules to implement this solution.

---

# Chapter 17 E-Mail Solution

MAINVIEW AutoOPERATOR provides a pair of sample EXECs that can be used to send an e-mail from a REXX EXEC.

## EXECs

The EXECs for the e-mail Solution are

**QAOSMTP1** Resides in BBSAMP.

**QAOSMTP2** Resides in BBPROC.

**Note:** Browse the MAINVIEW AutoOPERATOR product libraries to view the most current version of these sample EXECs. BMC Software might ship PTFs at any time to enhance them.

This SMTP sample uses REXX sockets, which are part of the IBM TCP/IP product.

## QAOSMTP1

QAOSMTP1 is a sample EXEC that demonstrates how you can pass e-mail data to QAOSMTP2, which performs the actual e-mail transmission. Most customers will be able to use QAOSMTP2 without modification.

To send an e-mail, make a copy of QAOSMTP1 and store it in UBBPROC using a new meaningful name. Replace the data in the user-input fields with appropriate data for your installation.

You can also incorporate the logic in QAOSMTP1 into your own REXX application to generate e-mails.

---

The areas that need to be modified in QAOSMTP1 are listed in the following table.

Name	Description
SMTP_SERVER	Local mail server. This server distributes mail to the POP3 and Exchange mail servers.
SMTP_REPLYTO	The e-mail address that receives replies when responding to this e-mail. It must be a valid address (or example, YourName@your-company.com).
Sender of the e-mail	The first line queued (your name), which might be different from the REPLYTO e-mail ID. Valid formats are dependant upon the mail server being used. Most mail servers accepts '<user@your-company.com>'. Other typical formats that are accepted are "your name" and "your name"<id@bid.company.com>'. Check your SMTP documentation for acceptable values.
Recipient(s)	A blank delimited list. It is queued second. Do not use the form "name"<user@big.company.com>.

**Note:** A sender's name, subject, and at least one line of message text are required. Keep line lengths reasonable so that the e-mail has a readable format. The utility EXEC (QAOSMTP2) inserts a CrLf after each text line in the generated e-mail.

A QAOSMTP1 sample EXEC follows:

```

/* rexx qaosmtp1 */                                00010002
/* DOC GROUP(AO)  FUNC(SAMPLE)  AUTHOR(BMC)        */00020002
/* DOC DESC(sample EXEC for sending email via a SMTP server) */00030002
                                                    00040002
                                                    00050001
/*-----*/                                       00060001
* Sample EXEC which demonstrates sending an e-mail via a SMTP mail * 00070005
*   server.                                                    * 00080005
*   *                                                         * 00090005
* This EXEC constructs the e-mail data then calls QAOSMTP2 which * 00100005
*   reformats the data and sends it using the SMTP protocol. * 00110005
*   *                                                         * 00120005
* QAOSMTP2 talks to the SMTP server at port 25 using IBM REXX * 00130005
*   Sockets. NOTE: QAOSMTP2 resides in BBPROC.                * 00131005
*   *                                                         * 00140005
* Change Log                                                  * 00150005
*   9-mar-01 bpo4962 Use queue command instead of push        * 00159005
*   9-feb-01 bpo4874 shipped to make bpo4874 prereq bpo4851   * 00159105
*   30-jan-01 bpo4851 update comments to note QAOSMTP2 is in BBPROC * 00159205
*   27-aug-99 bpo4319 initial release                          * 00160005
*   *                                                         * 00170005
/*-----*/                                       00180001
                                                    00190001
trace n                                              00200001
"IMFEXEC MSG 'QAOSMTP1 EID" IMFEID "Started'"          00210001
                                                    00211005
address mvs newstack                                /* make a new stack */ 00212005
                                                    00220001
/*-----*/                                       00230001
*   *                                                         * 00231005
* ----- Beginning of lines requiring changes ----- * 00232005
*   *                                                         * 00233005
* The external data queue is used to communicate the message text, * 00240005
*   subject and sender's name to the utility EXEC (qaosmtp2). * 00250005
*   *                                                         * 00260005
* smtp_server is your local SMTP server. This server will distribute * 00261005
*   mail to the POP3 and exchange mail servers. Many mail servers * 00262005
*   will have "mail" as the first part of their network name but * 00262105
*   is not required to follow this standard.                  * 00262205
*   *                                                         * 00262305
* smtp_replyto is the Email address that will receive replies when * 00262405
*   responding to this Email. You should have a valid address here. * 00263005
*   *                                                         * 00264005
* Sender of the Email is the first line queued. It will be your * 00270005
*   name which may be different from the ReplyTo e-mail id.    * 00280005
* The formats allowed are dependant upon the mail server in use. * 00290005
* '<user@your-company.com>' is accepted by most mail servers. Some * 00291005
* other typical formats that are accepted 'your name' and * 00291105
* 'your name'<id@bid.company.com>'. Check your SMTP documentation * 00291205
* for acceptable values.                                       * 00292005
*   *                                                         * 00293005
* Recipient(s) is a blank delimited list. It is queued second. * 00300005
* Do not use the form "name"<user@big.company.com>.          * 00320005
*   *                                                         * 00321005
* Subject line is the third queued line.                      * 00330005
*   *                                                         * 00370005
* Email message text will be the remainder of lines queued. * 00371005
*   *                                                         * 00372005
* There must be a sender's name, a subject and at least one line of * 00380005
*   message text.                                             * 00390005
*   *                                                         * 00400005
* Keep line lengths reasonable so that your e-mails will look nice. * 00410005
*   *                                                         * 00420005
* The utility EXEC will insert a CrLf after each text line in the * 00430005
*   generated e-mail.                                         * 00440005
*   *                                                         * 00443005
/*-----*/                                       00450005
                                                    00460005
smtp_server = 'mail.your-company.com' /* your SMTP mail server */ 00470005
smtp_replyto = 'YourName@your-company.com' /* Reply-to Email id */ 00470105
                                                    00471004
queue '<'strip(imforgn)_'_AutoOPERATOR@your-company.com>' /* sender */ 00472005
                                                    00475004
queue 'user1@xxx.com user2@yyy.com group1@zzz.com' /* recipients */ 00478005
                                                    00480001
queue 'Email subject' /* Emails subject line */ 00520005
                                                    00520104

```

```

queue 'Email text line 1'                /* Email body          */ 00520205
queue 'Email text line 2'                00520305
queue 'Email text line 3'                00521005
                                         00660105
/*-----*
* ----- End of required changes -----* 00661005
*                                         * 00662005
*                                         * 00663005
*-----*/ 00664005
                                         00670001
smtp_debug = 'n'                          /* optional debug parm */ 00680001
                                         00690001
smtp_error = 0                            /* preset our error flag */ 00700001
                                         00710001
if words(smtp_server) /= 1 then           00720001
do                                         00730001
"IMFEXEC MSG '.QAOSMTP1 EID' IMFEID "Error, blanks not allowed in",
  "smtp_server="smtp_server""           00740001
smtp_error = 1                           00750001
                                         00760001
end                                         00770001
                                         00780001
if words(smtp_replyto) /= 1 then         00790001
do                                         00800001
"IMFEXEC MSG '.QAOSMTP1 EID' IMFEID "Error, blanks not allowed in",
  "smtp_replyto="smtp_replyto""       00810001
smtp_error = 1                           00820001
                                         00830001
end                                         00840001
                                         00850001
if smtp_error = 1 then                   00860001
do                                         00870001
"IMFEXEC MSG '.QAOSMTP1 EID' IMFEID "Ended with error(s)'"
"IMFEXEC EXIT CODE(8)"                 00880001
exit                                     00890001
end                                         00900001
                                         00910001
                                         00920001
                                         00930001
/* call the utility EXEC to send the e-mail */ 00940001
call qaosmtp2 smtp_server smtp_replyto smtp_debug 00950001
                                         00960001
exec_rc = result                          /* save qaosmtp2 rc    */ 00970001
address mvs delstack                     /* delete our stack    */ 00980001
                                         00990001
if exec_rc /= 0 then                      /* clean up and exit   */ 01000001
  "IMFEXEC MSG '.QAOSMTP2 ended with rc=" exec_rc"" 01010001
                                         01020001
"IMFEXEC MSG '.QAOSMTP1 EID' IMFEID "Ended'" 01030001

```

---

# Chapter 18 SNMP Solution

MAINVIEW AutoOPERATOR provides a pair of sample EXECs that can be used to generate SNMP Traps.

## EXECs

The EXECs for the SNMP Solution are

**QAOSNMP1** Resides in BBSAMP.

**QAOSNMP2** Resides in BBPROC.

**Note:** Browse the MAINVIEW AutoOPERATOR product libraries to view the most current version of these sample EXECs. BMC Software might ship PTFs at any time to enhance them.

This SNMP sample uses REXX sockets, which are part of the IBM TCP/IP product.

## QAOSNMP1

QAOSNMP1 is a sample REXX EXEC that demonstrates how to generate a SNMP Trap. The actual Trap generation logic is encapsulated in QAOSNMP2, which should be usable without modification by most customers.

To generate an SNMP Trap, copy QAOSNMP1 into UBBPROC using a new meaningful name. Replace the data in the user-input fields with the appropriate data for your installation.

You can also incorporate the logic in QAOSNMP1 into your own REXX application to generate SNMP Traps.

---

SNMP Traps are directed to SNMP Managers. Examples of SNMP Managers are PATROL® Enterprise Manager, Tivoli Enterprise Console, etc. You must coordinate your efforts with the administrators of your SNMP Manager. Consult them for any SNMP values of which you are unsure.

The user-input fields are as follows:

Name	Description
Community	A text string, usually used as a simple security mechanism.
SnmpManager	The IP address or host name of your SNMP Manager.
SnmpPort	The listening port number used by your SNMP Manager.
Enterprise	An object identifier that identifies the device that generates the Trap. <b>Note:</b> Nodes must be less than 128.
GenericTrap	An integer, specified from the SNMP-defined values for a generic Trap.
SpecificTrap	An integer, specified for a specific Trap.
TimeTicks	An integer; it usually represents the device uptime in hundredths of a second.
Object Identifier	An unique identifier for this event Trap. <b>Note:</b> Nodes must be less than 128.
OIDtype	The type of data; QAOSNMP2 supports Octet String (4) only.
OIDtext	Trap text.

A QAOSNMP1 sample EXEC follows:

```

/* rexx */ 00010000
/* */ 00020000
/* Sample EXEC which calls QAOSNMP2 to issue a */ 00030000
/* SNMP v1 Trap */ 00040000
/* */ 00050000
/* Nearly all parameters are passed to QAOSNMP2 */ 00060000
/* as positional parameters. QAOSNMP2 resides */ 00070001
/* in BBPROC. */ 00071001
/* */ 00080000
/* Community and OIDtext are passed on the data */ 00090000
/* stack. */ 00100000
/* */ 00110000
/* Note: Object Identifiers must begin with 1.3.x */ 00120000
/* Subsequent nodes must be less than 128 */ 00130000
/* eg. valid = 1.3.6.2.4 */ 00140000
/* invalid = 2.9.1 */ 00150000
/* invalid = 1.3.200 */ 00160000
/* */ 00170001
/* change log: */ 00180001
/* 9feb2001 bpo4874 shipped to make bpo4874 */ 00188002
/* prereq bpo4851 */ 00188102
/* 30jan2001 bpo4851 update comments to note */ 00188202
/* QAOSNMP2 is in BBPROC */ 00189001
/* 14apr2000 bpo4467 sample created */ 00190001
/* */ 00200001
/* */ 00210000
/* */ 00220000
/* this is a string ... primitive SNMP v1 security */ 00230000
Community = 'public' 00240000
/* */ 00250000
/* this can be any valid IP address or host name */ 00260000
/* eg. 132.60.1.3 or manager.bigcompany.com */ 00270000
SnmpManager = 'snmpMgr.BigCompany.com' 00280000
/* */ 00290000
/* this can be any valid port number (integer) */ 00300000
/* normally it is 162 */ 00310000
SnmpPort = 162 00320000
/* */ 00330000
/* this Object Identifier identifies the 'device' */ 00340000
/* which is generating the Trap */ 00350000
/* note: nodes must be less than 128 */ 00360000
Enterprise = '1.3.6.1.4.2' 00370000
/* */ 00380000
/* SNMP defined values for Generic Trap are */ 00390000
/* 0 = cold start */ 00400000
/* 1 = warm start */ 00410000
/* 2 = link down */ 00420000
/* 3 = link up */ 00430000
/* 4 = authentication failure */ 00440000
/* 5 = egp neighbor loss */ 00450000
/* 6 = enterprise specific */ 00460000
GenericTrap = 6 00470000
/* */ 00480000
/* specify an integer for Specific Trap */ 00490000
SpecificTrap = 0 00500000
/* */ 00510000
/* Time Ticks is an integer, it usually represents */ 00520000
/* the device uptime in hundredths of a second */ 00530000
TimeTicks = 5100000 00540000
/* */ 00550000
/* Object Identifier which uniquely identifies */ 00560000
/* this event (Trap) */ 00570000
/* note: nodes must be less than 128 */ 00580000
ObjectId = '1.3.6.1.4.1.1.2' 00590000
/* */ 00600000
/* OIDtype represents the type of data, */ 00610000
/* QAOSNMP2 supports Octet String (4) only */ 00620000
OIDtype = 4 00630000
/* */ 00640000
/* This is the text of the Trap */ 00650000
OIDtext = '123 this is the text of the Alert' 00660000
/* */ 00670000
/* */ 00680000
/* Issue NewStack to insulate us from any callers */ 00690000
address mvs 'newstack' 00700000
/* */ 00710000
/* put on data stack */ 00720000

```

---

```
push Community                                00730000
push OIDtext                                  00740000
                                                00750000
call QAOSNMP2 SnmpManager SnmpPort Enterprise , 00760000
    GenericTrap SpecificTrap TimeTicks ObjectId OIDtype 00770000
                                                00780000
/* destroy our stack                          */ 00790000
address mvs 'delstack'                        00800000
```

---

---

# Chapter 19 Using the MAINVIEW AutoOPERATOR Web Sample

This chapter describes how to implement the MAINVIEW AutoOPERATOR Web sample.

AutoOPERATOR Web consists of a set of samples distributed with MAINVIEW AutoOPERATOR. The code for these samples was developed for the IBM HTTP Server for OS/390 (WebSphere).

The purpose of these samples is to demonstrate MAINVIEW AutoOPERATOR automation possibilities in Web browser environments. The samples will not be upgraded or modified in response to user requests. You can choose to personally implement new samples or modify existing samples, but these changes are at your discretion.

## Introduction

The MAINVIEW AutoOPERATOR Web solution provides sample code where REXX EXECs use AOAnywhere statements in a Common Gateway Interface (CGI) script. When the IBM HTTP Server for OS/390 (WebSphere) is in place, the sample code delivers examples of four AutoOPERATOR features viewed from a Web browser:

- Continuous State Manager (CSM)
- AutoOPERATOR ALERTs
- Automation Reporter
- Commands

This chapter documents the four samples.

The samples described in this chapter are not applications. Use these samples to get ideas about how you can create your own automation from a Web browser.

The sample code also delivers MAINVIEW for DB2 examples.

## What AutoOPERATOR Web Is

AutoOPERATOR Web is based on the AOAnywhere Application Program Interface. The AOAnywhere statements provide the ability to process key AutoOPERATOR functions from outside of the MAINVIEW AutoOPERATOR environment. With AOAnywhere, you can perform AutoOPERATOR automation from a batch program, an IMS MPP, a TSO CLIST, NetView, or UNIX System Services.

The function set of AOAnywhere covers basic AutoOPERATOR functions such as the variable interface, EXEC and ALERT functionality, and other automation functions. When present, SYSPLEX (XCF) connectivity provides multi-system support. AOAnywhere allows AutoOPERATOR programs (EXECs) to be run anywhere.

For more information about the AOAnywhere API and syntax, refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide*.

## How AutoOPERATOR Web Is Used

The AutoOPERATOR Web samples demonstrate how you can access MAINVIEW AutoOPERATOR from a Web browser. Accessing AutoOPERATOR from a browser provides you the ability to display automation data and drive automation anywhere where you can open a browser.

For example, you can implement graphical displays that illustrate the hierarchical arrangement of CSM objects or different locales interwoven in an automation Web. You can also use bar and pie charts. Graphically, these displays present a picture that is much more accessible to the operator, systems programmer, or support staff.

In addition, many forms of data representations can be much more easily implemented by using HTML, XML, Java, and so on. Even ordinary displays (such as tabular data formats) can become more usable in a browser where table sizing, scrolling, and panning are available.

## What the Sample Web Pages Look Like

When installed, the samples create pages that have a similar look and feel, where each sample shows a different aspect of AutoOPERATOR as accessed with AOAnywhere statements in a CGI script. The samples create only a few pages, which are split into a number of individually scrollable panes.

All pages contain a plain white background with an AutoOPERATOR watermark. All of the pages are designed for a browser maximized at 1024x768. For most of the text, a sans serif variable pitch font is used. Detail displays use a serif variable pitch font.

## Prerequisites

The section describes what you should know and how to prepare to install the AutoOPERATOR Web samples.

## What You Need to Know

You should be familiar with the basic function set of AOAnywhere statements. For more information about the AOAnywhere API and syntax, refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide*.

You should have a basic understanding of IBM UNIX System Services (USS) because you will have to make changes to UNIX files. For beginners, you can use the IBM TSO ISHELL EXEC to enable you to view directories and edit USS files. This facility requires that your TSO PROC include the SBPX#### libraries; otherwise, the ISHELL EXEC does not function properly. However, someone proficient with UNIX programming can use RLOGIN and use VI to modify these files. Additionally, you might use the OMVS and OEDIT commands to configure your system. You should also be familiar with the IBM HTTP Server for OS/390.

You also need to be aware of the following considerations:

- The samples are written in standard HTML, and the underlying CGI scripts are written in REXX. You should have some understanding of these areas. Only standard HTML is used (without particular extensions), and no style sheets are used. The graphics are in GIF format. In a number of instances, the sample might contain a small amount of Java Script code.

- These HTML files and CGI scripts must be distributed in separate directories, and these directories must have entries added to the WebSphere configuration file to make them accessible to WebSphere.
- You need to review your security. USS uses the UNIX security model. Implementation of any Web application can create security exposures if the application is not properly protected. Ensure that you are familiar with your requirements. If you are implementing security for the first time, you should consult with your UNIX system administrator for security issues. Consistent user ID numbers, group ID numbers, and security policies between platforms will make USS a secure environment.
- You must ensure that OMVS segments exist for user IDs that will access the AutoOPERATOR Web samples.

## Warning about Case Sensitivity

Everything in UNIX is case sensitive. Using file names, using directories, or simply modifying a file can have a serious impact if uppercase and lowercase characters are not used properly. If any problems do arise or if files are empty, ensure that you have used the correct case of every letter (for example, `/usr/local/data.txt` is different from `/usr/local/dAta.txt`).

## What Information to Gather

The installation process requires that you supply the following information:

**Installation directory name:** The AutoOPERATOR Web samples must reside in a USS directory. The default name is

```
/usr/local/bmc/ao62
```

where `/usr/local` is a standard UNIX name for local directories. Many sites do not follow this standard for installing products. You must determine the correct location for your environment. If a directory level is missing, the installation EXEC automatically creates it for you.

**Group ID name:** UNIX security uses group IDs. These group IDs are RACF-defined groups with an OMVS segment. The group ID that you specify is used for any new directories and all files that are created. Users in this group ID have access to the protected and non-protected applications.

**Superuser authority:** Superuser, also referred to as system administrator or root user, has access to write and delete all files and directories. To be defined as superuser, you must have BPX.ADMIN authority, or the UID number must be 0. If you are using BPX.ADMIN authority, you might also need to change the owner of the HTTP server configuration file.

You do not need superuser authority to install the AutoOPERATOR Web samples. Use of superuser authority simply makes the job easier. If you are not superuser, verify that the user ID used for installation can create the installation directory and create files and subdirectories in this directory. Write authority is also required for the HTTP server configuration file.

## What Software Must Be Installed

IBM's HTTP Server for OS/390 (WebSphere) must be installed and operational. You must also have access to USS.

**Note:** You should be aware of the following information:

- These samples have been tested on IBM HTTP server and they function in this environment.
- These samples should also function for any HTTP server available on OS/390 that supports REXX as a CGI language.
- These samples should also work with IBM WebSphere Application Server, IBM Domino Go Webserver, or APACHE without any modifications. All other HTTP servers might require some minor modifications.

# Installing and Customizing the Samples

This section describes the installation procedure.

## Running the Installation EXEC

To run the installation EXEC, you must be logged on as a TSO user who has access to USS.

- Step 1** To start the EXEC, issue the following command from the ISPF command line:

```
TSO EXEC 'hilvl.BBSAMP(QAOWEBIN)'
```

where `hilvl` is the data set prefix for the BBSAMP member.

The first screen prompts you for a function.

- Step 2** Enter **INSTALL** to install the product (or enter **REMOVE** to remove the product from USS).

- Step 3** Enter the name of the installation directory and group ID.

Use the information that you obtained from “What Information to Gather” on page 19-4.

- Step 4** Press **Enter**.

Messages about the installation are displayed.

- Step 5** Review these messages and ensure that the successful installation message is displayed.

## Configuring the HTTP Server

During the installation, the name of a USS file is displayed, `http.conf.changes`. This file resides in the **secure** subdirectory of the installation directory and contains directions for applying necessary changes to the HTTP server configuration file.

The HTTP server uses a pound sign (#) in column 1 to identify comments. The directions in the **http.conf.changes** file are denoted with a pound sign in column 1 so that they are not interpreted as configuration statements.

Enter the following information as instructed in the file:

- **PASS**—Defines a directory (including subdirectories) where HTML and graphic files are located. A virtual directory name is related to this real directory. The virtual directory name must be used in the Web browser to access the files in the real directory.
- **EXEC**—Defines a directory where the CGI scripts are located. A virtual directory name is related to the real directory. The virtual directory name must be used in the Web browser to access the files in the real directory.
- **PROTECT**—Defines which virtual directory names have a security method applied. The user ID is determined through the **PROTECTION** clause. Virtual directories that are not covered by a **PROTECT** clause will use the **PUBLIC** user ID specified.
- **PROTECTION**—Defines the security method and how the user IDs are determined. Many security methods are available in UNIX but in this case, the RACF sign-on method is implemented. Your RACF user ID and password will be used. When a Web browser accesses any directory covered by a **PROTECT** clause that specifies this **PROTECTION** clause, the browser will be prompted for a valid RACF user ID and password before access is allowed.

You must restart the HTTP server after these changes have been made.

## Making BBSAMP Available to AutoOPERATOR

In addition to the CGI scripts located in USS, the AutoOPERATOR Web samples have EXECs that execute in the AutoOPERATOR address space. These EXECs must be made available by ensuring that the BBSAMP data set is in the SYSPROC concatenation of the AutoOPERATOR PROC.

## Meeting AOAnywhere Requirements

AOAnywhere syntax is used throughout the CGI scripts. For this process to occur, BBLINK must be available to the CGI scripts. The CGI scripts run under USS and therefore, it might be easier to add BBLINK to LNKLST rather than trying to define a STEPLIB to a USS task.

When BBLINK is added to LNKLST, there might be a requirement to authorize BBLINK by using APF, depending on the BMC Software products in use. For details, see “Performing Automation Using AOAnywhere” in the *MAINVIEW AutoOPERATOR Advanced Automation Guide*.

## Reviewing Security

BMC Software recommends that your UNIX system administrator review the security that has been implemented. The administrator will be familiar with areas that might cause problems or areas that might be more vulnerable.

Subdirectory `secure` (and all of its subdirectories) in your installation directory are covered by one PROTECT clause that you added. A valid user ID and password must be entered before access is allowed to these secured applications. Subdirectory `secure` defaults to allowing access to users in the group ID specified during installation. If a user has access to `secure`, that user has access to all files and subdirectories in this directory. If `secure` existed prior to running the installation EXEC or was modified after installation, access to the files might be different than documented here.

All other files and subdirectories that are not in the `secure` subdirectory have been made public. These applications are display only.

You can limit public access to display-only applications by increasing the scope of the PROTECT clause in the HTTP server configuration. See “Configuring the HTTP Server” on page 19-6.

You might also consider limiting user access to directories by changing directory access modes. Because the installation process does not change directory access modes if they exist prior to installation, changes to directories are retained when updates are applied. Files on the other hand are replaced. Their file modes are lost. All changes to files will need to be re-applied when installing new versions.

For the best security results, consult with your UNIX system administrator.

## Accessing Multiple Systems

The samples were written with the AOAnywhere XCF capability to access AutoOPERATOR subsystems across the SYSPLEX. To implement this feature, at least one AutoOPERATOR 6.1 (or later) must be running on the same system as the HTTP server. If you have systems in different SYSPLEXs, multiple HTTP servers that will run these samples, or a configuration where the samples must run under different HTTP servers, you need to install the samples on each affected system. Most installations of USS do not share Hierarchical File System (HFS) directories. If this is your case, you need to install on each system where the AutoOPERATOR Web samples run under a HTTP server.

## Accessing the Samples

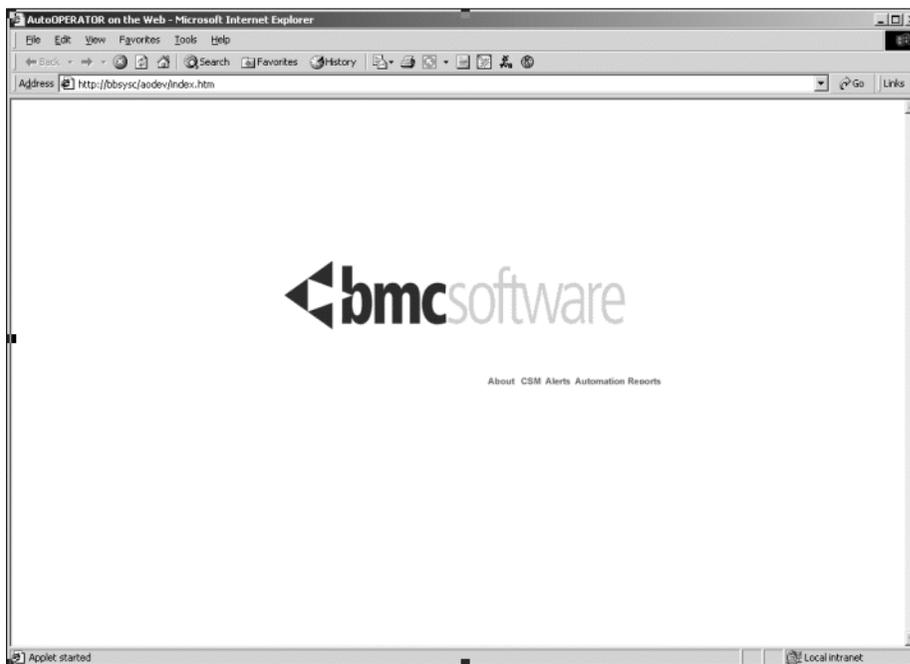
To access the samples, start a Web browser on your PC or UNIX system. Change the address in the Web browser to the DNS-defined name of your OS/390 TCP/IP address that is running the HTTP server, followed by /ao. If you do not know the DNS-defined name, substitute the TCP/IP address for that system (for example, `bmcysa/ao` or `192.0.0.1/ao`). If you did not change the HTTP server WELCOME statement, `index.html` will be automatically displayed. If `index.html` was removed, add `/index.html` to the address in the browser.

## Viewing the Entry Page

The entry page consists of a BMC Software logo and hyperlinks (see Figure 19-1).

Below the logo is a small navigation bar with hyperlinks. The hyperlink text changes colors when the cursor passes over it. These same hyperlinks are present on all pages. By clicking one of these hyperlinks, you can access the other pages.

**Figure 19-1** AutoOPERATOR Web: Entry Page

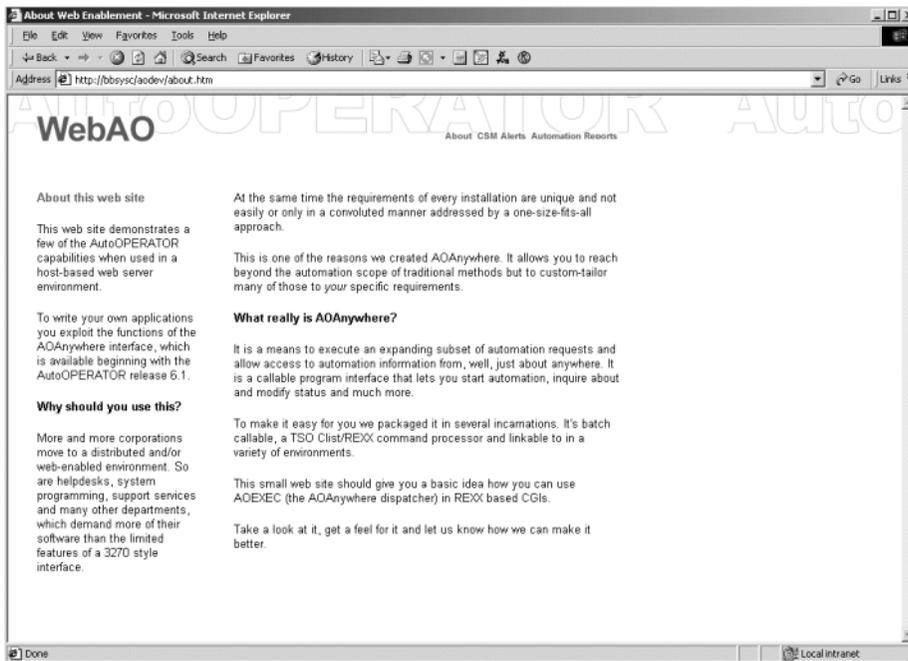


## Viewing the About Page

The About page shows the WebAO heading at the top of the page, which is common to all pages except the entry page. Using the navigation bar on the right side, you can link to all other pages on the site. The WebAO heading in the upper left is also a active hyperlink that changes color as the cursor passes over it. You can click on this heading to access the entry page.

The About page (see Figure 19-2) introduces the fundamentals of the site and AOAnywhere.

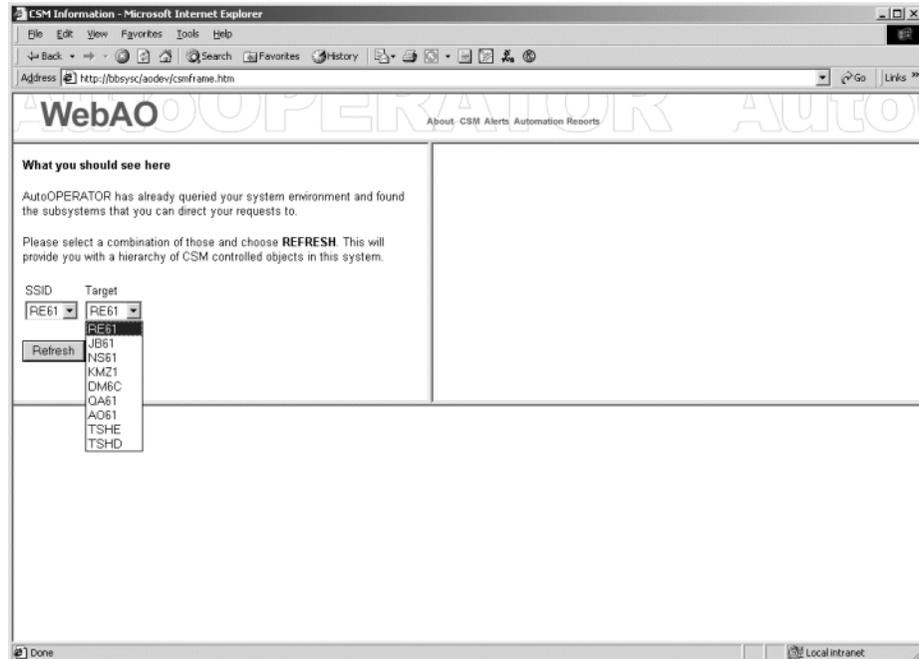
**Figure 19-2 AutoOPERATOR Web: About Page**



## Viewing the CSM Page

This sample requires that CSM be active on the targeted system. The CSM page, like all other WebAO informational screens, is split into four distinct panes (see Figure 19-3).

**Figure 19-3 AutoOPERATOR Web: CSM Page**



The top pane features the usual navigation and identification format. The middle left pane displays information, instructions, and two drop-down boxes. The drop-down box on the left side displays all AutoOPERATOR subsystems (except BBI-3 subsystems) on the current MVS image. Using the drop-down box on the right side, you can access any connected AutoOPERATOR system, whether it is on the current image or a remote image. When the request is directed against only one subsystem (without routing), both boxes should contain the same name.

You can use the Refresh button to retrieve (and refresh) data. No data is shown in Figure 19-3 because a target has not been chosen.

The initial set of data is overview material and is displayed in the bottom frame when a search is conducted, as shown in Figure 19-4.

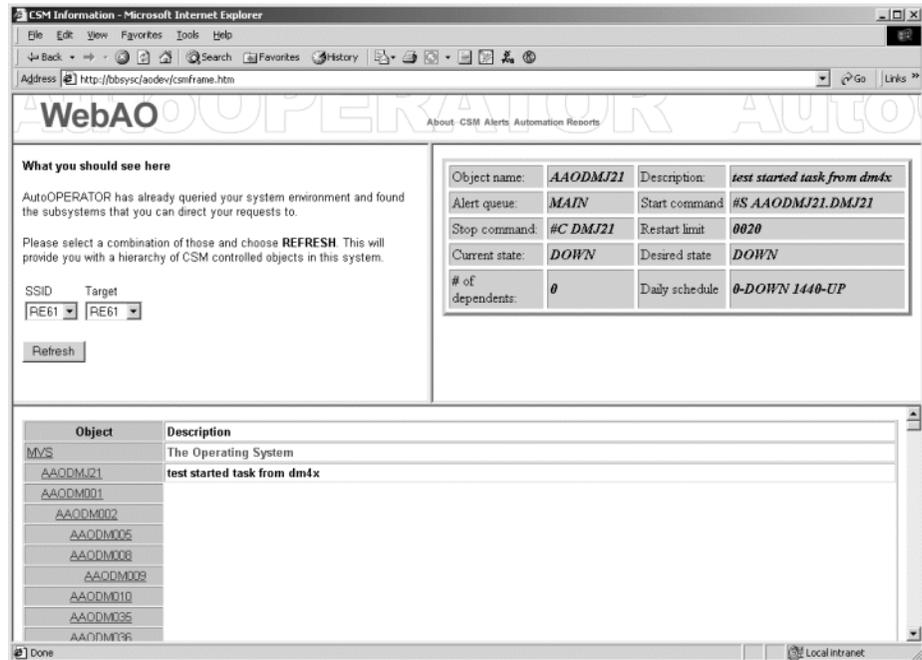
**Figure 19-4 AutoOPERATOR Web: CSM Page with Objects**



In Figure 19-4, the bottom pane shows CSM objects arranged hierarchically, where the indented position denotes the parent-child relationships. The pane is scrollable. The right portion of the pane shows the descriptions that are associated with each object. The descriptions are color-coded according to their hierarchy level.

The objects themselves can be hyperlinks and are DHTML animated (where they change color as the cursor passes over them). As hyperlinks, they are not color-coded according to their hierarchy level, but they change colors according to whether they have been viewed. If you click on one of the objects, detailed data is displayed in the middle right pane (see Figure 19-5).

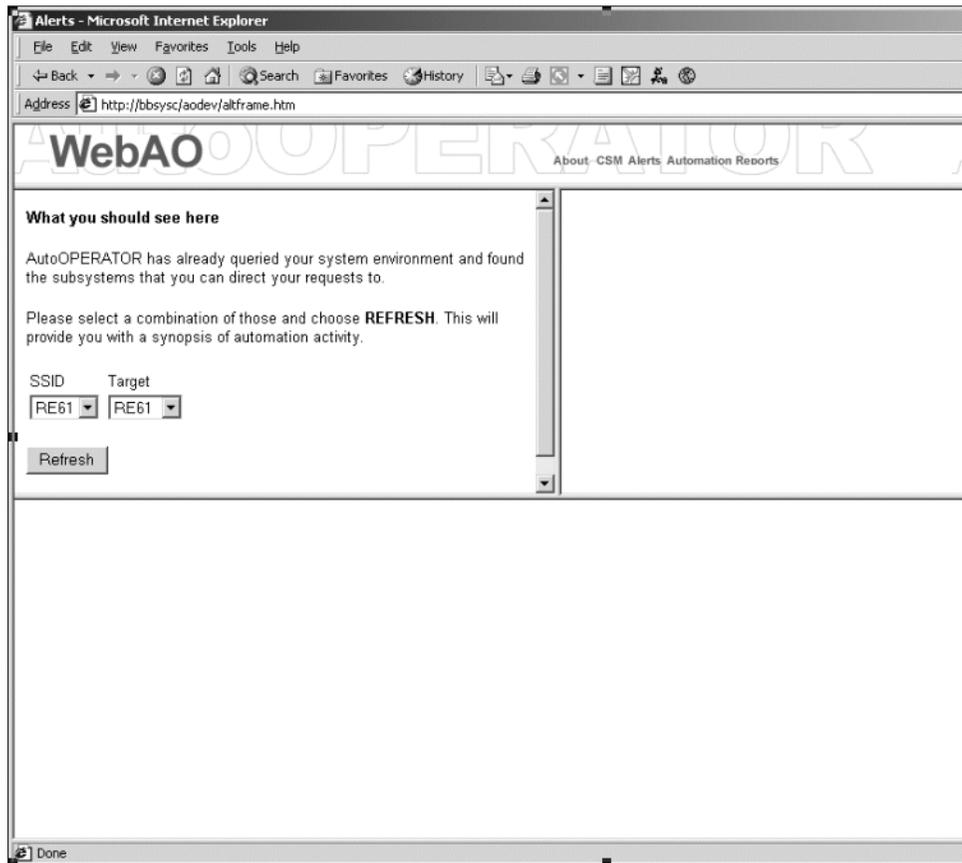
**Figure 19-5 AutoOPERATOR Web: CSM Object Description**



## Viewing the ALERTs Page

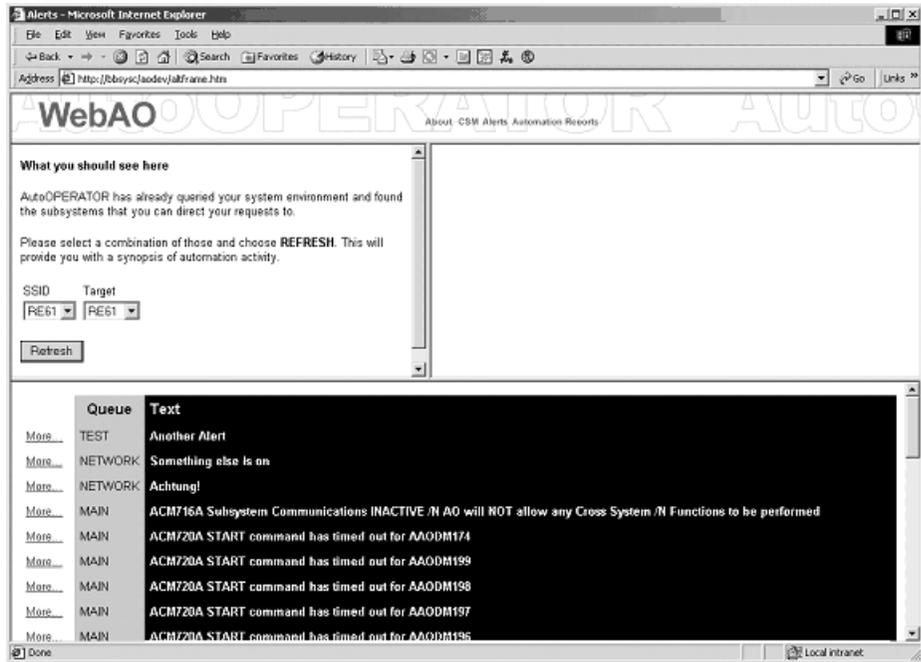
The initial ALERTs page is identical to the CSM page and Automation Reporter page, except for the name (“ALERTs”) in the title bar of the browser window (see Figure 19-6).

Figure 19-6 AutoOPERATOR Web: ALERTs Page



A **Refresh** button activates the actual retrieval (and refresh) of data. The data appears in the bottom pane (see Figure 19-7).

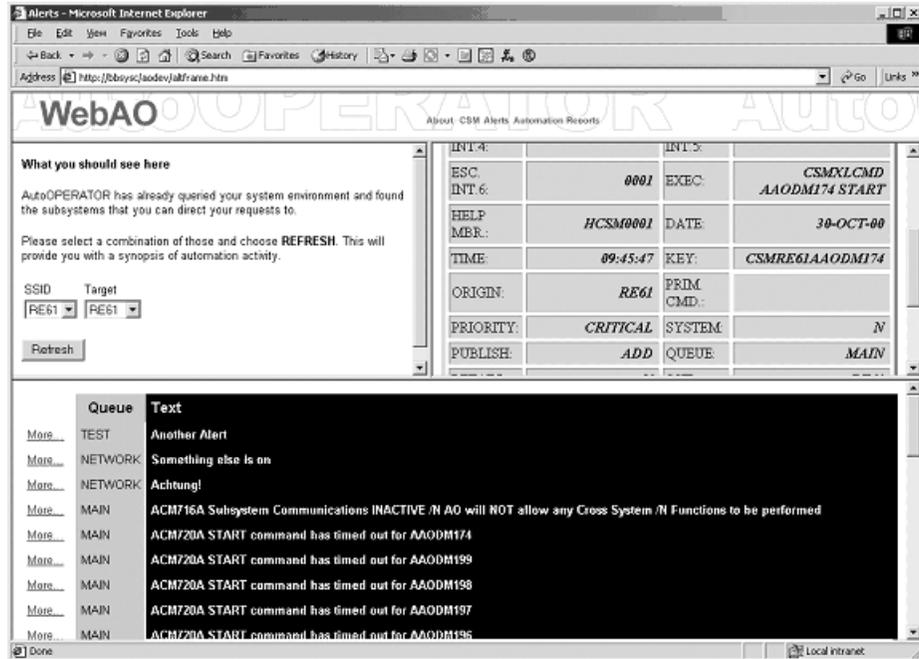
**Figure 19-7 AutoOPERATOR Web: ALERTs Page with Data**



The first column of the bottom pane consists of [More...](#) hyperlinks that link to further information. The **Queue** column shows the name of the ALERT queue.

The third column uses a black background for better visibility of the ALERT colors, which denote priority and severity. Clicking the **More...** hyperlink displays a scrollable middle right pane where all data that is associated with this particular ALERT is displayed.

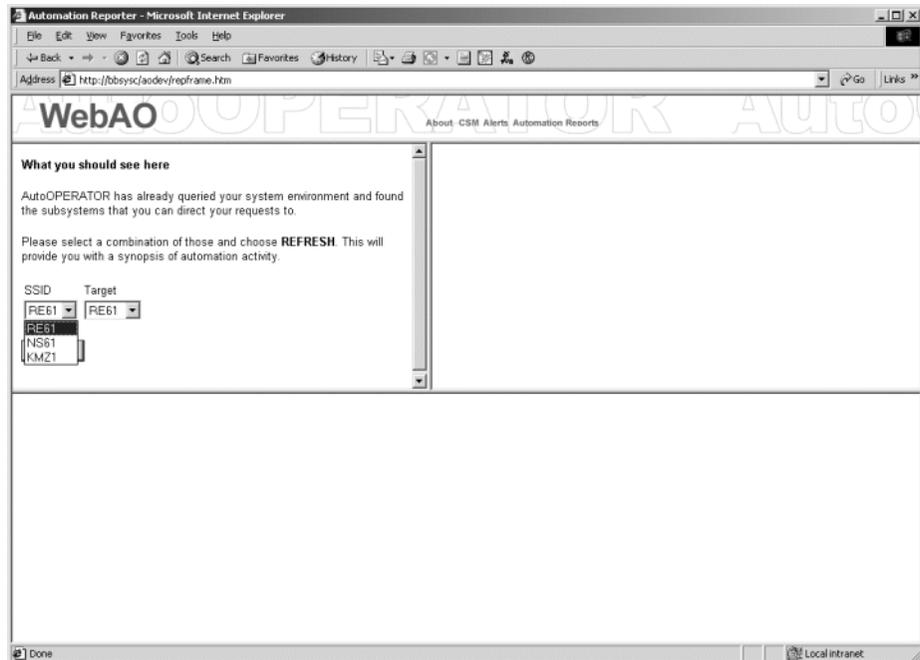
**Figure 19-8 AutoOPERATOR Web: Scrolling ALERTS**



## Viewing the Automation Reporter Page

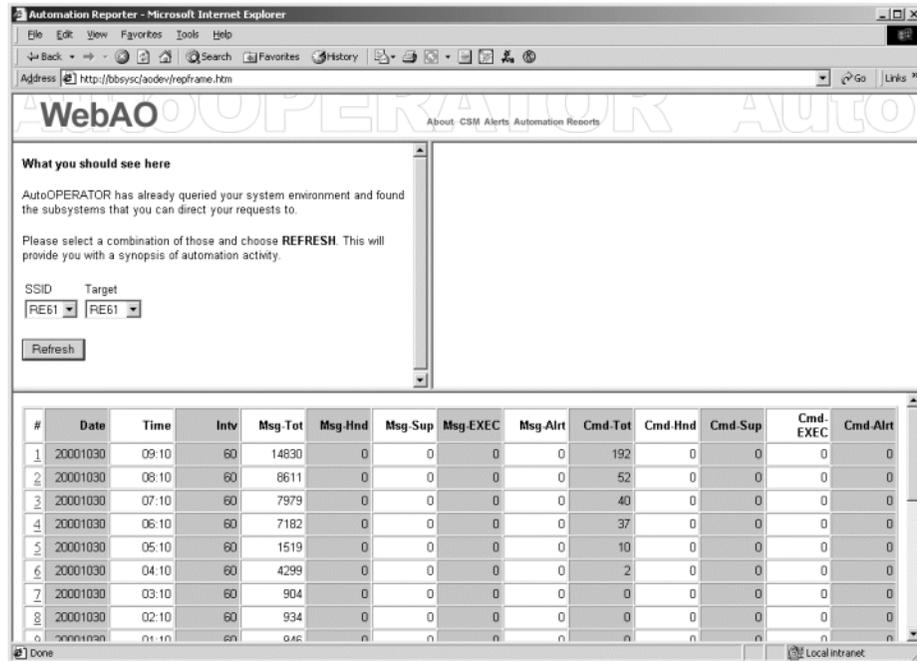
This sample requires Automation Reporter to be active on the targeted system. The initial Automation Reporter page is identical to the CSM page and ALERTs page, except for the name (“Automation”) in the title bar of the browser window (see Figure 19-9).

**Figure 19-9** AutoOPERATOR Web: Automation Reporter Page



When you specify subsystems and targets from the drop-down boxes, automation statistics are displayed, sorted from most recent statistics oldest statistics, as shown in Figure 19-10. This data is directly read from the Automation Reporter data gatherer.

**Figure 19-10 AutoOPERATOR Web: Automation Reporter Data Displayed**



If you click an interval in the first column, the full set of statistics is displayed in a scrollable pane, as shown in Figure 19-11:

**Figure 19-11 AutoOPERATOR Web: Automation Reporter Statistics Displayed**

The screenshot shows the 'Automation Reporter - Microsoft Internet Explorer' window. The main content area is titled 'WebAO' and 'About CSM Alerts Automation Reports'. It features a section 'What you should see here' with instructions to refresh the data. Below this are dropdown menus for 'SSID' (set to 'RE61') and 'Target' (set to 'RE61'), and a 'Refresh' button. To the right, a summary table displays statistics for the selected interval (20001030, 09:10, 60s). Below the summary is a large table with 15 columns: #, Date, Time, Intv, Msg-Tot, Msg-Hnd, Msg-Sup, Msg-EXEC, Msg-Alert, Cmd-Tot, Cmd-Hnd, Cmd-Sup, Cmd-EXEC, and Cmd-Alert. The table contains 8 rows of data for the date 20001030.

Sample-Date	20001030	Sample-Time	09:10	Intv-Time	60
Msg-tot	14830	Msg-Hnd	0	Msg-Sup	0
Msg-Sup	0	Msg-EXEC	0	Msg-Alert	0
Cmd-Tot	192	Cmd-Hnd	0	Cmd-Sup	0
Cmd-EXEC	0	Cmd-Alert	0	Cmd-HCS	0
Jrnl-Tot	125	Jrnl-Hnd	0	Jrnl-Sup	0
Jrnl-EXEC	0	Jrnl-Alert	0	Jrnl-HCS	0
Time-Tot	0	Time-Hnd	0	Time-Sup	0
Time-EXEC	0	Time-Alert	0	Time-HCS	0

#	Date	Time	Intv	Msg-Tot	Msg-Hnd	Msg-Sup	Msg-EXEC	Msg-Alert	Cmd-Tot	Cmd-Hnd	Cmd-Sup	Cmd-EXEC	Cmd-Alert
1	20001030	09:10	60	14830	0	0	0	0	192	0	0	0	0
2	20001030	08:10	60	8611	0	0	0	0	52	0	0	0	0
3	20001030	07:10	60	7979	0	0	0	0	40	0	0	0	0
4	20001030	06:10	60	7182	0	0	0	0	37	0	0	0	0
5	20001030	05:10	60	1519	0	0	0	0	10	0	0	0	0
6	20001030	04:10	60	4299	0	0	0	0	2	0	0	0	0
7	20001030	03:10	60	904	0	0	0	0	0	0	0	0	0
8	20001030	02:10	60	934	0	0	0	0	0	0	0	0	0

## Viewing the Commands Page

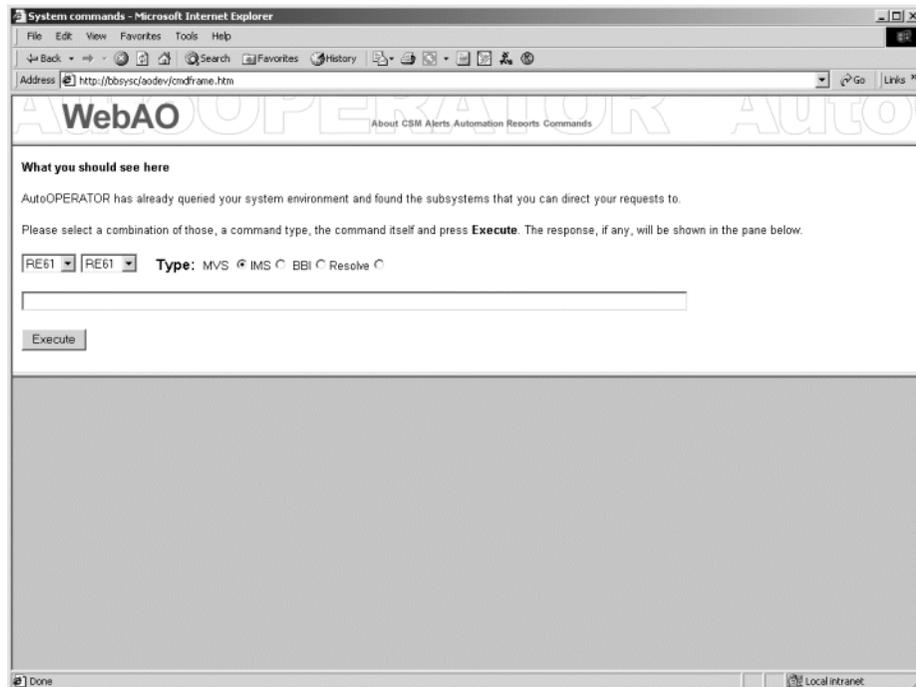
The initial Commands page is divided into three panes:

- WebAO heading and hyperlinks in the top pane
- various input fields in the middle pane
- an output pane at the bottom

The output pane is intentionally silver. In the middle pane, you can use the drop-down boxes to select local and target subsystems, and you can specify the type of command to be issued (prefix characters such as # and. are acceptable). Currently, MVS, IMS, BBI, and Resolve commands may be selected.

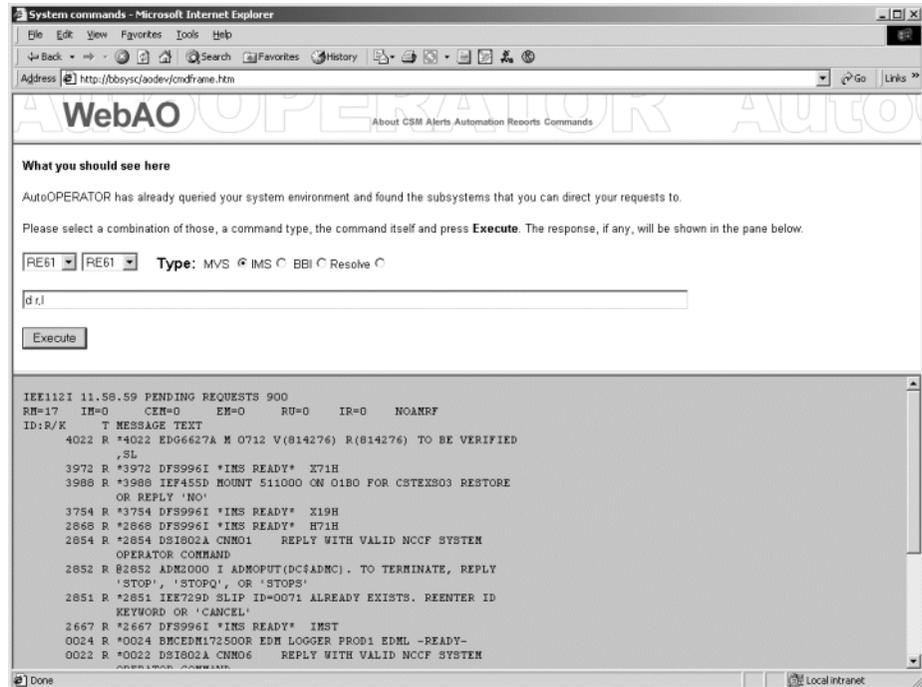
The text box (below the drop-down boxes) is designated as the input area for the command itself, and the Execute button submits the command.

**Figure 19-12 AutoOPERATOR Web: Commands Page**



When you make selections, output is displayed at the bottom (see Figure 19-13):

**Figure 19-13 AutoOPERATOR Web: Command Output Displayed**



The output is scrollable.

## Debugging the Distributed CGI Scripts

All of the CGI scripts were created to be as independent of their environment as possible. With the exception of parameter parsing, these EXECs will run successfully in any environment.

The following sections contain tips for understanding and debugging the sample CGI scripts.

## Understanding CGIs

The AutoOPERATOR Web samples consist of HTML tagging and CGI scripts. The CGI scripts are created as REXX EXECs. These EXECs obtain their input from the stack and then parse the information according to CGI requirements. All other statements in the AutoOPERATOR Web samples are compatible with any available REXX environment.

A CGI runs in the same way as a normal REXX EXEC except that all SYSTSPRT output is directed to the Web browser (either Netscape or Microsoft Internet Explorer). The 'SAY "Content-type: text/html "' statement is required because it tells the browser what to do with the SYSTSPRT data. In the sample code, it is coded to tell the browser to expect HTML from SYSTSPRT.

## Debugging the CGI Script in a TSO Environment

All sample CGIs can be copied to a CLIST library and run as TSO CLISTs. The only modification required is parsing the parameters and getting the test parameters into the EXEC. To do this, disable the parse routine and manually set the required variables. The output that would be directed to the browser now appears on your TSO session. Be aware that HTML formatted data will appear on your screen, enclosed in < > characters.

## Debugging the CGI Script without HTML Decoding

You can use the REXX TRACE command when running a CGI script. If TRACE is specified, only SYSTSPRT data occurring after the SAY "Content-type: text/html" statement appears in the browser window. One problem with this process is that trace data will be displayed as a continuous stream of data. To avoid this problem, turn off HTML decoding by changing the content type to "Content-type: text/plain".

## Avoiding Frames in the Display

Parts of the sample code contain HTML frames. Frames split the browser screen into windows. If the screen is split into frames, you can determine the frame definitions by changing the address specified in the address field of the browser. Each frame has HTML or CGI coding that has been executed. Determine which HTML or CGI code was executed for that frame and use that code in the browser's address field. This process separates the executable data into separate browser screens to make problem diagnosis a little simpler.

## Displaying the HTML Source in a Frame

You can choose to display only the HTML source for a frame definition. To display the source for a frame, right-click anywhere in the frame and choose Source or choose Source from the View menu.

## Refreshing HTML Source Code

HTML source code is stored in the browser's cache. To refresh changes that are made to the HTML source code

- For Internet Explorer, hold down the **Ctrl** key and click on the browser's Refresh button.
- For Netscape, hold down the **Shift** key and click **Refresh**.

CGI output is forced to refresh by using the PRAGMA directive with NO-CACHE into its output.

## Common Error Messages

This section contains common error messages that you might encounter and describes what they mean.

**Table 19-1 Common Error Messages (Part 1 of 2)**

Message	Description
Can't locate HTTP configuration file.	The configuration file is listed in the JCL startup proc for your HTTP server, after the -r option. If you are using the free IBM HTTP server and if you have only one copy, the name of the configuration file will be /etc/http.conf. If a period precedes the file name (/etc/http.conf), you need to check the LEPARM variable in the proc for the specification of _CEE_ENVFILE. The file specified here contains a home directory name. The file will be located relative to this home directory name if the configuration file name was specified beginning with a period.
"*_* address linkmvs "AOEXEC P1" +++ RC(-3) +++"	If this message is displayed in a frame, it means that AOAnywhere is not available in linklist or in the USS steplib variable.
Web browser 403 - 403	A security problem has occurred. You should review each directory level of your installation directory. Ensure that READ and EXECUTE are on of the GROUP and OTHER modes of each subdirectory.

**Table 19-1 Common Error Messages (continued) (Part 2 of 2)**

<b>Message</b>	<b>Description</b>
Web browser 404 - 404	This message indicates an "address cannot be found" condition. If this message occurs with the first screen, you need to verify that the PASS statement that you added is not preceded by the PASS /* in your HTTP configuration file. The address search starts from the top of the configuration file and stops when the first PASS or EXEC address is true for this address (for example, if /* precedes /ao/*, execution would stop at /* because it matches all address).
Frame is missing the available system list and the REFRESH button	The server side includes are not working in the HTTP server. Ensure that your HTTP configuration file specifies at least "imbeds on SSIOnly" and the ADDTYPE for ".shmtl", and ensure that text/x-ssi-html is specified.

---

# Chapter 20 Using the Paging Sample

MAINVIEW AutoOPERATOR provides sample code that enables you to communicate with one- or two-way wireless devices, including both text and numeric pagers. You can also obtain confirmations and responses if the destination device and its provider offer these capabilities.

This sample was developed by using the Simple Network Paging Protocol (SNPP rfc 1861) as implemented by SkyTel Communications, Inc.

## EXECs

The EXECs for the Paging sample are

**QAOSNPP1** Contains the calling communication logic.

**QAOSNPP2** Contains the communication logic.

The EXECs are not compiled, so you can modify them for your application.

## QAOSNPP1

The QAOSNPP1 EXEC is the calling sample and does not have input parameters.

---

## QAOSNPP2

The QAOSNPP2 EXEC can have from 0 to 9 parameters, depending on the mode. Each parameter (except the message text) has the following format:

---

PARAMETER\_NAME ( VALUE )

---

Parameters are separated by blanks and can be in any order. The parameters are described in the following table.

Parameter	Description
HELP	Optional. Default is HELP(NO). If HELP(YES) is specified, if the program is called without parameters, or if HELP is specified with no parameters, the calling format is printed to the BBI Journal.
MODE	Optional. Default is MODE(SEND). MODE can have one of the following values: <ul style="list-style-type: none"><li>• SEND - Send the message.</li><li>• STATUS - Check the message status.</li></ul>

### Parameters for MODE(SEND)

If you specify MODE(SEND), additional parameters are available that you can set. These parameters are described in the following table.

**Table 20-1 Parameters for MODE(SEND (Part 1 of 2)**

Parameter	Description
SERVER	Required. Specify the Internet name or address of the message server provider. For example, for SkyTel, the Internet name is snpp.skytel.com.
PORT	Required. Specify the message server provider port to which the program should connect. In the sample, the port is set to the value 7777.
PIN	Required. Specify the service provider access PIN or a direct telephone number.
USER	Optional. Specify the name of the user to whom you are sending a message. This value is used in the messages and ALERTs issued by the program. Default is NONE.
TIMEOUT	Optional. Specify the time (in seconds) that the program can wait for the TCP/IP response before timing out. Default is 10 seconds.

**Table 20-1 Parameters for MODE(SEND (Part 2 of 2)**

Parameter	Description
CONF	Optional. You can specify CONF(YES) if you want to receive a confirmation of a successful message delivery. CONF(YES) can be used only with two-way pagers or telephones with a messaging service. Default is CONF(NO). CONF(NO) can be specified when calling any pager or telephone.
DEBUG	Optional. Specify DEBUG(YES) to print the contents of all TCP/IP exchanges (and some other debugging information) to the BBI Journal.
<p><b>Note:</b> Message text should be written after key parameters. If the message text must contain an open parentheses symbol, (, separate it with &lt;&lt;. When QAOSNPP2 is called from an IMFEXEC SELECT statement, the total statement length is restricted to 256 characters (including information content). If QAOSNPP2 is called from another EXEC with a CALL statement, the length can be up to 2,500 characters.</p>	

**Parameters for MODE(STATUS)**

Use MODE(STATUS) only when an ALERT is created when a message is sent. MODE(STATUS) has one parameter, which is the variable name containing the list of values for status requests.

**Processing Flow**

The calling EXEC, QAOSNPP1, has four components:

- A tuning section where you can change parameters for your needs
- A checking section to verify the information
- The CALL statement
- A section for result messages processing

QAOSNPP2 contains the main logic and performs the following tasks:

- It extracts and checks parameters.
- It prints the calling format and exit if help is requested.
- It connects to the server/port and performs the TCP/IP message exchange with a server.
- In the CONF(YES) mode, it creates a MAINVIEW AutoOPERATOR ALERT that escalates within a minute and calls a follow-up EXEC in STATUS mode. It creates a unique variable name formatted PAGING.xxxxxxxx (where xxxxxxxx are the last eight numbers of the

---

current time) and puts all necessary information into a long profile variable with this name. The variable name becomes the second parameter of QAOSNPP2.

**Note:** In the CONF(NO) mode, no more actions are required.

- When QAOSNPP2 is called in STATUS mode, it sends a status request to the server and replaces the original ALERT with the ALERT escalating up (if the message is still not delivered). Otherwise, it sends a status request to the server and replaces the original ALERT with a clearing ALERT escalating down.

## Variables Returned from QAOSNPP2

If the QAOSNPP2 EXEC is invoked from IMFEXEC SELECT, the following variables are returned:

- IMFCC, which contains the return code from the IMFEXEC SELECT statement.
- IMFRC, which contains the condition code returned on the IMFEXEC EXIT statement of QAOSNPP2 when IMFEXEC SELECT WAIT(YES) is specified.
- LOCAL variables APFCC and APFRC, which contain condition and return codes.
- LOCAL variables APFLN1 through APFLNn, which contain additional information.
- LOCAL variable APFNOL, which contains the number of APFLN1-Nn variables.

The following table contains the error messages and the IMFRC, APFRC and APFCC values that are found in LOCAL variables APFLN1 through APFLNn.

Error Message in APFLNn	IMFRC	APFCC	APFRC
PF0000I SUCCESS	0	0	0
PF2001E Message timed out.	8	4	1
PF2002E <Message from the service provider>	8	4	2
PF2002E Transmission ID Error	8	4	2
PF2003E PIN is not recognized by service provider.	8	4	3
PF3001E QAOPAGE1: SOCKET(SOCKET) PF3001E QAOPAGE1: SOCKET(CONNECT) PF3001E QAOPAGE1: SOCKET(WRITE) PF3001E QAOPAGE1: SOCKET(SELECT)	8	8	TCP/IP return code
PF4001E QAOPAGE1: User ID is unknown.	8	12	1
PF4002E QAOPAGE1: Page service provider is unknown.	8	12	2
PF4003E QAOPAGE1: Unable to initialize SOCKET	8	12	3
PF5051E QAOPAGE1: Error found in MODE parameter	8	16	51
PF5052E QAOPAGE1: Error, SERVER or PORT parameter is not provided.	8	16	52
PF5053E QAOPAGE1: Error in CONF parameter (YES/NO).	8	16	53
PF5054E QAOPAGE1: Message text is not provided.	8	16	54
PF5055E QAOPAGE1: PIN number is not provided or not numeric.	8	16	55
PF5056E QAOPAGE1: Error in TIMEOUT parameter, it is not numeric.	8	16	56
PF5057E QAOPAGE1: Error in DEBUG parameter (YES/NO)	8	16	57
PF5061E QAOPAGE1: Error in parameter structure	8	16	61
PF5062E QAOPAGE1: Unknown parameter	8	16	62
PF5063E QAOPAGE1: Error during VGETL	8	16	63



---

# Chapter 21 Using Sample Solutions

The MAINVIEW AutoOPERATOR product provides a set of Sample Solutions for automation tasks in data centers. This set consists of Rules, EXECs, Timer Facilities, and SYSPROG commands.

Use the Sample Solutions to accomplish many of the initial automation tasks quickly and efficiently after you install the AutoOPERATOR product.

**Warning!** With the Sample Solutions, BMC Software delivers sample source code that is to be used as examples for solution implementation.

BMC Software Customer Support will not provide support for any Sample Solution that has been modified. Customers seeking technical support from BMC Software must be prepared to demonstrate that the distributed samples have not been modified.

## Distributed Sample Solutions

All EXECs for Sample Solutions are distributed in the BBPROC data set allocated to the SYSPROC DD statement in the AutoOPERATOR subsystem. All Rules for Sample Solutions are distributed DISABLED in various members of the BBPARM data set.

## Naming Conventions for EXECs

The naming convention for the Sample Solutions EXECs uses the format

`gffnnn1`

where

**g** is the first letter of the group (for example, M for MVS)

**ff** is a 2-character abbreviation for the functional area within the group

<b>Code</b>	<b>Function</b>
AM	ACCESS METHODS*
ER	ERROR RECOVERY*
IN	INFO SYS/MANAGEMENT
J2	JES2
J3	JES3*
MN	MONITORS
PE	PERFORMANCE*
ST	STORAGE SUBSYSTEMS*
UT	UTILITIES
VT	VTAM/NCP

\* The EXECs for these functional areas are available as part of the MV Solutions for AutoOPERATOR product.

**nnn** is a sequence from 001 to 999

**l** is the language of the particular EXEC (for example, X for REXX or C for CLIST)

The following three categories of EXECs do not follow this naming convention:

- Operator-initiated EXECs

Operator-initiated EXECs are executed as a result of an operator command.

- Text-ID-driven EXECs

Text-ID-driven EXECs are executed as a result of a WTO or WTOR. The names for Text-ID-driven EXECs are the Text-IDs.

- SYSPROG Services EXECs

SYSPROG Services EXECs are the command interfaces to SYSPROG Services. When a Sample Solution requires more data to complete its analysis, the SYSPROG Services EXECs can be used to provide concise and pertinent data.

The Sample Solutions EXECs distributed with this release have been renamed and enhanced with the addition of the documentation box.

Figure 21-1 on page 21-4 shows an example of the documentation box.

**Figure 21-1 Example of a Documentation Box**

```

/*****
/* DOC GROUP(MVS) FUNC(MONITORS) CODE(MN) */
/* DOC DISP(YES) AUTHOR(B&B) */
/* DOC DESC(RESOLVE WTO BUF SHORTAGE) */
/*-----*/
/* NAME: */
/* IEE249I */
/* */
/* DESCRIPTION: */
/* RESOLVE THE WTO BUFFER SHORTAGE SITUATION */
/* */
/* SEE ALSO: */
/* IEA404A */
/* IEA405E */
/* */
/* INVOKED BY: */
/* IEA404A */
/* IEA405E */
/* */
/* INPUT PARAMETERS: */
/* NONE */
/* */
/* OUTPUT: */
/* NONE */
/* */
/* OUTBOARD CALLS: */
/* NONE */
/* */
/* EXTERNAL ROUTINES CALLED: */
/* MVS "K" OR "CONTROL" COMMANDS ARE ISSUED. */
/* NOTE: FOR THE "K" COMMANDS TO BE PROPERLY AUTHORIZED, THEY MUST BE */
/* ---- ISSUED FROM A NON-SUBSYSTEM CONSOLE. EDIT THE MEMBER, */
/* BBISSP00, IN YOUR BBPARM DATA SET AND ADD THE KEY WORD, */
/* CMDCON=1; THIS WILL DIRECT ALL COMMANDS WHICH DO NOT REQUIRE */
/* A RESPONSE TO CONSOLE ID 1. */
/* */
/* ALERTS ISSUED: */
/* NONE */
/* */
/* VARIABLES USED: */
/* NAME: DESCRIPTION: DEFAULT VALUE: */
/* MSG MESSAGE ID IEE249I */
/* P1 WORD 1 CUA OF CONSOLE OR SYSLOG */
/* P2 WORD 2 CONSOLE ID */
/* P3 WORD 3 COND= */
/* P4 WORD 4 AUTH= */
/* P5 WORD 5 NBUF= */
/* */
/* TEST SCRIPT INFO: */
/* GENERATE A WTO FROM A TEST EXEC TO TRIGGER A CALL TO EITHER OF */
/* THE DRIVER EXECs: IEA404A OR IEA405E */
/* */
/* RETURN CODES: */
/* NONE */
/* */
/* CHANGE LOG: */
/* 03-05-90 BY MDR: ADDED DOCUMENTATION BOX */

```

The information contained at the top of the documentation box is used by the AutoOPERATOR EXEC Management Application to display information about the EXECs in the SYSPROC libraries. The GROUP, FUNC, CODE, DISP, AUTHOR, and DESC are displayed on one line by the AutoOPERATOR EXEC Management Application (refer to “Managing EXECs Using the EXEC Management Application” in the *MAINVIEW AutoOPERATOR Basic Automation Guide* for a detailed explanation). The rest of the information in the documentation box is for self-documentation and easy reference.

A brief description of the title fields shown in Figure 21-1 on page 21-4 follows:

<b>GROUP</b>	group to which this EXEC belongs (for example, MVS, CICS, DB2, or IMS)
<b>FUNC</b>	functional category within the group
<b>CODE</b>	two-character function code
<b>DISP</b>	indicates whether information about this EXEC should be displayed under the AutoOPERATOR EXEC Management Application (refer to “Managing EXECs Using the EXEC Management Application” in the <i>MAINVIEW AutoOPERATOR Basic Automation Guide</i> ).
<b>AUTHOR</b>	author of the EXEC
<b>DESC</b>	25 character (or less) description of what this EXEC does
<b>NAME</b>	name of this EXEC
<b>DESCRIPTION</b>	detailed description of what this EXEC does
<b>SEE ALSO</b>	cross-reference listing
<b>SCHEDULED BY</b>	list of other EXECs that call this EXEC
<b>INPUT PARAMETERS</b>	any input parameters are required, this field indicates the parameter name, description, and default value
<b>OUTPUT</b>	information or names of variables which is passed back
<b>OUTBOARD CALLS</b>	indicates if any pager calls are placed
<b>EXTERNAL ROUTINES CALLED</b>	other routines or services that are called along with any special considerations for those calls

<b>ALERTS ISSUED</b>	brief text for long EXECs that issue ALERTs; otherwise, a YES or NO
<b>VARIABLES USED</b>	variable name, description, and default value if variables are used
<b>TEST SCRIPT INFO</b>	necessary environment and steps that are needed to test this EXEC if changes are required
<b>RETURN CODES</b>	shows what return codes (if any) are passed back
<b>CHANGE LOG</b>	chronology of changes

## Standards for Sample Solutions

Some national characters do not translate correctly when you go from EBCDIC format to ASCII format and then back to an EBCDIC format. In particular, the national characters #, !, %, and @ translate differently depending on the country in which the translation takes place. BMC Software recommends that you avoid using national characters in your naming conventions.

## Sample VTAM Solutions

### VTAM/NCP Solution

Much of the work in data centers is performed interactively using online systems and terminal networks. However, the commands and replies used to interface with the network software can be cumbersome to use.

BMC Software provides the following Sample VTAM/NCP Solution:

**VTAM Commands** Provides command interface for VTAM

**Warning!** With the Sample Solutions, BMC Software delivers sample source code that is to be used only as examples for solution implementation.

BMC Software Customer Support will not provide support for any Sample Solution which has been modified. Customers seeking technical support from BMC Software must be prepared to demonstrate that the distributed samples have not been modified.

## VTAM Commands Solution

The following sections describe the Sample VTAM Commands Solution.

### Setting Variables for the Sample VTAM Commands Solution

Activating and inactivating VTAM/NCP resources requires memorizing VTAM command formats. This Sample Solution provides command interfaces for enabling and disabling resources and alleviates the need to memorize command formats.

This Sample Solution does not require any values to be set for Sample Solution variables.

### Invoking the Sample VTAM Commands Solution

This Sample Solution is operator-invoked. To invoke this Sample Solution, enter

```
%NETACT P1 P2
```

on the **COMMAND** line to activate a resource where

<b>P1</b>	Resource name
<b>P2</b>	Scope (optional)

#### Inactivating a Resource

To inactivate a resource, enter

```
%NETINACT P1 P2
```

on the **COMMAND** line where

<b>P1</b>	Resource name
<b>P2</b>	Type

### Reactivating a Resource

To inactivate and then reactivate a resource, enter

**%NETBOUNC P1 P2 P3**

on the **COMMAND** line where

- P1** Resource name
- P2** Type for inactivate  
(default = immediate)
- P3** Scope for activate

### Processing Flow for the Sample VTAM Commands Solution

NETACT issues a **V NET,ACT** command for the resource specified. If specified by the operator, **SCOPE** is included in the command. NETINACT issues a **VARY NET,INACT** command for the resource specified. If specified by the operator, **TYPE** is included in the command.

NETBOUNC issues a **VARY NET,INACT** command for the resource specified. A type of **IMMEDIATE** is used in the command, unless the operator specifies **FORCE**. After waiting 10 seconds for the resource to become inactive, a **VARY NET,ACT** command is issued. **SCOPE** is included in the command if specified by the operator.

### EXECs for the Sample VTAM Commands Solution

The following list describes the EXECs for the Sample VTAM Commands Solution.

<b>NETACT</b>	Activates resources
<b>NETBOUNC</b>	Bounces resources
<b>NETINACT</b>	Inactivates resources

### Rules for the Sample VTAM Commands Solution

There are no Rules for the Sample VTAM Commands Solution.

# Sample Monitoring Solution

The Sample Monitoring Solution is designed to assist data center personnel monitor events occurring in the system. BMC Software provides the following Sample Monitoring Solution:

**Operations Monitors** Provides problem determination information

**Warning!** With the Sample Solutions, BMC Software delivers sample source code that is to be used only as examples for solution implementation.

BMC Software Customer Support will not provide support for any Sample Solution which has been modified. Customers seeking technical support from BMC Software must be prepared to demonstrate that the distributed samples have not been modified.

## Sample Operations Monitors Solution

This Sample Solution provides monitors to be used by the operator to gather detailed information about problem areas.

### Setting Variables for the Sample Operations Monitors Solution

This Sample does not require any values to be set for Sample Solution variables.

### Invoking the Sample Operations Monitors Solution

This Sample Solution is operator-invoked. It is initiated by positioning the cursor under certain fields on the AutoOPERATOR MVS STATUS panel and pressing **Enter**.

For example, positioning the cursor underneath the TSOCPU field invokes the TSOCPU monitor.

## Processing Flow for the Sample Operations Monitors Solution

The following sections describe the processing flow for the Sample Operations Monitors Solution.

### **CSAABV / CSABLO**

The SYSPROG CSSUM command is issued to determine which address spaces have the most extended CSA (CSAABV) or CSA (CSABLO) allocated. The top five users are reported, using the ALERT application.

### **TOTSIO**

The SYSPROG MDEV command is issued to monitor all DASD devices for 15 seconds. The ALERT facility reports the five volumes with the highest I/O rates. A follow-up EXEC (TIOTSIO2) is associated with the ALERTs.

TOTSIO2 issues the SYSPROG MDEV command to monitor the device specified in the ALERT for 15 seconds. ALERTs are issued to show which address spaces are using the volume and the percentage of utilization attributed to each address space.

### **TOTCPU / JOBCPU / TSOCPU / STCCPU**

The SYSPROG CPU command is issued to determine which address spaces are major users of the CPU. Each monitor reports on a different type of top user:

- TOTCPU - top overall CPU user
- JOBCPU - top batch job CPU user
- TSOCPU - top TSO CPU user
- STCCPU - top Started Task CPU user

Another ALERT is generated in addition to the one identifying the major CPU users. The additional ALERT asks the operator if the address space should be cancelled. If the operator confirms the cancel, the address space is cancelled.

### **WATCH**

The SYSPROG PROGRESS command is issued to determine current statistics for the job under which the cursor is positioned. The information is reported, using the ALERT facility.

**EXECs for the Sample Operations Monitors Solution**

The following list describes the EXECs for the Sample Operations Monitors Solution:

<b>\$CSAABV</b>	Determines top five users of extended CSA
<b>\$CSABLO</b>	Determines top five users of CSA
<b>\$JOBCPU</b>	Determines top batch job user of CPU
<b>\$STCCPU</b>	Determines top Started Task user of CPU
<b>\$TSOCPU</b>	Determines top TSO user of CPU
<b>\$TOTCPU</b>	Determines top user of CPU
<b>\$TOTSIO</b>	Determines top five busiest DASD devices
<b>\$TOTSIO2</b>	Determines which address spaces are accessing a particular device
<b>\$WATCH</b>	Provides job statistics
<b>MMN002C</b>	Issues message asking if the address space should be cancelled
<b>MMN003C</b>	Cancels address space

**Rules for the Sample Operations Monitors Solution**

There are no Rules for the Sample Operations Monitors Solution.

## Sample CICSplex Solution

During its operations, CICSplex issues a number of messages with various actions or severity levels. Some of these messages require immediate attention.

The Sample CICSplex Solution

- transforms these messages (listed in “Invoking the Sample CICSplex Solution”) into AutoOPERATOR ALERTs
- assigns a color to each message based on the message's ID and severity level of the CICSplex problem

Rules for BB\* type messages are processed first for address spaces other than CICSplex address spaces that issue these messages. Therefore, ALERTS created by these rules provide the visibility of problems in other areas (such as a CAS).

If you require that the Sample Solution applies only to the operations of CICSplex, you must modify the necessary BB\* Rules to include the appropriate logic to determine the job name issuing the message.

### Invoking the Sample CICSplex Solution

This Sample Solution is invoked by message-initiated Rules that match the following categories. The appropriate AutoOPERATOR ALERTs are created (with throttling) in the CICSplex ALERT queue.

Message ID Types	Description	Notes
BB*U	All unrecoverable messages	CICSplex processing terminates.  A red critical ALERT is created that includes the time, date, job name, message text, and an ALERT key of UNRECOVERABLE ERROR.
BB*S	All severe error messages.	CICSplex processing is suspended.  A red critical ALERT is created with time, data, job name, message text, and an ALERT key of SEVERE ERROR.
BBMZA800W	CAS(ssid) connect interface disabled.	A red critical alert is created with time, date, job name, and message text.
BBC*W	All warning messages related to the Communications Manager.	A problem might exist, but CICSplex processing continues. A yellow major ALERT is created with time, date, job name, message text, and an ALERT key of COMMUNICATIONS ERROR: INVESTIGATE.

Message ID Types	Description	Notes
BB*W	All warning messages.	Something may be wrong, but CICSplex processing continues. A yellow major ALERT is created with time, date, job name, message text, and an ALERT key of WARNING ERROR: INVESTIGATE.
BBMZA995A	CAS(SSID) reply "U" to wait or "C" to continue termination.	WTOR for termination. A pink major ALERT is created with time, date, job name, message text, and an ALERT key of WTOR FOR TERMINATION.
BBMZA002A	CAS(SSID) operator interface is ready.	A green informational ALERT is created with time, date, job name, message text, and an ALERT key of CAS OPERATOR INTERFACE READY.
BBMZA094A	CAS(SSID) reply "Y" to confirm cold start or "N" to terminate.	WTOR to confirm cold start as coordinating address space determined that SSID is currently or previously used. A pink major ALERT is created with time, date, job name, message text, and an ALERT key of COLD START WTOR OUTSTANDING.
EYU*U	All unrecoverable messages.	CICSplex processing is terminates. A red critical ALERT is created with time, date, job name, message text and an ALERT key of UNRECOVERABLE ERROR
EYU*S	All severe error messages.	CICSplex processing is suspended. A red critical ALERT is created with time, data, job name, message text and an ALERT key of SEVERE ERROR.
EYUC*W	All warning messages related to the Communications Manager.	Something may be wrong, but CICSplex processing continues. A yellow major ALERT is created with time, date, job name, message text and an ALERT key of COMMUNICATIONS ERROR: INVESTIGATE
EYUXL0205D	Parameter errors have occurred: reply "GO" or "CANCEL".	WTOR for handling parameter errors. A pink critical ALERT is created with time, date, job name, message text, and an ALERT key of WTOR OUTSTANDING.

## EXECs

There are no EXECs for this solution.

## Rules

One message Rule for each message or message category is listed in "Invoking the Sample CICSplex Solution" on page 21-12. The Rules are located in BBPARM member RULBCPSM.

## Sample Info/System Management Solution

The Sample Info/System Management Solution helps data center personnel add problem records to the INFO database.

**Warning!** With the Sample Solutions, BMC Software delivers sample source code that is to be used only as examples for solution implementation.

BMC Software Customer Support will not provide support for any Sample Solution which has been modified. Customers seeking technical support from BMC Software must be prepared to demonstrate that the distributed samples have not been modified.

Due to the requirements and site-specific implementations of Info/System Management, BMC Software does not provide ready-to-use Sample Solutions. Instead, BMC Software distributes a sample interface to the INFO database in the BBUSER data set; the member is OSPINFO1. OSPINFO0 was created using the OSPI application.

Refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide* for more information about the OSPI application.

The member OSPINFO1 was created from the base OSPINFO0. After the initial member was created, the input information was changed into keywords, a documentation box, and comments to explain the flow of the OSPINFO1 EXEC.

All applications require the use of a USERID and password. To eliminate hard-coded personal USERIDs and passwords, BMC Software recommends that data centers define a common USERID for use in logging on to INFO.

## Invoking the Sample Info/System Management Solution

When the Rule for IEF450I is triggered, a comparison is made against the job name contained in the text and the user specified job names. If a match is found, an ALERT is sent to inform the operator of the failure.

The flow of IEF450I continues by calling the OSPINFO1 EXEC to generate an INFO System Management record. Upon return from OSPINFO1, a VGET is issued for the variables SYSPROG and RECNUM.

## EXECs for the Sample Info/System Management Solution

The following list describes the EXECs for the Sample Info/System Management Solution:

<b>IEF450I</b>	Call OSPINFO1 based on job name
<b>OSPINFO1</b>	Write an Info/System Management Record

## Rules for the Sample Info/System Management Solution

The following list describes the Rule for Sample Info/System Management Solution:

<b>IEF450I</b>	Text-ID = IEF450I, distributed DISABLED
----------------	---

## Sample MVALARM Solution

The MVALARM solution enables you to convert alarms sent from MAINVIEW Alarm Manager to AutoOPERATOR ALERTs. The AutoOPERATOR ALERTs can be viewed with the ALERT Management Facility, which makes you aware that an alarm has been generated and sent to AutoOPERATOR for automation.

To ensure that only current, valid alarm information is displayed in the ALERT Management Facility, the ALERT's priority is upgraded every 10 minutes until it reaches CRITICAL. At CRITICAL, the ALERT Management Facility deletes the ALERT.

If the condition that created the alarm is resolved before the ALERT reaches CRITICAL, the ALERT Management Facility will delete the ALERT.

## Enabling the MVALARM Solution

The MVALARM Solution comprises two Rules in the AAORUL00 Rule Set: ALRMSTRT and ALRMSTOP.

To enable the Rules, follow these steps:

- Step 1** Invoke the Rule Processor application by typing **RULES** from any BBI terminal session screen.
- Step 2** Ensure that the AAORUL00 Rule Set is enabled. If not, enable the Rule Set by entering the (E)nable line command next to the Rule Set name.
- Step 3** Select the AAORUL00 Rule Set.
- Step 4** Enable the ALRMSTRT and ALRMSTOP Rules by

- enabling the Rule on the Rule Processor Detail Control panel.
- using the (E)nable line command on the Ruleset Overview panel.
- issuing the BBI command:

```
.T RULE,ENA,ALRMSTRT  
.T RULE,ENA,ALRMSTOP
```

Once the Rules are enabled, when MVALARM sends an alarm to AutoOPERATOR, the Rule ALRMSTRT fires and an ALERT is created. The ALERT contains the text of the alarm and is upgraded in priority every 10 minutes.

After the ALERT reaches the CRITICAL priority, it is deleted. If the situation that caused the alarm is resolved before this point, the Rule ALRMSTOP fires and deletes the ALERT.

## Stopping the MVALARM Solution

To inactivate the solution, disable the Rules by

- disabling the Rule on the Rule Processor Detail Control panel.
- using the (D)isable line command on the Ruleset Overview panel.
- issuing the BBI command:

```
.T RULE,DIS,ALRMSTRT  
.T RULE,DIS,ALRMSTOP
```

---

---

# Appendix A Diagnosing CSM Errors

The following sections describe

- how to create a cross-reference report

Occasionally, BMC Software Customer Support staff will ask you to create a cross-reference report to aid in diagnosing CSM problems. “Creating a Cross-Reference Report” describes how to do this.

- how to gather information about your system’s environment

These discussions include how to use the CSMDEBUG EXEC to trace CSM start and stop EXECs, and trace all CSM EXEC activity.

- common questions and answers

This section includes commonly asked questions and answers regarding how CSM works.

## Creating a Cross-Reference Report

Occasionally, to diagnose the source of a problem in CSM, BMC Software Customer Support might request a copy of your database. This is used to get a better picture of your environment and determine how the objects are related to one another.

To gather the requested information, you can create a CSM database cross-reference report. This reporting facility is built into CSM dialog. When completed, the output is routed to a sequential data set as well as, if requested, a sysout class for printing.

To generate a database cross-reference report, follow these steps:

**Step 1** Display the CSM panel (Figure A-1).

**Figure A-1 Global Overview (CSM Administration Panel): Generating a Cross-Reference Report**

```

BMC Software ----- Global Overview ----- MAINVIEW AutoOPERATOR
COMMAND ==>>                                     SCROLL ==>> CSR
                                                    TGT --- KMZ1

Primary Commands: EXCeption  Group      SORT      SAVE      ADD
                  CAleNDAR  CANcel    UNSort    CONVERT   Build
                  Locate    CMDSHOW  Xref      SAVERULE  SHUTSYS

LC CMDS ----- (S)tart  b(O)unce  rese(T)   depe(N)dents (A)dd
                  sto(P)   (EX)ec   (E)dit    c(L)ients   (D)delete
                  (CAN)cel (M)ove  (R)peat  (B)rowse   s(Y)nc

LC  Object  Group  System  Actual  Desired  Ind  Type
___  _____  _____  _____  _____  _____  _____  _____
___  MVS     KMZ1   SJSD    UP      UP      _____  NORM

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

**Step 2** Enter the Xref primary command.

Figure A-2 is displayed.

**Figure A-2 CSM Diagnostic Cross-Reference Panel**

```

BMC Software ----- CSM Diagnostic Cross-Reference ----- MAINVIEW AutoOPERATOR
COMMAND ==>>                                     TGT --- TB62

Generate cross-reference report:

Repository ==>> BMCTI.C720808.AOCT.BBIREPS_____
Group id    ==>> AOCT_____

The report will be written to:

Data Set Name ==>> 'BAOTYB.CSMXREF.CSMXGRP2'_____

Browse report ==>> Y (Y/N)
SYSOUT class ==>> _
    
```

**Step 3** If you would rather use a different data set name, overwrite the **Data Set Name** field. Remember to enclose the data set name in quotation marks.

If you want to have the cross-reference report printed on any JES printer, specify a valid output class in the **Sysout class for printed report** field.

**Step 4** Press **Enter** to generate the report.

If the **Browse report after generation** field is set to Y, the report data set is displayed and you can browse the report.

**Step 5** When you are finished browsing the report data set, press **PF3/END**.

**Step 6** Write the report data set to tape and send it or the printed report or both to BMC Software Customer Support.

## Gathering Environmental Information

The following section describes how to use the EXEC CSMDIAG to gather information about the BBI-SS where CSM is running.

### Generating CSM Exec Diagnostics for Support

In some cases, BMC Software Customer Support might request information that you can obtain only with traces. The CSMDEBUG EXEC is designed to provide REXX traces at two levels: EXEC or Global. These traces are produced in BBILOG.

To schedule the EXEC, enter from the BBI Journal

```
%CSMDEBUG L(level) A(action) P(I|R) T(trace)
```

These parameters are described in the following table.

Parameters	Possible Values	Description
L(level)	GLOBAL or EXEC	Specifying GLOBAL produces a trace of all the CSM EXECs.  Specifying EXEC produces a trace of that EXEC.  Specifying L (GLOBAL) might produce a lot of output, especially if you use it with the T (TRACE) parameter.
A(action)	SET or UNSET	Specifying SET activates the trace.  Specifying UNSET deactivates the trace.
P(I R)	I or R	REXX trace options.  You must use this parameter in conjunction with the T (TRACE) parameter.
T(trace)	TRACE DISPLAY QUIET	Specifying TRACE produces a REXX trace.  Specifying DISPLAY produces a trace of the EXECs that have been executed.  Specifying QUIET stops all trace activity at the level specified by the L (LEVEL) parameter.

## Tracing Starts and Stops of CSM EXECs

The CSMDEBUG EXEC can be used to detect the EXECs that were executed and to perform a comprehensive trace of these EXECs. Schedule the CSMDEBUG EXEC from the BBI Journal by entering

```
%CSMDEBUG L(GLOBAL) A(SET) T(DISPLAY)
```

You will receive (in BBLOG) a message for every time an EXEC starts (ENTRY) and a message each time an EXEC stops (EXIT). The messages are time-stamped in seconds and hundredths of a second so you can determine the exact sequence of the EXEC invocation.

You also can set this display for a specific EXEC (CSMACT); for example:

```
%CSMDEBUG L(CSMACT) A(SET) T(DISPLAY)
```

## Tracing All Activity in a CSM EXEC

If a problem is suspected in a specific CSM EXEC (for example, EXEC CSMACT), you can produce a trace quickly and easily to the BBI Journal by scheduling the CSMDEBUG EXEC as follows:

```
%CSMDEBUG L(CSMACT) A(SET) P(I) T(TRACE)
```

A REXX trace which traces all clauses before execution (as well as shows the intermediate results during evaluation of expressions and substituted names) is written to the BBI Journal. You might be asked to send the results of this trace to BMC Software to be examined by Customer Support staff.

**Note:** The trace output might produce many lines of output in the BBI Journal. Parameter MAXTPUT in BBPARM member AAOEXP00 controls how much data can be written to the BBI Journal, so you must make sure this parameter is set correctly to handle the generated output.

## Stopping a Trace

When the diagnostic data is generated at the Global level, the debug facility should be deactivated in the following manner:

```
%CSMDEBUG L(GLOBAL) A(SET) T(QUIET)
```

This command causes no diagnostic data to be written to the BBI Journal for any CSM EXEC.

When the diagnostic data is generated at the EXEC level (for example, the EXEC named CSMACT), the debug facility should be deactivated in the following manner:

```
%CSMDEBUG L(CSMACT) A(SET) T(QUIET)
```

This command causes no diagnostic data to be written to the BBI Journal for the EXEC specified (in this example, CSMACT).

Each time CSMDEBUG executes, it issues a BBI display variable command for CSM.DEBUG, which allows confirmation of trace deactivation. If the trace is active, the entry in the profile pool contains the value TRACE along with the REXX trace option. If it is inactive, the entry contains the value QUIET.

## Creating CSM Start and Stop Rules

There are two objects that you might need to create Start and Stop Rules for: APPC and TCPIP.

### Creating CSM Start and Stop Rules for the APPC Object

An example of an object for which you might create Start and Stop Rules is the object APPC. The CSM-generated Start and Stop Rules are created for APPC, but do not fire because these Rules have an IMFOJOB variable value equal to APPC as selection criteria and the IMFOJOB variable value for APPC is equal to CONSOLE.

The APPC object may be started and UP, but the actual state is never recognized by CSM as UP.

To define APPC as a CSM object and create the Start and Stop Rules, follow these steps:

- Step 1** On the Add NEW Object panel for APPC, specify **NO** in the **Generate the Rules** field.
- Step 2** Build a Start Rule:
- 2.A** On the Selection Criteria panel, specify **ATB007I** on the **Text Id** field.
  - 2.B** On the Variable Dependencies panel, specify **!CSM.ACTIVEFLAG EQ YES**.
  - 2.C** On the Automation Specification panel, specify **YES** for the **Display at dest.** field, and specify **CSMUP CSMUP APPC** on the **EXEC Name/Parms** field.
- Make sure that the Rule resides in a Rule Set other than AAORULCM and AAORULBC.
- Step 3** Build a Stop Rule:
- 3.A** On the Selection Criteria panel, specify **ATB002I** on the **Text Id** field
  - 3.B** On the Variable Dependencies panel, specify **!CSM.ACTIVEFLAG EQ YES**

- 3.C** On the Automation Specification panel, specify **YES** for the **Display at dest.** field, and specify **CSMDOWN CSMDOWN APPC** on the **EXEC Name/Parms** field

Make sure that the Rule resides in a Rule Set other than AAORULCM and AAORULBC.

Note that, any time you create a Start Rule and Stop Rule, you must not attempt to add them to Rule Sets AAORULCM and AAORULBC. The Rules can be added to any other Rule Set.

## Creating CSM Start and Stop Rules for the TCPIP Object

Another example of an object for which you might create Start and Stop Rules is the object TCPIP. The CSM-generated Start and Stop Rules are created for TCPIP, but do not fire because these Rules have an IMFOJOB variable value equal to TCPIP as selection criteria and the IMFOJOB variable value for TCPIP is equal to CONSOLE.

The TCPIP object may be started and UP, but the actual state is never recognized by CSM as UP.

To define TCPIP as a CSM object and create the Start and Stop Rules, follow these steps:

- Step 1** On the Add NEW Object panel for APPC, specify **NO** in the **Generate the Rules** field.
- Step 2** Build a Start Rule:
- 2.A** On the Selection Criteria panel, specify **EZB6473I** on the **Text Id** field.
- 2.B** On the Variable Dependencies panel, specify **!CSM.ACTIVEFLAG EQ YES**.
- 2.C** On the Automation Specification panel, specify **YES** for the **Display at dest.** field, and specify **CSMUP CSMUP TCPIP** on the **EXEC Name/Parms** field.

Make sure that the Rule resides in a Rule Set other than AAORULCM and AAORULBC.

- Step 3** Build a Stop Rule:

- 3.A** On the Selection Criteria panel, specify **IEF404I** on the **Text Id** field.
- 3.B** On the Variable Dependencies panel, specify **!CSM.ACTIVEFLAG EQ YES**.
- 3.C** On the Automation Specification panel, specify **YES** for the **Display at dest.** field, and specify **CSMDOWN CSMDOWN TCPIP** on the **EXEC Name/Parms** field.

Make sure that the Rule resides in a Rule Set other than AAORULCM and AAORULBC.

Note that, any time you create a Start Rule and Stop Rule, you must not attempt to add them to Rule Sets AAORULCM and AAORULBC. The Rules can be added to any other Rule Set.

## Common Questions and Answers

This section contains, in a “question and answer” format, various discussions about the use of CSM. The discussions cover these topics:

- errors discovered during the starting and stopping of objects
- errors discovered during the restarting of objects
- defining special objects
- CSMACT usage
- terminal sessions
- TSO/E or environmental problems
- installation problems
- implementation problems
- other tips and hints

## Starting/Stopping Errors

**Question:** A Normal type object has been *started* and is running. Its STATE variable is set to STARTING and its DESIRED variable is set to UP. Why is the state STARTING and not UP status?

**Answer:** Check the Start Rule for that object in Rule Set AAORULCM. It probably has not fired. CSMUP is the EXEC the Start Rule schedules and the CSMUP EXEC sets the object’s STATE variable to UP. All you need to do is correct the Rule so that it fires when the object starts.

**Question:** I am trying to define *CA-TOP SECRET* to CSM. The object has been defined and its Rule has been built. It starts at the scheduled time and the object comes UP. The STATE variable is set to STARTING and the Rule has not fired. In addition, an ALERT has been issued indicating a START command timeout. Why?

**Answer:** Unlike most other objects, CSM must always issue a MVS **D A, TOP** command for the CA-TOP SECRET object. To create an object for CA-TOP SECRET correctly, put the Started Task for CA-TOP SECRET in a member called TOP and name the procedure in this member TOP.

This ensures that all commands issued for this object are for TOP. The IMFOJOB variable (which is used throughout CSM) is set to TOP and the dynamically generated Rule is for TOP. This causes the Rule to fire properly, the object's state will be UP, and the timeout does not occur.

**Question:** An object is *stuck* with its state as STARTING. I know it is UP and have corrected the problem in either the Rule or the object definition. How do I get the state changed to UP?

**Answer:** From the BBI Journal, schedule the EXEC CSMUP for the object as follows:

```
%CSMUP objectname
```

This changes the object state variable STATE variable to UP.

**Question:** An object is *stuck* with its state as STARTING, but I know the object has terminated. In addition, the object terminated before the ASOD variable was created by the START event. How do I get CSM to recognize the object as terminated?

**Answer:** When an object has a Start command specified in its definition, CSMEOM automatically recognizes any end-of-memory events that occur while the object is still in STARTING status.

However, if an object has a Start EXEC specified in its definition, you must make the following modifications to ensure that CSMEOM recognizes end-of-memory events that occur while the object is still in STARTING status.

When you are using your own Start object EXEC, insert following the Start command. For REXX EXECS:

```
"IMFEXEC SELECT EXEC(CSMASID "IMFCASID" "OBJECT" )  
WAIT(YES) "
```

For CLIST EXECs:

```
IMFEXEC SELECT EXEC(CSMASID &IMFCASID &OBJECT) WAIT(YES)
```

**Note:** When passing special characters in the parameter list of IMFEXEC SELECT EXEC, you need to enclose the entire parameter list in single quotation marks, for example:

---

```
IMFEXEC SELECT EXEC('STRTCHIN TQV4MSTR EARLYUP (CSQY022I  
TQV4MQM Q MANAGER INITIALIZATION DONE')
```

---

In this example, the extra *right parenthesis* might interfere with parsing of the parameter list causing you to receive unexpected results.

Before calling CSMASID, ensure that the object variable is set correctly and that it contains the name of the object that was started. The IMFCASID variable is set by the object's Start command. CSMASID returns a value of 8 in IMFRC, if it was not successful, and the messages accompanying the return code describe the error.

## Restart-Only Objects

**Question:** How does *Restart Only Control = Y* work?

**Answer:** When an object is defined to CSM with the **Restart Only Control** field set to Y (yes), CSM does not start it automatically during IPL. The object is started and stopped externally by other means (console command, scheduler, and so on).

Once it has been started the first time, the Restart Only Control=YES setting means CSM will control restarts for this object only if the object terminates unscheduled.

## Defining Special Objects

**Question:** I am trying to stop VTAM with CSM. I use message *IST020I* as my stop message ID but the Rule never fires and the object's STATE variable is set to STOPPING. Why?

**Answer:** The Rule never fires because MVS and VTAM do not supply a job name for message IST020I. Since job name is part of the selection criteria for the Rule, the Rule will never fire.

The best way around this problem (other than using \$HASP395 or IEF404I as the Stop Text-ID) is to replace the Stop command in the object definition with a short EXEC. The EXEC needs to do two things: first: it must issue the Stop command for the object with an IMFEXEC CMD statement; second, it must issue an IMFEXEC MSG statement to send the following message to the BBI Journal:

```
ACM902I objname
```

where objname is the object to be stopped.

Make sure an invalid Stop Text-ID (one that causes the Rule to never fire) is coded on the object definition panel. The ACM902I message in the Journal causes CSMDOWN to fire, using the correct object name, which ensures that the CSM variables are set properly.

**Question:** What is the best way to define *DB2* as an object to CSM. Should all three DB2 address spaces (MSTR, DBM1, and IRLM) be defined separately to the CSM database?

**Answer:** When defining DB2 as an object to CSM, define only the DB2MSTR address space. By design, when the DB2MSTR address space is started (and successful initialization is confirmed), DB2DBM1 and DB2IRLM are started automatically. Whenever DB2MSTR is stopped (by way of the STOP DB2 command) DB2DBM1 and DB2IRLM also are automatically stopped.

If any of the three DB2 address spaces abnormally terminates, the resolution procedure requires that all three DB2 address spaces are taken down, with the STOP DB2 command, and then restarted, with the START DB2 command.

Therefore, DB2MSTR should be defined as a normal, client/server object. Since DB2MSTR can provide database services for either CICS or IMS, DB2MSTR can be a server of either CICS or IMS in your system.

The start command for DB2MSTR is

```
#xSTART DB2
```

where *x* is the subsystem character for DB2.

The stop command for DB2MSTR is

```
#xSTOP DB2
```

Select the Start and Stop Text-IDs for DB2MSTR using the same guidelines as you would for any other object to be defined to CSM.

The message DSN9022I is issued to indicate that the DB2TMSTR object has successfully started and ended. DB2TMSTR successfully starts but it will never be stopped by CSM. So for CSM to properly manage the object, you need to create special Rules to start and stop DB2TMSTR objects.

You are advised to define a Rule in a Rule Set other than AAORULBC or AAORULCM that triggers when the IEF450I message is issued by DB2MSTR, DB2DBM1, or DB2IRLM. With this Rule in place, all three components of DB2 are recycled properly (and CSM variables are updated properly) in the event that any DB2 component fails.

### To Define DB2TMSTR As a CSM Object and Create the Start and Stop Rules

- Step 1** On the **Add New Object** panel for **DB2TMSTR**, specify **NO** in the **Generate the Rules** field.
- Step 2** Build a Start Rule:
- 2.A** On the Selection Criteria panel, specify **DSN9022I** on the **Text ID** field.
- 2.B** On the Variable Dependencies panel, specify
- CSM.ACTIVEFLAG** \_\_\_\_\_ **EQ** **YES** \_\_\_\_\_ and
- WORD2** \_\_\_\_\_ **EQ** **-DB2T** \_\_\_\_\_ and
- WORD3** \_\_\_\_\_ **EQ** **DSNYASCP** \_\_\_\_\_ and
- WORD4** \_\_\_\_\_ **EQ** **\*START** \_\_\_\_\_
- 2.C** On the Automation Specification panel, specify **YES** for **Display at Dest** field, and specify **CSMUP CSMUP DB2TMSTR** on the **EXEC Name/Parm** field.
- Step 3** Verify that the Rule resides in a Rule Set other than AAORULCM and AAORULBC.

**Step 4** Build a Stop Rule:

**4.A** On the Selection Criteria panel, specify **DSN9022I** in the **Text ID** field.

**4.B** On the Variable Dependencies panel, specify

**CSM.ACTIVEFLAG**\_\_\_\_\_ **EQ** **YES**\_\_\_\_\_ and

**WORD2**\_\_\_\_\_ **EQ** **-DB2T**\_\_\_\_\_ and

**WORD3**\_\_\_\_\_ **EQ** **DSNYASCP**\_\_\_\_ and

**WORD4**\_\_\_\_\_ **EQ** **\*STOP**\_\_\_\_\_

**4.C** On the Automation Specification panel, specify **YES** for **Display at Dest** field, and specify **CSMUP CSMUP DB2TMSTR** on the **EXEC Name/Parm** field.

**Step 5** Verify that the Rule resides in a Rule Set other than AAORULCM and AAORULBC.

**Note:** When creating a Start or Stop Rule, verify that you do not add them to Rule Sets AAORULCM and AAORULBC. The Rules can be added to any other Rule Set.

**Question:** Why doesn't CSM recognize the start and stop of the APPC object?

**Answer:** If the Start and Stop Rules have an IMFOJOB variable value equal to APPC as the selection criteria, and if the IMFOJOB variable value for APPC is equal to CONSOLE, the Rules do not fire.

There are two possible ways to define APPC to CSM.

- Define the APPC object to have a start and stop EXEC, and do not use a command to start and stop the APPC object. The start EXEC can be coded as:

---

```

/* REXX */
ADDRESS IMFEXEC
"IMFEXEC CMD XXXX"           /* XXXX IS THE START COMMAND */
"IMFEXEC MSG 'ACM901I YYYY'" /* YYYY IS THE OBJECT NAME */
"IMFEXEC EXIT CODE(0)"
EXIT

```

The stop EXEC can be coded as:

```

/* REXX */
ADDRESS IMFEXEC
"IMFEXEC CMD XXXX"           /* XXXX IS THE STOP COMMAND */
"IMFEXEC MSG 'ACM902I YYYY'" /* YYYY IS THE OBJECT NAME */
"IMFEXEC EXIT CODE(0)"
EXIT

```

---

The Text-ID ACM901I will cause the CSMUP EXEC to fire for object YYYY and Text-ID ACM902I will cause the CSMDOWN EXEC to fire. Rules for these Text-IDs are in the CSM Rule Set AAORULBC.

- If you want to create your own Rules to Start and Stop the APPC object, refer to “Creating Start and Stop Rules for CSM Objects” on page 2-15.

## CSMACT Usage

**Question:** How do I *return control* of an object’s schedule to CSM?

**Answer:** The RESET command can be used to return control of a parent and its dependents to CSM control.

When using the CSM panel, use the **rese(T)** line command and be sure to specify a scope of **c** for children. From the console or EXCP/VTAM sessions, schedule the CSMACT EXEC as follows:

**CSMACT object RESET DEP**

where *object* is the name of the CSM controlled object.

**Question:** I have tried to stop the MVS object when trying to bring the system down and have received this *ALERT message*:

```
ACM705A Please verify DOWN request for MVS
```

I know this means that a dependent of MVS has Verify Force Down set to YES, but is there a way around this ALERT?

**Answer:** If you shut down the MVS object using CSMACT in the following manner:

**CSMACT MVS MAEARLY DEP X**

the Verify Force Down option for the MVS dependent is ignored for this execution of CSMACT. ALERT ACM705A is not issued and all of the dependents of MVS will (assuming they have been properly defined) come down.

**Question:** I am trying to execute CSMACT by scheduling it *through a Rule*. The Rule fires and CSMACT is scheduled. The problem I have is that CSMACT appears to have not affected the object specified.

The following parameters were entered:

**CSMACT objectname UPEARLY NOREQ**

**Answer:** CSMACT expects the first parameter passed to it to be the EXEC name. When scheduled by a Rule, CSMACT does not get the EXEC name as the first parameter because of the difference in EXEC scheduling between the Rules Processor and other EXEC submission vehicles. See the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs* for more information.

To make CSMACT work properly when scheduled through the automation action panel, enter its parameters:

**CSMACT CSMACT objectname UPEARLY NOREQ**

**Question:** Explain how the *REQ/NOREQ* and *DEP/NODEP* parameters should be used when CSMACT is scheduled manually.

**Answer:** Use REQ|NOREQ whenever an UP command needs to be processed (for example, UPEARLY, UP, and so on).

Specifying REQ causes the parents of the object named in the CSMACT parameter list to be checked and, if necessary, brought up before the object named is started. NOREQ causes the check not to be performed.

Use DEP|NODEP whenever a DOWN command is to be processed (for example, DOWNEARLY, DOWN, and so on). Specifying DEP causes the children of the object named in the CSMACT parameter list to be checked and, if necessary, stopped before the object named is stopped. NODEP causes the check not to be performed.

Either DEP|NODEP or REQ|NOREQ can be used with a RESET or RESETC command. If DEP is used with a RESET|RESETC command, the named object, along with its children, is affected. If REQ is used with a RESET|RESETC command, the named object, along with its parents, is affected.

If neither DEP|NODEP nor REQ|NOREQ is specified in the CSMACT parameter list, only the named object is affected.

## BBI Terminal Sessions (BBI-TS)

**Question:** Why are the *CSM dialog panels* not displayed when I select it? No error message is generated. Why?

**Answer:** Check the BBITSP00 member in BBPROF. If ISPF=NO is specified, the terminal session (even though ISPF is engaged in the TSO user address space) runs in TSO mode.

The terminal session must be in ISPF mode for the CSM dialogs to function. Set ISPF=YES in the BBITSP00 member.

## TSO/E or Environmental Problems

**Question:** When I attempt to initialize CSM, CSMBUILD issues message:

```
ACM203E ERROR - Error allocating &CSM.PARMDATASET. RC-  
>8.
```

What do I do?

**Answer:** If the ALLOCATE command processor program is moved from SYS1.COMDLIB to SYS1.LPALIB, this error occurs. To correct the problem, either move the ALLOCATE module back to SYS1.COMDLIB or add a STEPLIB to the BBI-SS for SYS1.LPALIB.

## CSM Installation

**Question:** I have just installed AutoOPERATOR 3.1. Are there any steps related to CSM in *AutoCustomization*?

**Answer:** No. There are no steps related to CSM in the AutoCustomization procedure. If you have successfully installed AutoOPERATOR 3.1, the components of CSM should be ready for use.

**Question:** How can I verify CSM is *installed* correctly?

**Answer:** After installing AutoOPERATOR 3.1, refer to the *MAINVIEW AutoOPERATOR Customization Guide* for more information about verifying installation.

**Question:** I am *converting* to CSM version 3.1 from CSM version 1.1. When I start CSM version 3.1 for the first time, a panel is displayed that indicates that the format of the CSM database will be changed.

The BBPARM data set and the target subsystem specified are from a previous release. Where is this information coming from?

**Answer:** EXEC CSMSPF01 is taking the target ID and the BBPARM data set name from variables in the ISPF profile pool by way of an IMFEXEC VGET.

To make sure CSM uses the correct target ID and BBPARM data set when the database is saved, reply GO to the conversion panel. When CSM displays the object list, issue the PROFILE primary command. At this point, the target subsystem and BBPARM data set used by CSM can be changed.

## Effective Implementation

**Question:** I regularly shut down an object (for example, CICS) at a scheduled period each day. Batch maintenance processing for that object takes place during that shutdown period (for example, file backups or database reorganizations). I want to have the object *automatically start back up* when the maintenance processing is complete. What's the best way to do this?

**Answer:** Add a step to the end of the *batch maintenance* job to run IMFSUBEX. Use this additional step to schedule CSMACT for the object.

See the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs* for information on IMFSUBEX usage.

Schedule the CSMACT EXEC by entering

**CSMACT object\_name UPEARLY NOREQ**

Make sure you specify UPEARLY and not UP.

**Question:** I need to bypass AO/CSM control of the *system startup* on occasion. How can I accomplish this?

**Answer:** CSM startup is triggered by message PM0010I. When that message is issued, it schedules the EXEC ACMM000. ACMM000 calls CSMINIT and so on.

To bypass AO/CSM control of the system, you could write a small EXEC that based on the reply to a WTOR, schedules or does not schedule ACMM000 with an IMFEXEC SELECT EXEC statement; for example:

```

/* REXX */

"IMFEXEC WTOR 'DO YOU WANT CSM TO START THE SYSTEM?  REPLY "Y" TO",
"START.' WAIT(60) REPLY(REP)"
IF IMFCC = 8 THEN DO
  "IMFEXEC WTO 'TIME HAS BEEN EXCEEDED.  CSM WILL NOT START SYSTEM.'"
  "IMFEXEC EXIT CODE(0)"
  EXIT
END

IF REP = 'Y' THEN DO
  "IMFEXEC SELECT EXEC (ACMM000)"
END
ELSE DO
  "IMFEXEC WTO 'CSM WILL NOT START SYSTEM.'"
END

"IMFEXEC EXIT CODE(0)"
EXIT

```

After writing this EXEC (or something similar), change Rule ID ACMRC005 in Rule Set AAORULBC to schedule this new EXEC instead of ACMM000.

Remember, anytime a PTF performs maintenance against AAORULBC, Rule ID ACMRC005 must be changed to call the EXEC, instead of ACMM000.

## Other Tips and Questions

**Question:** Instead of simply restarting a CSM managed object following an abend, is there some way to have a *recovery procedure* executed prior to restarting the object?

**Answer:** CSM does not distinguish between a normal and abnormal termination for an object. However, it is possible to implement a recovery process with an EXEC that performs the start in conjunction with additional Rules or EXECs.

One approach is to create a Rule for message IEF450I to capture the abend/termination status and store it in a variable in the shared pool called USER.object.TERMTYPE.

The start EXEC for the object would use this variable to determine whether or not the recovery procedure is needed. Following an abend, the start EXEC initiates the appropriate recovery process, after which the failed object is started. But following a normal termination for the object, the start EXEC would simply start the object.

**Question:** I cannot *access a terminal session* and I need to perform a SYNC. Can I perform this from a console?

**Answer:** Yes. To issue a SYNC, two EXECs must run one after the other. First, execute CSMSTAT. After CSMSTAT is complete, execute CSMSCAN. Neither EXEC requires input parameters.

**Question:** How can I deactivate CSM?

**Answer:** Schedule EXEC CSMDEL. It will set the CSM.ACTIVEFLAG variable to NO, which will prevent CSM processing from taking place.

**Question:** I used to see CSM messages being written to the BBI Journal (for example, ACM001I, ACM003I, ACM104I, ACM986I, and ACM987I). Now, I do not see them anymore. Why?

**Answer:** Rules in the Rule Set member AAORULBC intercept ACM messages written to the journal. If the AUDIT parameter in BBIPARM member BBISSP00 is set to YES, the messages are written to the BBI Journal. If the parameter is set to NO, the messages are not written to the Journal (the only exception to this Rule is message ACM488I).

**Question:** When is the Start Command limit reset?

**Answer:**

- At midnight each day.
- By invoking the CSMACT EXEC with command RESETC for the object.
- By successfully starting the object and its actual state being set to UP.
- At IPL.

**Question:** How do I start or stop an object from an operator's *console* and still keep the object under CSM control?

**Answer:** Build two Rules in a Rule Set other than AAORULBC or AAORULCM. For one, set the selection criteria to

```
Text-id = P
Event type = CMD
Variables = &IMFOJOB/ NE &QSSNAME/ AND !CSM.WORD2.STATE
NE ''
```

Set the action to

```
Reject Cmd. = Y
EXEC = CSMACT . &WORD2 DOWNEARLY NODEP
```

For the other, set the selection criteria to

```
Text-id = S
Event type = CMD
Variables = &IMFOJOB/ NE &QSSNAME/ AND !CSM.WORD2.STATE
NE ''
```

Set the action to

```
Reject Cmd. = Y
EXEC = CSMACT . &WORD2 UPEARLY NOREQ
```

This allows the object to be either started or stopped from the console with the normal Start or Stop commands, but still allows CSM to function properly.

If DB2 is an object in the CSM database, the Rule should be modified to not fire for START DB2 commands. DB2 requires that the START command for the internal jobs (MSTR,DBM1) are issued internally and not issued by CSM.

To modify the Rule, change

```
VARIABLES=&IMFOJOB/ NE &QSSNAME/ AND !CSM.WORD2.STATE NE "
```

to

```
VARIABLES=&IMFOJOB/ NE &QSSNAME/ AND !CSM.WORD2.STATE NE "
AND WORD2 NE DB2
```

---

## Appendix B Sample EXECs

This appendix introduces the JESDOWN EXEC and describes where you can find more information about the @TIMER EXEC.

### @TIMER

For more information about the @TIMER EXEC, refer to the chapter on “Using Utility EXECs Supplied by MAINVIEW” in the *MAINVIEW AutoOPERATOR Advanced Automation Guide*.

### JESDOWN

The JESDOWN EXEC is located in BBPROC. It is invoked by the operator to shutdown JES2. When invoked, a WTOR message is displayed. The operator must respond with Y to continue with the shutdown or N to cancel processing. If Y is entered, the following steps are taken:

1. A \$PI command is issued to drain all of the initiators.
2. A \$P command is issued to stop all new JES requests and reset current activities.
3. Commands are issued to display and then to stop all remotes.
4. Commands are issued to display and then to stop all lines.
5. Commands are issued to display and then to stop all active batch jobs.
6. Commands are issued to display and then to stop all active printers.

7. A IMFEXEC WAIT command is issued.
8. A WRITELOG command is issued.
9. A \$PJES2 command is issued next.
10. If the operator gets a response that the system is not dormant, a WTOR message is displayed giving the operator the option to force a shutdown.
11. If the operator replies with a Y to force the shutdown, a \$PJES2, ABEND command is issued.
12. If JES2 ends successfully, a WTO message “*JES2 shutdown completed successfully*” is displayed. If JES2 does not end, a WTO message “*JES2 is still draining, please initiate manual procedures*” is displayed.

The JESWAIT Profile variable can be customized. It is located in the MSU002C EXEC.

---

# Appendix C Sample REXX EXECs

REXX has many built-in functions, data queues, and stacks that make it a powerful interpretive language. The BMC Software sample REXX EXECs use some of the features of the REXX Language. The REXX samples, which have been distributed, demonstrate these features. For a complete discussion of the constructs and the syntax of the REXX language, refer to the IBM manuals for the version of REXX installed in your environment.

The following sample EXECs are the basis for the comparison and discussion of the REXX procedural language. Although further customization and changes are *required* to fully utilize these EXECs in a production environment, they are intended to illustrate some of the features of the REXX language.

These sample EXECs for REXX are distributed in the BBPROC data set. BMC Software provides the following REXX Sample EXECs.

## **REXX Compound Variable Initialization**

Set values for variables using the REXX STEMS

## **REXX DASD Reserve Analysis**

Provides immediate insight into contention problems in a shared DASD environment

## **REXX QUEUE/PULL And VPUT/VGET Usage**

Use VPUT/VGET and QUEUE/PULL instructions to store and retrieve data

## REXX Compound Variable Initialization EXECs

Two sample EXECs, MSU006X and MSU006X2, are distributed to show initialization of REXX compound variables.

In CLISTs, compound variables are defined with ampersands (&) and the REXX equivalent of ampersands is the STEM variable. The REXX compound variables are defined with STEM where the STEM variable is of the form VAR.*n* and *n* is an integer.

The following instructions reference values for SYS1 and SYS2 within a CLIST:

---

```
IMFEXEC VGET (SYSN) PROFILE
SET &N = 1           /* INITIALIZE THE INDEX VALUE      */
DO WHILE &N LE &SYSN /* SET UP THE LOOP                  */
IMFEXEC VGET (SYS&N) PROFILE
SET &SS = &&SYS&N    /* DERIVE THE VALUE                 */
```

---

The following set of instructions refer to the same values in a REXX EXEC:

---

```
ADDRESS IMFEXEC      /* SET IMFEXEC AS THE EXTERNAL ENVIRON */
IMFEXEC VGET (SYSN) PROFILE
DO N=1 TO SYSN BY 1 UNTIL N>SYSN
IMFEXEC VGET ('SYS'N) PROFILE
SS = VALUE('SYS'N)  /* DERIVE THE VALUE                   */
```

---

Notice that with CLISTs, the value of a compound variable must be derived using the && expression, whereas REXX uses the built-in function VALUE.

The apostrophes around SYS define it as a literal to REXX and distinguish the literal from the variable N. If a STEM variable was defined at initialization, this example could have used the following instruction to get the values for SYS.N:

```
IMFEXEC VGET (SYS.N) PROFILE
```

With this instruction, there are no apostrophes to distinguish SYS as a literal because 'SYS.' is a STEM and REXX defaults it to a literal.

In Figure C-1 on page C-3, note that there is no `SET` command in REXX. Any expression with an equal sign (=) implies an `ASSIGNMENT` command that functions the same as a CLIST `SET` command. `MSU006X` initializes the `SYSN` variables for reference by both CLIST and REXX EXECs. `MSU006X` is a REXX EXEC because the `STEM` variable, `SYS.1`, is not a valid expression within a CLIST.

`MSU006X` illustrates the assignment of values to both `SYS1` and `SYS.1` variables but you must code the values within the EXEC. `MSU006X2` illustrates the assignment of values to the `STEM` variables `SYS.n` by deriving the values of `SYS&n` within the context of MAINVIEW AutoOPERATOR initialization.

**Figure C-1      REXX EXEC - MSU006X**

---

```

/*-----*/
SYS1="NONE"           /* TARGET SYSTEM(NAME#1) FOR CLIST REFS */
SYS2="NONE"           /* TARGET SYSTEM(NAME#2) FOR CLIST REFS */

SYS.1="NONE"          /* TARGET SYSTEM(NAME#1) FOR REXX REFS */
SYS.2="NONE"          /* TARGET SYSTEM(NAME#2) FOR REXX REFS */
/*-----*/

```

---

At MAINVIEW AutoOPERATOR initialization, the values for `SYS&n` were specified to match the number of target systems within your environment. To build the REXX `STEM` variables, you can either change the whole start up EXEC, `MSU002C` or use `MSU006X2`, which dynamically assigns the values.

### EXECs for REXX Compound Variable Initialization

- |                 |   |
|-----------------|---|
| <b>MSU006X</b>  | Assigns values for <code>SYS1</code> , <code>SYS2</code> , <code>SYS.1</code> and <code>SYS.2</code> statically                             |
| <b>MSU006X2</b> | Assigns values for <code>SYS.1</code> and <code>SYS.2</code> dynamically by deriving the values for <code>SYS1</code> and <code>SYS2</code> |

## REXX DASD Reserve Analysis EXEC

The sample REXX DASD Reserve Analysis EXEC is modeled after the CLIST version of the same EXEC. This discussion concentrates on the use of REXX functions and instructions. For the complete discussion of this EXEC, see “DASD Reserve Analysis Solution” on page 9-4.

For the EXEC to function properly in your environment, you should establish values for the following variables.

<b>Variable</b>	<b>Description</b>
<b>SYSN</b>	Number of MVS systems in the shared DASD configuration
<b>SYS1-SYSn</b>	Names (for example, SMFID) of each system in a shared DASD configuration

This EXEC is initiated by a Rule for the following IOS message:

```
IOS071I  ucb,cbpid,jobname,start pending.
```

When REXX EXECs are started based on a `TEXT ID, MAINVIEW AutoOPERATOR` does not parse the message text prior to scheduling the specified EXEC. The REXX EXEC must do the parsing using the REXX instruction, `PARSE`.

The following instruction parses the IOS071I message in the distributed EXEC, IOS071IX.

```
PARSE ARG IOS071IX P1 ',' P2 ',' P3 ',' P4 P5 .
```

The `PARSE` instruction has other keyword operands besides `ARG`.

You should consult the *TSO Extensions Version 2 Reference* for a full discussion of the `PARSE` instruction. In this EXEC, the `ARG` keyword indicates that parsing should be done on the passed parameter strings.

The parameter string contains commas and blanks; in REXX EXECs, the EXEC must specify the parsing pattern. This example has commas in the pattern to subdivide the message. The following values would be placed in the variables:

Variable	Value
<b>IOS071IX</b>	IOS071I
<b>P1</b>	ucb
<b>P2</b>	chpid
<b>P3</b>	jobname
<b>P4</b>	start
<b>P5</b>	pending

Because this discussion does not go beyond the P5 variable, the EXEC has a period (.) as the last variable.

IOS071IX uses the ADDRESS command to inform REXX that non-REXX language instructions should be passed on to the external environment. After a VGET is completed to derive the value for SYSN (the number of shared systems), IOS071IX goes into the following loop:

---

```

DO N=1 TO SYSN BY 1 UNTIL N>SYSN      /* LOOK AT ALL SYSTEMS */
  IMFEXEC VGET ('SYS'N) PROFILE        /* START GETTING SS IDS*/
  TARG = VALUE('SYS'N)                /* SET TARGET SYS NAME */
  IF TARG \= IMFORGSS THEN DO         /* DON'T PROCESS THIS */
    ADDRESS IMFEXEC
    IMFEXEC SELECT,                  /* COMMAS ARE USED FOR */
                                     /* CONTINUATION IN REXX*/
    "EXEC(MPE001X" IMFORGSS P1 P3 ") TARGET("TARG")"
  END
END
END

```

---

Since REXX does not permit the use of ampersands, this example has apostrophes to distinguish a literal from a variable when the VGET was issued for the SS IDs. Remember that SYSN is the total number of subsystems in a shared DASD complex and 'SYS'N is the subsystem ID. The built-in function VALUE is used to assign TARG; if this value function was not used, the following IF clause would fail every time:

```
IF 'SYS'N = IMFORGSS THEN DO
```

The next instruction could have been used successfully in the `IF` construction:

```
IF IMFORGSS = VALUE('SYS'N) THEN DO
```

Because follow-up EXEC MPE001X needs to be scheduled on the TARGET system, it is easier to use the `ASSIGNMENT` clause to set `TARG`. Note that when MPE001X is scheduled, double quotation marks ( " ") are used to distinguish the literals from the variables which are then passed to the `IMFEXEC` environment. If double quotation marks were not used, REXX would have interpreted `EXEC(MPE001X)` as a built-in function.

MPE001X is the REXX counterpart to MPE001C. The MPE001X REXX EXEC uses the `REXX PARSE` and `ADDRESS` commands along with the `VALUE` built-in function. MPE001X issues a `VGET` for `IMFRC` after the call to the `RRES` utility and then MPE001X `VGETs` the number of lines passed back from `RRES` utility. For this discussion, the following command was added:

---

```
IF DATATYPE(RRESROL1)1 = 'NUM' THEN DO /* IS IT NUMERIC? */
  IMFEXEC MSG "'RRESROL1 NOT NUMERIC, RRESROL1='RRESROL1'"
  SIGNAL MPEXIT /* EXIT */
END
```

---

The REXX built-in function, `DATATYPE`, ensures that the number of lines is numeric. The `DATATYPE` function also can verify other characteristics. The `IMFEXEC MSG` facility requires the text of the message to be enclosed in single quotation marks, but within REXX, double and single quotation marks are interchangeable.

The samples use double quotation marks to tell REXX that the contents are literals so that REXX passes on the single quotation marks to the external environment, `IMFEXEC`. The `SIGNAL` command of REXX can be used to branch or trap certain conditions. REXX does not have a `GOTO` command so loops or branches must be controlled by `SIGNAL`, `LEAVE`, or `CALL` commands.

If the UCB in the `IOS071I` message has an outstanding reserve, MPE001X schedules the MPE002X EXEC. MPE002X issues an `ALERT` back to the system requesting the information about outstanding reserves. MPE002X is functionally equivalent to MPE002C.

BMC Software distributes two REXX samples, MPE002X and MPE002X2. Both use the `REXX PARSE` and `ADDRESS` instructions and the difference between MPE002X and MPE002X2 is how they build text for `ALERTs`. MPE002X uses single and double quotation marks and MPE002X2 uses string concatenations.

An ALERT KEY is built with the following instruction:

```
ALERTKEY=INSERT(SYSID,UCB)
```

Using the REXX INSERT function, a unique ALERT KEY is built. This instruction inserts the value of SYSID before the value of UCB, which is placed subsequently in a variable called ALERTKEY.

MPE002X issues the ALERT with the following instruction:

```
"IMFEXEC ALERT" ALERTKEY,
  '.MPE002X JOB "JOBNAME " ON SYSTEM "SYSID" USING ',
  UCB" HELD BY SYSTEM "SS"' "COLOR(WHITE)"
```

**Note:** Commas indicate continuation in REXX.

There are seven pairs of double quotation marks and one pair of single quotation marks in the above instruction. The same ALERT is issued with the following set of instructions in MPE002X2.

---

```
WORK1="'" /* DEFINE LEADING/TRAILING QUOTE */
WORK2='.MPE002X2 JOB ' /* FIRST PART OF THE MESSAGE */

/* CONCATENATE WORK1, WORK2 AND MORE OF THE TEXT */
WORK3=WORK1||WORK2||JOBNAME||' ON SYSTEM 'SYSID' USING 'UCB

/* BUILD REST OF THE TEXT WITH THE DELIMITING QUOTE AT END */
WORK4=', HELD BY SYSTEM '||SS||WORK1

TEXT=INSERT(WORK3,WORK4) /* INSERT WORK3 INFRONT OF WORK4 */
"IMFEXEC ALERT" ALERTKEY TEXT "COLOR(WHITE)"
```

---

MPE002X and MPE002X2 are designed to demonstrate the use of quotation marks and string concatenations in REXX. The choice is really stylistic; both results are the same.

### EXECs for REXX DASD Reserve Analysis

<b>IOS071IX</b>	Schedules remote EXECs threshold counters
<b>MPE001X</b>	Examines currently held reserves (runs on remote systems)
<b>MPE002X</b>	Issues ALERT back to requesting system
<b>MPE002X2</b>	Issues ALERT back to requesting system

## REXX QUEUE/PULL and VPUT/VGET Usage

Two sample EXECs, MUT002X and MUT003X, are distributed to show the use of REXX QUEUE/PULL and IMFEXEC's VPUT/VGET instructions. As with all REXX EXECs, the PARSE and ADDRESS commands are used. After these commands, MUT002X calls MUT003X to issue the MVS commands, D R,L. When entered, MUT003X establishes the IMFEXEC as the external environment with the ADDRESS instruction.

The following command displays the outstanding replies:

```
"IMFEXEC CMD 'D R,L' RESPONSE(IEE112I) WAIT(20)"
```

Upon return, the response lines are in the local pool. The variable IMFNOL contains the number of lines returned. The variable NUMLINES is ASSIGNED the values of IMFNOL and VPUT to the local pool.

MUT003X places information into the MAINVIEW AutoOPERATOR local pool and the REXX EXTENDED DATA QUEUE with the following set of instructions:

---

```
DO N=1 TO IMFNOL BY 1 UNTIL N>IMFNOL
  "IMFEXEC VDCL REXXL"N" LIST(TEXT B C D E F)"
  "IMFEXEC VGET LINE"N" INTO(REXXL"N") LOCAL"
  "IMFEXEC VPUT LINE."N" FROM(REXXL"N") LOCAL"
  QUEUE 'LINE.'N TEXT B C D E F
END
```

---

Although 'LINE'.N looks like a REXX STEM, a specific queued line cannot be referenced by the PULL command; therefore the pseudo stem, 'LINE.'.N, becomes a part of the data that queued.

The QUEUE command builds the extended data queue. After the loop is completed, MUT003X issues the RETURN command and processing is continued in MUT002X.

Upon return from MUT003X, MUT002X uses the REXX built-in function QUEUED to establish the number of lines queued and displays this information before setting up the loop to PULL from the extended data queue. The following sequence of instructions is executed:

---

```
CNT=QUEUED()  
IMFEXEC MSG '.MUT002X: THE NUMBER OF QUEUED LINES  
IS: 'CNT  
DO N=1 TO CNT BY 1 UNTIL N>CNT  
    PULL LINE TEXT B C D E F  
    IMFEXEC MSG '.MUT002X: PULL VARIABLES ARE:' LINE  
TEXT B C D E F  
END
```

---

Notice that `LINE` is considered a part of the data that is queued and not a REXX `STEM`. REXX discards the data once it has been pulled from the extended data queue. If you want to erase the data queue, build a `PULL` loop without specifying an argument template. The `PULL` instruction also can be specified as `PARSE PULL`.

`MUT002X` continues execution by getting `NUMLINES` from the `LOCAL POOL` and setting up a similar loop to issue `VDCL` and `VGET` instructions to derive the values that were stored in the `LOCAL POOL` by `MUT003X`.

### EXECs for REXX QUEUE/PULL and VPUT/VGET Usage

- MUT002X**      Invokes `MUT003X` to gather data then `VGET` the data from the `LOCAL POOL` and `PULL` the data from the `REX DATA QUEUE`
- MUT003X**      Issues the MVS command `D R,L` and places the data from the response in the `LOCAL POOL` and the `REXX DATA QUEUE`



---

# Appendix D MVS Solution Variables

To ensure that the MVS Solutions function properly in your environment, you must establish values for several variables. Table D-1 lists all MVS Solution variables, their default values, and the solutions each one affects. Edit member MSU002C in the BBPROC data set, distributed in MAINVIEW AutoOPERATOR, to set values for the appropriate variables.

A Rule for message PM0010I is required to schedule MSU002C upon PAS initialization.

**Table D-1 MVS Solution Variables (Part 1 of 2)**

Name	Description	Default	Solutions
dsmm1mig	ML1 migration in process flag for DASD/Space Monitoring	n	Dump Data Set Monitoring
dumpcler	task to clear dump data sets	None	Dump Data Set Monitoring
dumpmf	minimum number of dump data sets to keep clear	1	Dump Data Set Monitoring
dumpnds	total number of dump data sets	4	Dump Data Set Monitoring
lbgclas	class priority list from high to low	MDRAI53	Load Balancing
lbgcpuh	CPU utilization to decrease workload	90	Load Balancing
lbgcpul	CPU utilization to increase workload	50	Load Balancing
lbgibeg	beginning initiator number to manage	None	Load Balancing
lbgidec	Number of initiators to stop	1	Load Balancing
lbiend	ending initiator number to manage	None	Load Balancing
lbiinc	number of initiators to start	2	Load Balancing
lbgpage	Page rate to decrease workload	xxx	Load Balancing
logrec	task name to clear SYS1.LOGREC	None	Logrec Data Set Monitoring
maxacsa	maximum extended CSA utilization	80	Performance Management

**Table D-1 MVS Solution Variables (continued) (Part 2 of 2)**

<b>Name</b>	<b>Description</b>	<b>Default</b>	<b>Solutions</b>
maxbcsa	maximum CSA utilization below the limit	80	Performance Management
maxccpu	maximum complex CPU utilization	95	Performance Management
maxdevu	maximum device utilization	80	Performance Management
maxtcpu	maximum TSO CPU utilization	50	Performance Management
mvscall	name of the support person to be paged; this name must be defined to the Elan Workstation	None	LOGREC Data Set Monitoring VSAM Failures
smfalt	suffix of alternate SMF parameters	None	SMF Data Set Monitoring
smfclear	task to dump SMF data sets	None	SMF Data Set Monitoring
sysbeep	information to be placed on pager	None	WTO Buffer Monitoring
sysjes	version, release, and modification level of JES	SP3.1.3	Solutions that depend upon release-specific messages
sysmvs	version, release, and modification level of MVS	SP4.2.2	Solutions that depend upon release-specific messages
sysn	number of MVS systems in shared DASD complex	2	DASD Reserve Analysis Shared DASD Control
sysprog	tsO USERID of primary system programmer to receive warning messages	None	WTO Buffer Monitoring Logrec Data Set Monitoring
sys1 - sysn	system IDs of systems in a shared DASD configuration	None	DASD Reserve Analysis

---

## Appendix E CICS Solution Variables

To ensure that the CICS Solutions function properly in your environment, you must establish values for several variables. Table E-1 on page E-2 lists all CICS Solution variables, their default values, and the solutions each one affects. Edit member CSUINIT in the BBPROC data set, distributed with MAINVIEW AutoOPERATOR, to set values for the appropriate variables.

If JRNLEXEC=YES is specified in BBPARM member BBISSP00, MSU002C is automatically scheduled to execute when the PM0010I BBI/PM Initialization Complete message is received. Therefore, it sets the variable values each time the MAINVIEW AutoOPERATOR subsystem is started.

If JRNLEXEC=N is specified, or JRNLEXEC is not specified, a Rule is required for PM0010I.

The CICS Solution variables also can be set by an operator. To reset the CICS Solution variables, enter

```
%CSUINIT
```

**Table E-1 CICS Solution Variables**

<b>Name</b>	<b>Description</b>	<b>Default</b>	<b>Solutions</b>
FOCALPT	SSID of MAINVIEW AutoOPERATOR subsystem to receive all ALERTs	None	<ul style="list-style-type: none"> <li>• Error Recovery</li> <li>• Performance Management</li> </ul>
ONLCNT01 - ONLCNT0n	name of the controlling subsystem	None	<ul style="list-style-type: none"> <li>• Error Recovery</li> <li>• File Degradation Analysis</li> <li>• VSAM Control Split Monitor</li> <li>• Transaction Response Time Monitor</li> </ul>
ONLALT01 - ONLALT0n	color of ALERTs messages for target system n	RED	<ul style="list-style-type: none"> <li>• Error Recovery</li> <li>• File Degradation Analysis</li> <li>• VSAM Control Split Monitor</li> <li>• Transaction Response Time Monitor</li> </ul>
ONLTYP01 - ONLTYP0n	type of target system n or BBI-SSn	None	<ul style="list-style-type: none"> <li>• Error Recovery</li> <li>• File Degradation Analysis</li> <li>• VSAM Control Split Monitor</li> <li>• Transaction Response Time Monitor</li> </ul>
ONLALM01 - ONLALM0n	alarm indicator for ALERT in target system n	Y	<ul style="list-style-type: none"> <li>• Error Recovery</li> <li>• File Degradation Analysis</li> <li>• VSAM Control Split Monitor</li> <li>• Transaction Response Time Monitor</li> </ul>
ONLSSN	number of online subsystems	4	<ul style="list-style-type: none"> <li>• Error Recovery</li> <li>• File Degradation Analysis</li> <li>• VSAM Control Split Monitor</li> <li>• Transaction Response Time Monitor</li> </ul>
ONLSS01 - ONLSS0n	names of the target subsystems	None	<ul style="list-style-type: none"> <li>• Error Recovery</li> <li>• File Degradation Analysis</li> <li>• VSAM Control Split Monitor</li> <li>• Transaction Response Time Monitor</li> </ul>

---

---

# Appendix F DB2 Solution Variables and Parameters

To ensure that the DB2 Solutions function properly in your environment, you must establish values for several variables and determine threshold parameters.

## Data Sets Containing Variables

DB2 Solution variables are found in the following data sets:

- hilevel.UBBPROC(DSU001C)
- hilevel.UBBPROC(DSU101C)
- hilevel.UBBPARM(BBIISP00)
- hilevel.UBBPARM(BBISSP00)

**DSU001C** an EXEC that initializes DB2 variables required for all DB2 Solutions

**DSU101C** an EXEC that initializes variables per DB2 subsystem

You can have as many as nine DB2 subsystems. Variables for these additional subsystems are set by copying member DSU101C and renaming the member

### **DSU10nC90**

where *n* is the DB2 subsystem (1-9) for which you are setting variables.

---

**BBIISP00** contains a startup list to which you add BLK members DPE002B and DPE030B during initial customization

**BBISSP00** contains Rules parameters that you update during initial customization

**Note:** See the *MAINVIEW Common Customization Guide* for information on the installation of UBBPARM and UBBPROC data sets.

## Data Sets Containing Threshold Parameters

DB2 threshold parameters are found in these data sets:

- hilevel.UBBPARM(DPE002B)
- hilevel.UBBPARM(DPE030B)
- hilevel.UBBPARM(DMRBEX00)

**DPE002B** DPE002B and DPE030B contain parameters that specify the names of monitors you add during initial customization of the DB2 Solutions

**DPE030B** See DEP002B.

**DMRBEX00** contains parameters for setting threshold levels in the DB2RNWY Solution during initial customization of the DB2 Solutions

## Tables That Describe Variables and Parameters

Variables and threshold parameters for the DB2 solutions are organized in tables according to the solutions that each one affects:

**All DB2 Solutions** Table F-1

**DB2DLOK** Table F-2

**DB2RESP Variables** Table F-3

**DB2RESP Thresholds** Table F-4

**DB2RNWY** Table F-5

**DB2TFUL** Table F-6

**DB2THRD** Table F-7

**Note:** You must set the following variables listed in the **All DB2 Solutions** (see Table F-1):

- DB2
- D\_DB2NUM

These variables are part of the initial environment set up and you *must* set both of them before using any of the DB2 Solutions.

The rest of the variables found in the following five tables are optional. Defaults are provided for them.

## All DB2 Solutions

**Table F-1** Variables for All DB2 Solutions in hilevel.UBBPROC

Variable Name	Description	Default	Data Set Member
d_db2num	number of DB2s monitored by this subsystem  <b>Note:</b> Setting this variable is mandatory.	1	DSU001C
d_alert_ids	USERIDs to be notified of DB2 Alerts	Null	DSU001C
d_alert_q	queue name to use for Alerts	DB2	DSU001C
d_alert_send	indicates (Y N) that SEND command is to be issued	N	DSU001C
d_alert_sendopt	SEND option: either LOGON or NOW	NOW	DSU001C
db2A	name you supply to identify each DB2 subsystem	DB2D	DSU101C
d_alert_tgt	target for MAINVIEW Alerts	&IMFORGSS	DSU001C
mstr	DB2 MSTR address space name associated with this DB2 (for example, DB2AMSTR, DSNBMSTR, and so on)	&DB2.MSTR	DSU101C
dbm1	DB2 DBM1 address space name associated with this DB2 (for example, DB2ADBM1, DSNBDBM1, and so on)	&DB2.DBM1	DSU101C
dist	DB2 DIST address space name associated with this DB2 if applicable at your site (for example DB2ADIST, DSNBDIST, and so on)	&DB2.DIST	DSU101C
irlm	IRLM PROC name associated with this DB2 subsystems (for example DB2PROC, IRLMPROC, and so on)	&DB2.PROC	DSU101C
imst	optional IMS target name associated with this DB2 subsystem (null if not applicable)	Null	DSU101C

---

## DB2DLOK

**Table F-2 DB2DLOK Variables in hilevel.UBBPROC**

<b>Variable Name</b>	<b>Description</b>	<b>Default</b>	<b>Data Set Member</b>
cnt	total number of deadlock/timeout history entries to be maintained in the dynamic Deadlock History Table for this DB2 subsystem	20	DSU101C
intvl	length of time in seconds to wait between LOCKD displays; this should be half your IRLM deadlock time	5	DSU101C
max_age	discard deadlock events over this age (rolling time interval in minutes that a particular plan involved in deadlock/time can be considered a candidate for analysis for this DB2 subsystem)	30 minutes	DSU101C
maxeither	maximum number of times that a plan can be EITHER a lock owner or a lock requestor in a deadlock/timeout within the MAX_AGE time interval before triggering the Deadlock Analysis Solution	3	DSU101C
maxowner	maximum number of times that a plan can be a lock OWNER in a deadlock/timeout within the MAX_AGE time interval before triggering the Deadlock Analysis Solution	2	DSU101C
maxreq	maximum number of times that a plan can be a lock REQUESTOR in a deadlock/timeout within the MAX_AGE time interval before triggering the Deadlock Analysis/Timeout Solution	2	DSU101C
stoptime	length of time in minutes to run the LOCKD data extractor to analyze lock contentions	10 minutes	DSU101C

## DB2RESP Variables

**Table F-3 DB2RESP Variables in hilevel.UBBPROC (Part 1 of 2)**

Variable Name	Description	Default	Data Set Member
max_tsoelap	maximum elapsed time per TSO user	00:01:00 (1 minute)	DSU101C
max_tsostmt	maximum number of SQL statements issued per TSO user	500	DSU101C
max_tsolock	maximum number of LOCKS held per TSO user	500	DSU101C
max_batelap	maximum elapsed time per BATCH transaction	01:00:00 (1 hour)	DSU101C
max_batstmt	maximum number of SQL statements issued per BATCH transaction	100,000	DSU101C
max_batlock	maximum number of LOCKs held per BATCH transaction	2,000	DSU101C
max_utlelap	maximum elapsed time per UTILITY transaction	00:30:00 (30 minutes)	DSU101C
max_utlstmt	maximum number of SQL statements issued per UTILITY transaction	0	DSU101C
max_utllock	maximum number of LOCKs held per UTILITY transaction	50	DSU101C
max_cafelap	maximum elapsed time per CAF transaction	00:30:00 (30 minutes)	DSU101C
max_cafstmt	maximum number of SQL statements issued per CAF transaction	100,000	DSU101C
max_caflock	maximum number of LOCKs held per CAF transaction	2,000	DSU101C
max_tpelap	maximum elapsed time per CICS or IMS transaction	00:00:05 (5 seconds)	DSU101C
max_tpstmt	maximum number of SQL statements issued per CICS or IMS transaction	100	DSU101C
max_tpllock	maximum number of LOCKs held per CICS or IMS transaction	100	DSU101C
max_edmfails	maximum number of EDM pool failures per minute	0	DSU101C
max_edmloads	maximum percent of EDM pool requests needing loads	20%	DSU101C
maxbfr_unavail	maximum number of buffer unavailable conditions per minute	0	DSU101C
maxalloc_delay	maximum number of allocation delay conditions per minute	0	DSU101C
max_ridut	maximum RID pool percent utilization	80%	DSU101C
max_bfrexp	maximum number of buffer pool expansions per minute	0	DSU101C

**Table F-3 DB2RESP Variables in hilevel.UBBPROC (Part 2 of 2)**

<b>Variable Name</b>	<b>Description</b>	<b>Default</b>	<b>Data Set Member</b>
max_bfrdmc	maximum number of Data Manager Critical conditions per minute	0	DSU101C
max_bfriw	maximum number of Immediate Write conditions per minute	0	DSU101C
min_getrio	minimum number of GETPAGES per read I/O in the INTVL	2.0	DSU101C
min_swspws	minimum number of system page updates per system pages written in the INTVL	1.0	DSU101C
min_pwsvio	minimum number of system pages written per write I/O in the INTVL	5.0	DSU101C
resp_min	number of minutes between monitor warning messages before triggering the EXEC again if necessary	1	DSU101C
max_suspct	percent suspended threads (total active threads divided by total suspended threads)	25%	DSU101C
max_thdutil	thread utilization (total active threads divided by maximum threads allowed)	90%	DSU101C
max_#susp	maximum suspensions per active thread per minute (suspensions per minute divided by number of active threads)	1	DSU101C
max_#dead	maximum deadlocks per minute in the interval	0	DSU101C
max_#timeout	maximum timeouts per minute in the interval	0	DSU101C
max_pagepct	maximum DB2 paging as a percent of total system (DB2 paging divided by total system paging)	10%	DSU101C
max_edmut	maximum EDM pool utilization	80%	DSU101C
max_openpct	maximum open data set utilization	90%	DSU101C

## DB2RESP Monitor Thresholds

**Table F-4 Monitor Thresholds for DB2RESP in hilevel.UBBPARM**

Monitor	Connection	Description	Threshold	Data Set Member
@ELAP	IMS	average elapsed time	5 seconds	DPE002B
@ELPD	IMS	average elapsed time in DB2	3 seconds	DPE002B
@CPU	IMS	average CPU time	0.5 seconds	DPE002B
@CPUD	IMS	average CPU time in DB2	0.3 seconds	DPE002B
@ELAP	TSO	average elapsed time	15 seconds	DPE002B
@ELPD	TSO	average elapsed time in DB2	5 seconds	DPE002B
@CPU	TSO	average CPU time	1 second	DPE002B
@CPUD	TSO	average CPU time in DB2	0.5 seconds	DPE002B
@ELAP	CICS	average elapsed time	5 seconds	DPE002B
@ELPD	CICS	average elapsed time in DB2	3 seconds	DPE002B
@CPU	CICS	average CPU time	0.5 seconds	DPE002B
@CPUD	CICS	average CPU time in DB2	0.3 seconds	DPE002B
@CPU	BATCH	average CPU time	0.5 seconds	DPE002B
@CPUD	BATCH	average CPU time in DB2	0.3 seconds	DPE002B

## DB2RNWY

**Table F-5 DB2RNWY Threshold Parameters in hilevel.UBBPARM**

Keyword	Connection	Threshold Parameter	Data Set Member
MSG=DZ0610W	IMS	GPRATE=200 CYCLES=3	DMRBEX00
MSG=DZ0620W	CICS	GPRATE=200 CYCLES=3	DMRBEX00
MSG=DZ0630W	TSO	GPRATE=200 CYCLES=3	DMRBEX00
MSG=DZ0640W	BATCH	GPRATE=500 CYCLES=3	DMRBEX00
MSG=DZ0650W	CAF	GPRATE=500 CYCLES=3	DMRBEX00
MSG=DZ0660W	UTILITY	GPRATE=1000 CYCLES=3	DMRBEX00

## DB2TFUL

**Table F-6 DB2TFUL Variables in hilevel.UBBPROC**

Variable Name	Description	Default	Data Set Member
MAX_UTIL	default maximum database/tablespace (DB/TS) utilization for all DB/TSs associated with this subsystem; zero means no default threshold  This value can be overridden for specific DB/TSs within each DB2 subsystem.	0	DSU101C
MAX_EXTS	default maximum number of extents for all DB/TSs associated with this DB2 subsystem  This value can be overridden for specific DB/TSs within each DB2 subsystem.	0	DSU101C
NUM	total number of overrides for this DB2 subsystem	4	DSU101C
NAME $n$	$n$ th DBTS name to be overridden for percent space utilization and extent monitoring for this DB2 subsystem  Generic DBTS names may be used.	NA	DSU101C
NAME1	first DBTS name	DSN*	DSU101C
NAME2	second DBTS name	DSNDB06_ DSNAPH01	DSU101C
NAME3	third DBTS name	DSNDB07_ DSN4K01	DSU101C
NAME4	fourth DBTS name	DSNDB06_ SYS*	DSU101C
MAX_UTIL $n$	utilization threshold for all DB/TSs that satisfy the NAME $n$ qualification  Zero means ignore percent utilization.	NA	DSU101C
MAX_UTIL1	utilization threshold for NAME1	0	DSU101C
MAX_UTIL2	utilization threshold for NAME2	90	DSU101C
MAX_UTIL3	utilization threshold for NAME3	50	DSU101C
MAX_UTIL4	utilization threshold for NAME4	50	DSU101C
MAX_EXTS $n$	extent threshold for all DB/TS's that satisfy the NAME $n$ qualification  Zero means ignore percent utilization.	NA	DSU101C
MAX_EXTS1	extent override for NAME1	0	DSU101C
MAX_EXTS2	extent override for NAME2	1	DSU101C
MAX_EXTS3	extent override for NAME3	20	DSU101C
MAX_EXTS4	extent override for NAME4	1	DSU101C

---

**DB2THRD****Table F-7 DB2TFUL Variables in hilevel.UBBPROC**

<b>Variable Name</b>	<b>Description</b>	<b>Default</b>	<b>Data Set Member</b>
Q_VERIFY	indicates (Y N) whether or not to issue an ALERT to verify the TSO cancel	Y	DSU101C



---

# Index

## Symbols

\$CSAABV 21-11  
\$CSABLO 21-11  
\$HASP050 3-2, 3-3, 3-4  
\$HASP093 3-2, 3-3, 3-4  
\$HASP095 3-2, 3-3, 3-4  
\$HASP355 3-3, 3-4  
\$HASP646 3-2, 3-3, 3-4  
\$JOB CPU 21-11  
\$STCCPU 21-11  
\$TOTCPU 9-7, 21-11  
\$TOTSIO 21-11  
\$TOTSIO2 21-11  
\$TSOCPU 9-7, 21-11  
\$WATCH 21-11  
&plan 15-5  
@DMPCLRO 5-3  
@TIMER B-1

## A

AAORUL00 21-1  
abend of CICS 10-4  
Access Methods Solution 6-1, 17-1, 18-1  
    *See also* VSAM Failures Solution  
Access Methods Solutions  
    *See* VSAM Failures Solution  
AEW messages 9-5  
ALERT application 9-5  
ALERT profile, modifying 13-4

ALERTS ISSUED field, documentation box  
    21-6  
AUTHOR field  
    documentation box 21-5  
    where displayed 1-3  
Automation Reporter 19-1  
AutoOPERATOR  
    ALERTs 19-1  
    introduction to using Sample Solutions 21-1  
        to 21-6  
    Sample CICSplex Solution 21-12  
    Sample Info/System Management Solution  
        21-14 to 21-15  
    Sample Monitoring Solution 21-9 to 21-11  
    Sample VTAM/NCP Solution 21-6 to 21-8  
    Web 19-1

## B

batch jobs 11-4 to 11-6  
BBISSP00 5-4  
BBPARM data set 21-1  
    *See also* each member name  
BBPROC data set 21-1  
    *See also* each member name  
BBUSER data set 21-14  
bouncing objects, CSM 2-116

---

## C

- canceling objects, CSM 2-122
- Case Sensitivity 19-4
- CAT2291D 4-3
- CER001C 10-3
- CHANGE LOG field, documentation box 21-6
- CICS 2-8, 10-4
- CICS Abnormal Termination Solution
  - description 10-4
  - EXECs 10-5
  - invoking 10-4
  - processing flow 10-4
  - Rules 10-5
  - variables 10-4
- CICS MANAGER, MONITOR command 11-9
- CICS Performance Management Solutions 11-1 to 11-6
  - See also* File Allocation - Deallocation Solution, File Degradation Analysis Solution, Transaction Response Time Monitor Solution, VSAM Control Area Split Monitor Solution
- CICS Solutions
  - Error Recovery Solutions 10-1 to 10-11
    - See also* CICS Abnormal Termination Solution, CICS Temporary Storage Suspensions Solution, Storage Violation Solution, VSAM Subtask Abnormal Termination Solution
  - Performance Management Solutions 11-1 to 11-9
    - See also* File Allocation - Deallocation Solution, File Degradation Analysis Solution, Transaction Response Time Monitor Solution, VSAM Control Area Split Monitor Solution
  - CICS Temporary Storage Suspensions Solution
    - EXECs 10-8
    - invoking 10-7
    - processing flow 10-7
    - Rules 10-8
    - variables 10-7
  - CICS, abend 10-4
  - client-server relationship, CSM 2-8
  - CMDCON parameter 5-4
  - CODE field 1-3, 21-5
  - commands 19-1
  - COMPLETE, CSM object state 2-133
  - Conditional Command Specifications Panel 2-161
  - conflicts, CSM schedules 2-104
  - connecting CSM objects 2-15
  - Continuous State Manager (CSM)
    - Abnormal Termination Events 2-18, 2-64
    - actual states 2-10
    - associating objects with a group 2-68
    - benefits 2-3
    - bouncing objects 2-116
    - canceling objects 2-122
    - client-server relationship 2-8
    - command specification 2-57
    - conditional command specification 2-61
    - connecting 2-15
    - creating a group 2-51
    - cross-system management
      - description 2-37
      - enabling 2-108
    - defining objects 2-51
    - defining parents 2-71
    - defining servers 2-72
    - diagnosing errors A-1
    - disconnecting 2-15
    - dmessages 2-14
    - exceptional schedules, example 2-30
    - grouping object 2-6, 2-75
    - IPL, example 2-22
    - masking panels 2-43
    - moving objects 2-70, 2-125
    - normal object 2-5
    - object states 2-130 to 2-141
    - objects 2-5, 2-8
    - overview 2-2 to 2-32
    - parent-child relationships 2-6, 2-7
    - parents
      - adding 2-79
      - deleting 2-83
    - planning 2-32
      - commands 2-35
      - object relationships 2-34
      - schedules 2-34
    - recovering objects 2-17
    - resetting objects 2-119
    - routine schedules, example 2-29

---

## Continuous State Manager (CSM) *(continued)*

- schedules 2-11
  - conflicts 2-104
  - creating 2-93
  - modifying 2-96, 2-100
  - overriding 2-106
- scheduling EXEC 2-128
- scheduling objects 2-17
- servers
  - adding 2-86
  - deleting 2-90
- shutdown, example 2-25
- specifying start/stop events 2-63
- specifying user notification 2-66
- start an object 2-110
- Start Command Limits 2-18
- starting commands 2-14
- starting CSM 2-16
- stop an object 2-114
- stopping commands 2-14
- transient object 2-5, 2-74
- viewed from Web browser 19-1

## CONTROLM Solution

- invoking 8-1
- processing flow 8-2
- Rules 8-2
- variables 8-1

## CPU usage 9-3

## CPU, Sample Monitoring Solution 21-10

## cross-system management, enabling 2-108

## CSAABV monitor, Sample Monitoring Solution 21-10

## CSABLO monitor, Sample Monitoring Solution 21-10

## CSM Global Overview panel

- description 2-44 to 2-47
- field descriptions 2-143
- line commands 2-147
- primary commands 2-144

## CSM panel description

- Client/Server Relationships 2-174
- Command Specifications 2-155
- Conditional Command Specifications 2-160
- CSM Global Overview 2-142
- Event Specifications 1 2-163
- Event Specifications 2 2-166
- Global Calendar Override 2-180
- Group Detail Control 2-188

## CSM panel description *(continued)*

- Object Detail Control 2-152
- Object Groups 2-170, 2-184
- Object Requisites 2-172
- Object Scheduling 2-177
- User Notification 2-168

## CSMACT EXEC

- commands 2-136
- description 2-21
- examples 2-136
- modify objects 2-135
- parameters 2-135

## CSMINFO EXEC 2-133

## CSSUM command, Sample Monitoring Solution 21-10

## customization

- EXECs 13-1 to 13-4
- initial 13-1 to 13-4

## D

## D\_ALERT\_IDS variable 14-6

## D\_ALERT\_Q variable 14-6

## D\_ALERT\_SEND variable 14-6

## D\_ALERT\_SENDOPT variable 14-6

## D\_ALERT\_TGT variable 14-6

## D\_DB2n variable 14-6

## D\_DB2NUM variable 14-6

## D\_DBM1n variable 14-6

## D\_DISTn variable 14-6

## D\_IRLMn variable 14-6

## D\_MSTRn variable 14-6

## DASD Reserve Analysis Solution

### EXECs 9-5

### invoking 9-4

### processing flow 9-4

### Rules 9-5

### variables 9-4

## DB2 2-8, 19-2

## DB2 Environment Set-Up

### EXECs 14-7

### invoking 14-7

### processing flow 14-7

### Rules 14-7

### variables 14-6

---

DB2 Global Operations Solutions  
*See* DB2ALRT Solution, DB2INIT Solution, DMRALRT Solution, DMRUTIL Solution

DB2 Major Messages  
 EXECs 14-5  
 invoking 14-2  
 processing flow 14-5  
 Rules 14-5  
 variables 14-2

DB2 Performance Management Solutions 16-1 to 16-6  
*See also* DB2RESP Solution, DB2RNWY Solution

DB2 Resource Contention Analysis Solutions  
*See* DB2DLOK Solution, DB2TFUL Solution, DB2THRD Solution

DB2 Response Time  
 EXECs 16-6  
 invoking 16-5  
 processing flow 16-5  
 Rules 16-6  
 variables 16-4

DB2 Runaway Query Control  
 EXECs 16-3  
 invoking 16-2  
 processing flow 16-2  
 Rules 16-3  
 variables 16-2

DB2 Solutions  
 DB2 Global Operations Solutions 14-1 to 14-12  
 DB2 Performance Management Solutions 16-1 to 16-6  
 Resource Contention Analysis Solutions 15-1 to 15-11

DB2 Subsystems, determining the number of 13-2

DB2 Table Space Filling Up Solution (DB2TFUL)  
 EXECs 15-9  
 invoking 15-9  
 processing flow 15-9  
 Rules 15-9  
 variables 15-8

DB2 Thread Control  
 EXECs 15-11  
 invoking 15-10  
 processing flow 15-11  
 Rules 15-11  
 variables 15-10

DB2ALRT Solution  
 definition 14-1  
 EXECs 14-5  
 invoking 14-2  
 processing flow 14-5  
 Rules 14-5  
 variables 14-2

DB2DLOK processing flow  
 ALERTs 15-5  
 data extractor 15-5  
 history table 15-3

DB2DLOK Solution  
 EXECs 15-6  
 invoking 15-3  
 processing flow 15-3 to 15-5  
 Rules 15-7  
 variables 15-2

DB2INIT Solution  
 description 14-5  
 EXECs 14-7  
 invoking 14-7  
 processing flow 14-7  
 Rules 14-7  
 variables 14-6

DB2RESP Solution  
 EXECs 16-6  
 invoking 16-5  
 processing flow 16-5  
 Rules 16-6  
 variables 16-4

DB2RNWY Solution  
 EXECs 16-3  
 invoking 16-2  
 processing flow 16-2  
 Rules 16-3  
 variables 16-2

DB2TFUL Solution  
 EXECs 15-9  
 invoking 15-9  
 processing flow 15-9  
 Rules 15-9  
 variables 15-8

---

DB2THRD Solution  
     EXECs 15-11  
     invoking 15-10  
     processing flow 15-11  
     Rules 15-11  
     variables 15-10  
 Deadlock/Timeout Analysis  
     EXECs 15-6  
     invoking 15-3  
     processing flow 15-3 to 15-5  
     Rules 15-7  
     variables 15-2  
 defining CSM exceptional schedules 2-180  
 DESC field, documentation box 1-3, 21-5  
 Description field, documentation box 21-5  
 desired states 2-10  
 determining CSM schedules 2-93  
 DFH0508 10-2, 10-3  
 DFH0606 10-4, 10-5  
 DFH0901 10-6  
 DFH1311 10-9, 10-10  
 DFH3437I 10-11  
 disconnecting CSM objects 2-15  
 DISP field, documentation box 1-3, 21-5  
 DMR Major Messages  
     EXECs 14-9  
     invoking 14-8  
     processing flow 14-9  
     Rules 14-10  
     variables 14-8  
 DMR Utilities  
     EXECs 14-11  
     invoking 14-10  
     Rules 14-12  
     variables 14-10  
 DMRALRT Solution  
     EXECs 14-9  
     invoking 14-8  
     processing flow 14-9  
     Rules 14-10  
     variables 14-8  
 DMRUTIL Solution  
     EXECs 14-11  
     invoking 14-10  
     Rules 14-12  
     variables 14-10  
 documentation box  
     description 1-2 to 1-4  
     example 21-3  
     fields 21-5 to 21-6  
 DOWNFORCE, CSM object state 2-132  
 DOWNLOGICAL, CSM object state 2-132  
 DOWNLOGICALE, CSM object state 2-132  
 DPE002C 14-11, 16-6  
 DPE005C 16-6  
 DPE007C 16-6  
 DPE010C 15-6  
 DPE011C 15-6  
 DPE012C 15-6  
 DPE013C 15-6  
 DPE014C 14-11, 15-6  
 DPE015C 15-6  
 DPE019C 15-6  
 DPE020C 16-3  
 DPE021C 16-3  
 DPE022C 16-3  
 DPE023C 16-3  
 DPE024C 16-3  
 DPE025C 16-3  
 DPE026C 16-3  
 DPE027C 16-3  
 DPE030C 15-9  
 DPE031C 15-9  
 DPE040C 15-11  
 DPE041C 15-11  
 DPE900C 14-11  
 DPE900D 14-11  
 DPE901C 14-11  
 DPE901D 14-11  
 DPE902C 14-11  
 DPE902D 14-11  
 DPE903C 14-11  
 DPE903D 14-11  
 DPE905C 14-11  
 DPE905D 14-11  
 DPE908C 14-11  
 DPE908D 14-11  
 DPE908DR 14-11  
 DPE909C 14-11  
 DPE917C 14-11  
 DPE919C 14-11  
 DPE920C 14-11  
 DPE930C 14-11  
 DPE931C 14-11

---

DPE932C 14-12  
DPE933C 14-12  
DPE935C 14-12  
DSNT375I 15-3, 15-7  
DSNT376I 15-3, 15-7  
DSNT501I 15-3  
DSNT501X 15-7  
DSU001C 14-7  
DSU10nC 14-7  
Dump Data Sets Monitoring Solution  
    EXECs 5-3  
    invoking 5-2  
    processing flow 5-2  
    Rules 5-3  
    variables 5-1  
DUMPCLER variable 5-1  
DUMPMF variable 5-1  
DUMPNDS variable 5-1

## E

enabling, cross-system management 2-108  
ERB102I 5-10  
Error Recovery Solutions 10-1 to 10-11  
    *See also* CICS Abnormal Termination  
        Solution, CICS Temporary Storage  
        Suspensions Solution, Storage Violation  
        Solution, Temporary Storage Data Set  
        Extension Failure Solution, Termination  
        Errors Solution, VSAM Subtask  
        Abnormal Termination Solution  
Exception Monitoring Solution  
    EXECs 9-7  
    invoking 9-6  
    processing flow 9-7  
    Rules 9-8  
    variables 9-5  
EXEC Management application  
    documentation box 1-2, 21-5  
    JESDOWN B-1

EXECs  
    \$CSAABV 21-11  
    \$CSABLO 21-11  
    \$HASP050 3-3  
    \$HASP093 3-3  
    \$HASP095 3-3  
    \$HASP355 3-3  
    \$HASP646 3-3  
    \$JOBCPU 21-11  
    \$STCCPU 21-11  
    \$TOTCPU 9-7, 21-11  
    \$TOTSIO 21-11  
    \$TOTSIO2 21-11  
    \$TSOCPU 9-7, 21-11  
    \$WATCH 21-11  
    @DMPCLRO 5-3  
    BBPROC 21-1  
    CER001C 10-3  
    CSM scheduling 2-128  
    DFH0508 10-3  
    DFH0606 10-5  
    DFH1311 10-10  
    DPE002C 14-11, 16-6  
    DPE005C 16-6  
    DPE007C 16-6  
    DPE010C 15-6  
    DPE011C 15-6  
    DPE012C 15-6  
    DPE013C 15-6  
    DPE014C 14-11, 15-6  
    DPE015C 15-6  
    DPE019C 15-6  
    DPE020C 16-3  
    DPE021C 16-3  
    DPE022C 16-3  
    DPE023C 16-3  
    DPE024C 16-3  
    DPE025C 16-3  
    DPE026C 16-3  
    DPE027C 16-3  
    DPE030C 15-9  
    DPE031C 15-9  
    DPE040C 15-11  
    DPE041C 15-11  
    DPE900C 14-11  
    DPE900D 14-11  
    DPE901C 14-11  
    DPE901D 14-11

---

EXECs (*continued*)

DPE902C 14-11  
DPE902D 14-11  
DPE903C 14-11  
DPE903D 14-11  
DPE905C 14-11  
DPE905D 14-11  
DPE908C 14-11  
DPE908D 14-11  
DPE908DR 14-11  
DPE909C 14-11  
DPE917C 14-11  
DPE919C 14-11  
DPE920C 14-11  
DPE930C 14-11  
DPE931C 14-11  
DPE932C 14-12  
DPE933C 14-12  
DPE935C 14-12  
DPE994C 14-9  
DSU001C 14-7  
DSU101C 13-1  
DSU10nC 14-7  
ERB102I 5-10  
IDC3009I 6-2  
IDC3351I 6-2  
IEA404A 5-4  
IEA405E 5-4  
IEA911E 5-3  
IEA994A 5-3  
IEA994E 5-3  
IEC701D 4-5  
IEE249I 5-4  
IEE361I 5-7  
IEE362A 5-7  
IEE364I 5-7  
IEE366I 5-7  
IEE391I 5-7  
IEE392I 5-7  
IEE393I 5-7  
IEE949I 5-7  
IEE974I 5-7  
IEF238D 4-5  
IEF433D 4-5  
IEF450I 21-15  
IKT001D 12-2  
IKT010D 12-2  
IKT012D 12-2

EXECs (*continued*)

IOS071I 9-5  
JESDOWN B-1  
Job Scheduling Solution 7-2  
LBSTART 9-3  
MJSU001C 7-2  
MJSU002C 7-2  
MJSU003C 7-2  
MJSU004C 7-2  
MJSU005C 7-2  
MJSU006C 7-2  
MJSU007C 7-2  
MJSU008C 7-2  
MMN001C 5-7  
MMN002C 21-11  
MMN003C 21-11  
MMN004C 5-7  
MPE001C 9-5  
MPE002C 9-5  
MPE003C 9-3  
MPE004C 9-3  
MPE005C 9-3  
MPE006C 9-3  
MSU002C 7-2  
MSU005C 7-2  
naming conventions 21-1  
national characters 1-5, 21-6  
NETACT 21-8  
NETBOUNC 21-8  
NETINACT 21-8  
OSPINFO1 21-15  
PM00101 7-2  
PWSCPU01 9-7  
PWSCPU02 9-7  
PWSCSA01 9-7  
PWSDEV01 9-7  
SCHEDCOM 8-2  
TIOTSIO2 21-10  
TLMS 4-4  
EXTERNAL ROUTINES CALLED field,  
documentation box 21-5

---

## F

failed objects, recovering 2-17

File Allocation - Deallocation Solution

EXECs 11-6

invoking 11-5

JCL 11-4

processing flow 11-5

Rules 11-6

variables 11-4

File Degradation Analysis Solution

EXECs 11-3

invoking 11-2

processing flow 11-2

Rules 11-4

variables 11-2

FOCALPT variable 10-2

FT094W 11-9

FUNC field 1-3, 21-5

## G

GROUP field 1-3

Group field 21-5

Group ID 19-4

grouping objects

defining 2-75

description 2-6

## I

IDC3009I 6-2

IDC3351I 6-2

IEA404A 5-4, 5-5

IEA405E 5-4, 5-5

IEA911E 5-2, 5-3

IEA994A 5-2, 5-3

IEA994E 5-2, 5-3

IEC701D 4-4, 4-5

IEE249I 5-4

IEE361I 5-6, 5-7

IEE362A 5-6, 5-7

IEE364I 5-6, 5-7

IEE366I 5-6, 5-7

IEE391I 5-6, 5-7

IEE392I 5-6, 5-7

IEE392II 5-7

IEE393I 5-6, 5-7

IEE949I 5-7

IEE974I 5-7

IEF238D 4-4, 4-5

IEF433D 4-4, 4-5

IEF450I, Info/System Management Solution  
21-14, 21-15

IFB040I 5-8, 5-9

IFB060E 5-8, 5-9

IFB070I 5-8, 5-9

IKT001D 12-1, 12-2

IKT010D 12-1, 12-2

IKT012D 12-1, 12-2

Info/System Management Sample Solution

EXECs 21-15

invoking 21-14

Rules 21-15

Initial Customization Steps 13-1 to 13-4

initiators

ACTIVE 9-3

DRAINED 9-3

HALTED 9-3

inactive 9-3

INPUT PARAMETERS field, documentation  
box 21-5

Invocation, Job Scheduling Solution 7-1

INVOKED BY field, documentation box 21-5

IOS071I 9-4, 9-5

## J

Java 19-2

JES2 2-8

JES2 Monitoring Solution

EXECs 3-3

invoking 3-2

processing flow 3-3

Rules 3-4

variables 3-2

JES2 Solution

*See* JES2 Monitoring Solution

Job Scheduling Solution 7-1 to 7-2

*See also* CONTROLM Solution

JOB CPU monitor, Sample Monitoring Solution  
21-10

---

## Jobs

- processing 7-2
- scheduling 7-1

## K

- K Q command 5-4

## L

- LBGCLAS variable 9-2, 9-3
- LBGCPUH variable 9-2
- LBGCPUL variable 9-2
- LBGIBEG variable 9-2
- LBGIDEC variable 9-2, 9-3
- LBGIEND variable 9-2
- LBGIINC variable 9-2
- LBGPAGE variable 9-2
- LBSTART 9-3
- Load Balancing Solution
  - EXECs 9-3
  - invoking 9-2
  - processing flow 9-2
  - Rules 9-3
  - variables 9-2
- LOGREC Data Set Monitoring Solution 5-8
- LOGREC variable 5-8

## M

- MAINVIEW Alarm Manager Solution
  - ALRMSTOP 21-16
  - ALRMSTRT 21-16
  - disabling 21-16
  - enabling 21-15
- masking panel displays 2-43
- MAX\_AGE variable 15-2
- MAX\_EITHER variable 15-2
- MAX\_EXTS variable 15-8
- MAX\_EXTSn variable 15-8
- MAX\_OWNER variable 15-2
- MAX\_REQ variable 15-2
- MAX\_UTIL variable 15-8
- MAX\_UTILn variable 15-8
- MAXACSA variable 9-5
- MAXBCSA variable 9-5

- MAXCCPU variable 9-5
- MAXDEVU variable 9-5
- MAXTCPU variable 9-5
- MDEV command, Sample Monitoring Solution 21-10
- MJSU001C 7-2
- MJSU002C 7-2
- MJSU003C 7-2
- MJSU004C 7-2
- MJSU005C 7-2
- MJSU006C 7-2
- MJSU007C 7-2
- MJSU008C 7-2
- MMN001C 5-7
- MMN002C, Sample Monitoring Solution 21-11
- MMN003C, Sample Monitoring Solution 21-11
- MMN004C 5-7
- MONITOR command 11-9
- Monitoring Solutions 5-1 to 5-10
  - See also* Dump Data Sets Monitoring Solution, LOGREC Data Set Monitoring Solution, RMF Monitoring Solution, SMF Data Set Monitoring Solution, WTO Buffers Monitoring Solution
- MPE001C 9-5
- MPE002C 9-5
- MPE003C 9-3
- MPE004C 9-3
- MPE005C 9-3
- MPE006C 9-3
- MPP 19-2
- MSU002C 7-2
- MSU005C 7-2
- MV Solution Samples
  - CICSplex Sample Solution 21-12
  - Info/System Management Sample Solution 21-14 to 21-15
  - Monitoring Sample Solution 21-9 to 21-11
  - VTAM/NCP Sample Solution 21-6 to 21-8
- MVS 2-8
- MVS Performance Management Solutions
  - See* DASD Reserve Analysis Solution, Exception Monitoring Solution, Load Balancing Solution

---

## MVS Solutions

- Access Methods Solution
    - See* VSAM Failures Solution
  - JES2 Solution 3-1 to 3-4
    - See also* JES2 Monitoring Solution
  - Job Scheduling Solution 7-1 to 7-2
    - See also* CONTROLM Solution
  - Monitoring Solutions 5-1 to 5-10
    - See also* Dump Data Sets Monitoring Solution, LOGREC Data Set Monitoring Solution, RMF Monitoring Solution, SMF Data Set Monitoring Solution, WTO Buffers Monitoring Solution
  - Performance Management Solutions 9-1 to 9-8
  - Storage Subsystems Solutions 4-1
    - See also* Shared DASD Control Solution, Storage Reply Solution, TLMS Solution
  - VTAM/NCP Solution 12-1
    - See also* TCAS Reply Solution
- MVSCALL variable 3-2

## N

- Name field, documentation box 21-5
- NAMEn 15-8
- naming convention
  - exceptions to 21-3
  - for Sample Solutions EXECs 21-1 to 21-3
  - national characters 1-5, 21-6
- national characters 1-5, 21-6
- NET 2-8
- NETACT 21-8
- NETBOUNC 21-8
- NETINACT 21-8
- normal objects, CSM
  - adding parents 2-71
  - adding servers 2-72
  - description 2-5
- NUM 15-8

## O

- object relationships, CSM 2-6
- objects, CSM
  - bouncing 2-116
  - canceling 2-122
  - description 2-14
  - moving 2-125
  - resetting 2-119
  - starting 2-110
  - stopping 2-114
- ONLALMN variable 10-2
- ONLALTN variable 10-2
- ONLCNTN variable 10-2
- ONLSYSN variable 10-2
- ONLTPN variable 10-2
- Operations Monitors Sample Solution
  - EXECs 21-11
  - invoking 21-9
  - processing flow 21-10
- operator-initiated EXECs 21-3
- OSPI application 21-14
- OSPINFO1, Info/System Management Solution 21-15
- OUTBOARD CALLS field, documentation box 21-5
- OUTPUT field, documentation box 21-5
- overriding CSM schedules 2-106, 2-180

## P

- page threshold counters 9-3
- parent-child relationship, CSM 2-7
- parents, CSM
  - adding 2-79
  - deleting 2-83
- Performance Management Solutions
  - CICS 11-1 to 11-9
  - MVS 9-1 to 9-8
- PM00101 7-2
- processing flow 7-2
- PROGRESS command, Sample Monitoring Solution 21-10
- PWSCPU01 9-7
- PWSCPU02 9-7, 9-8
- PWSCSA01 9-7, 9-8
- PWSDEV01 9-7, 9-8

---

## R

RECNUM variable, Info/System Management Solution 21-14

repeat, CSM, line commands 2-76

REPLIES command 4-3

RESERVE command 9-4

resetting objects, CSM 2-119

RESOLVE AEW messages 9-5

RESOLVE commands

- REPLIES 4-3
- RESERVE 9-4

resources, inactivating and reactivating 21-7

response time 11-1

RETURN CODES field, documentation box 21-6

RMF Monitoring Solution 5-10

Rule Processor application

- BBPARM 21-1
- IEF450I 21-15

Rules

- \$HASP050 3-4
- \$HASP093 3-4
- \$HASP095 3-4
- \$HASP355 3-4
- \$HASP646 3-4
- activate 13-2
- DFH0901 10-6
- DFH1311 10-10
- DSNT375I 15-7
- DSNT376I 15-7
- DSNT501X 15-7
- ERB102I 5-10
- IDC3009I 6-2
- IDC3351I 6-2
- IEA404A 5-5
- IEA405E 5-5
- IEA911E 5-3
- IEA994A 5-3
- IEA994E 5-3
- IEC701D 4-5
- IEE361I 5-7
- IEE362A 5-7
- IEE364I 5-7
- IEE366I 5-7
- IEE391I 5-7
- IEE392I 5-7
- IEE393I 5-7

## Rules (*continued*)

- IEE949I 5-7
- IEE974I 5-7
- IEF238D 4-5
- IFB040I 5-9
- IFB060E 5-9
- IFB070I 5-9
- IKT001D 12-2
- IKT010D 12-2
- IKT012D 12-2
- IOS071I 9-5
- PWSCPU02 9-8
- PWSCSA01 9-8
- PWSDEV01 9-8
- update 13-2

## S

Sample Monitoring Solution 21-9

SCHEDCOM 8-2

schedules

- See also* CONTROL-M Solution

schedules, CSM

- conflicts 2-104
- creating 2-93
- description 2-11
- determining 2-93
- exceptional 2-12
- modifying 2-96, 2-100
- overriding 2-106
- routine 2-12

scheduling CONTROLM Solution

- EXECs 8-2
- invoking 8-1
- processing flow 8-2
- variables 8-1

SEE ALSO field, documentation box 21-5

server, CSM

- adding 2-86
- deleting 2-90

service level objectives 11-9

---

Shared DASD Control Solution  
description 4-1  
EXECs 4-2  
invoking 4-2  
processing flow 4-2  
Rules 4-3  
variables 4-2  
shared device 5-5, 9-4  
SMF Data Set Monitoring Solution  
EXECs 5-7, 5-9  
invoking 5-6, 5-8  
processing flow 5-6, 5-9  
Rules 5-7, 5-9  
variables 5-5  
SMFALT variable 5-5  
SMFCLEAR variable 5-5  
solution, MVALARM  
disabling 21-16  
enabling 21-15  
starting CSM objects 2-110  
state, CSM  
actual 2-10  
desired 2-9, 2-10  
STCCPU monitor, Sample Monitoring Solution  
21-10  
stopping CSM objects 2-114  
Storage Reply Solution  
EXECs 4-5  
invoking 4-4  
processing flow 4-4  
Rules 4-5  
variables 4-4  
Storage Subsystems Solutions  
*See* Shared DASD Control Solution, Storage  
Reply Solution, TLMS Solution  
Storage Violation Solution  
EXECs 10-3  
invoking 10-2  
processing flow 10-3  
Rules 10-3  
variables 10-2  
Superuser authority 19-5  
SYS1.LOGREC data set 5-8  
SYS1.MANx 5-6  
SYSBEEP variable 3-2  
SYSN variable 4-2, 9-4

SYSPROG commands  
CPU 21-10  
CSSUM 21-10  
MDEV 21-10  
PROGRESS 21-10  
SYSPROG Services EXECs 21-11  
SYSPROG variable 3-2, 21-14  
system dump data sets 5-1

## T

Tape  
*See* Storage Subsystems Solutions  
TCAS Reply Solution  
EXECs 12-2  
invoking 12-1  
processing flow 12-2  
Rules 12-2  
variables 12-1  
Temporary Storage Data Set Extension Failure  
Solution  
EXECs 10-10  
invoking 10-9  
processing flow 10-9  
Rules 10-10  
variables 10-9  
Terminal Errors Solution  
EXECs 10-11  
invoking 10-11  
processing flow 10-11  
Rules 10-11  
variables 10-10  
TEST SCRIPT INFO field, documentation box  
21-6  
text ID-driven EXECs 21-3  
threshold level 13-3  
TIOTSIO2, Sample Monitoring Solution 21-10  
TLMS Solution  
EXECs 4-4  
invoking 4-3  
processing flow 4-3  
Rules 4-4  
variables 4-3  
TMSINIT 2-8  
TOTCPU monitor  
EXECs 9-7  
Sample Monitoring Solution 21-10

---

TOTSIO monitor, Sample Monitoring Solution  
21-10  
Transaction Response Time Monitor Solution  
EXECs 11-9  
invoking 11-9  
JCL 11-8  
processing flow 11-9  
Rules 11-9  
variables 11-8  
transient object, CSM 2-5  
TSO 2-8  
TSOCPU monitor  
EXECs 9-7  
Sample Monitoring Solution 21-10

## U

UBBPROC data set 21-1  
UPEARLY, CSM object state 2-131  
UPFORCE, CSM object state 2-131  
UPLOGICAL, CSM object state 2-131

## V

variables  
CICS Solution E-1 to E-2  
DB2 Solution F-1 to F-9  
job scheduling 7-1  
MVS Solution D-1 to D-2  
VARIABLES ISSUED field, documentation box  
21-6  
VSAM Control Area Split Monitor Solution  
EXECs 11-8  
invoking 11-7  
JCL 11-7  
processing flow 11-7  
Rules 11-8  
variables 11-7  
VSAM Failures Solution  
EXECs 6-2, 18-1  
invoking 6-2  
processing flow 6-2  
Rules 6-2  
variables 6-1

VSAM Subtask Abnormal Termination Solution  
EXECs 10-6  
invoking 10-6  
processing flow 10-6  
Rules 10-6  
variables 10-5  
VTAM Commands Sample Solution  
inactivating a resource 21-7  
invoking 21-7  
processing flow 21-8  
reactivating a resource 21-8  
Rules 21-8

## W

WATCH monitor, Sample Monitoring Solution  
21-10  
WTO buffer shortages 5-3  
WTO Buffers Monitoring Solution  
EXECs 5-4  
invoking 5-4  
processing flow 5-4  
Rules 5-5  
variables 5-3  
WTORS  
IKT001D 12-2  
IKT010D 12-2  
TCAS 12-1



# END USER LICENSE AGREEMENT NOTICE

**BY OPENING THE PACKAGE, INSTALLING, PRESSING "AGREE" OR "YES" OR USING THE PRODUCT, THE ENTITY OR INDIVIDUAL ENTERING INTO THIS AGREEMENT AGREES TO BE BOUND BY THE FOLLOWING TERMS. IF YOU DO NOT AGREE WITH ANY OF THESE TERMS, DO NOT INSTALL OR USE THE PRODUCT, PROMPTLY RETURN THE PRODUCT TO BMC OR YOUR BMC RESELLER, AND IF YOU ACQUIRED THE LICENSE WITHIN 30 DAYS OF THE DATE OF YOUR ORDER CONTACT BMC OR YOUR BMC RESELLER FOR A REFUND OF LICENSE FEES PAID. IF YOU REJECT THIS AGREEMENT, YOU WILL NOT ACQUIRE ANY LICENSE TO USE THE PRODUCT.**

This Agreement ("**Agreement**") is between the entity or individual entering into this Agreement ("You") and BMC Software Distribution, Inc., a Delaware corporation located at 2101 CityWest Blvd., Houston, Texas, 77042, USA or its affiliated local licensing entity ("BMC"). "You" includes you and your Affiliates. "Affiliate" is defined as an entity which controls, is controlled by or shares common control with a party. IF MORE THAN ONE LICENSE AGREEMENT COULD APPLY TO THE PRODUCT, THE FOLLOWING ORDER OF LICENSE AGREEMENT PRECEDENCE APPLIES: (1) WEB BASED LICENSE AGREEMENT WITH BMC, (2) WRITTEN LICENSE AGREEMENT WITH BMC, (3) SHRINK-WRAP LICENSE AGREEMENT WITH BMC PROVIDED WITH THE PRODUCT, AND (4) THIS ELECTRONIC LICENSE AGREEMENT WITH BMC. In addition to the restrictions imposed under this Agreement, any other usage restrictions contained in the Product installation instructions or release notes shall apply to Your use of the Product.

**PRODUCT AND CAPACITY.** "**Software**" means the object code version of the computer programs provided, via delivery or electronic transmission, to You. Software includes computer files, enhancements, maintenance modifications, upgrades, updates, bug fixes, and error corrections.

"**Documentation**" means all written or graphical material provided by BMC in any medium, including any technical specifications, relating to the functionality or operation of the Software.

"**Product**" means the Software and Documentation.

"**License Capacity**" means the licensed capacity for the Software with the pricing and other license defining terms, including capacity restrictions, such as tier limit, total allowed users, gigabyte limit, quantity of Software, and/or other capacity limitations regarding the Software. For licenses based on the power of a computer, You agree to use BMC's current computer classification scheme, which is available at <http://www.bmc.com> or can be provided to You upon request.

**ACCEPTANCE.** The Product is deemed accepted by You, on the date that You received the Product from BMC.

**LICENSE.** Subject to the terms of this Agreement, as well as Your payment of applicable fees, BMC grants You a non-exclusive, non-transferable, perpetual (unless a term license is provided on an order) license for each copy of the Software, up to the License Capacity, to do the following:

- (a) install the Software on Your owned or leased hardware located at a facility owned or controlled by You in the country where You acquired the license;
- (b) operate the Software solely for processing Your own data in Your business operations; and
- (c) make one copy of the Software for backup and archival purposes only (collectively a "**License**").

If the Software is designed by BMC to permit you to modify such Software, then you agree to only use such modifications or new software programs for Your internal purposes or otherwise consistent with the License. BMC grants You a license to use the Documentation solely for Your internal use in Your operations.

**LICENSE UPGRADES.** You may expand the scope of the License Capacity only pursuant to a separate agreement with BMC for such expanded usage and Your payment of applicable fees. There is no additional warranty period or free support period for license upgrades.

**RESTRICTIONS:** You agree to **NOT**:

- (a) disassemble, reverse engineer, decompile or otherwise attempt to derive any Software from executable code;
- (b) distribute or provide the Software to any third party (including without limitation, use in a service bureau, outsourcing environment, or processing the data of third parties, or for rental, lease, or sublicense); or
- (c) provide a third party with the results of any functional evaluation or benchmarking or performance tests, without BMC's prior written approval, unless prohibited by local law.

**TRIAL LICENSE.** If, as part of the ordering process, the Product is provided on a trial basis, then these terms apply: (i) this license consists solely of a non-exclusive, non-transferable evaluation license to operate the Software for the period of time specified from BMC or, if not specified, a 30 day time period ("**Trial Period**") only for evaluating whether You desire to acquire a capacity-based license to the Product for a fee; and (ii) Your use of the Product is on an AS IS basis without any warranty, and **BMC, ITS AFFILIATES AND RESELLERS, AND LICENSORS DISCLAIM ANY AND ALL WARRANTIES (INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT) AND HAVE NO LIABILITY WHATSOEVER RESULTING FROM THE USE OF THIS PRODUCT UNDER THIS TRIAL LICENSE ("Trial License").** BMC may terminate for its convenience a Trial License upon notice to You. When the Trial Period ends, Your right to use this Product automatically expires. If You want to continue Your use of the Product beyond the Trial Period, contact BMC to acquire a capacity-based license to the Product for a fee.

**TERMINATION.** This Agreement shall immediately terminate if You breach any of its terms. Upon termination, for any reason, You must uninstall the Software, and either certify the destruction of the Product or return it to BMC.

**OWNERSHIP OF THE PRODUCT.** BMC or its Affiliates or licensors retain all right, title and interest to and in the BMC Product and all intellectual property, informational, industrial property and proprietary rights therein. BMC neither grants nor otherwise transfers any rights of ownership in the BMC Product to You. Products are protected by applicable copyright, trade secret, and industrial and intellectual property laws. BMC reserves any rights not expressly granted to You herein.

**CONFIDENTIAL AND PROPRIETARY INFORMATION.** The Products are and contain valuable confidential information of BMC (“**Confidential Information**”). Confidential Information means non-public technical and non-technical information relating to the Products and Support, including, without limitation, trade secret and proprietary information, and the structure and organization of the Software. You may not disclose the Confidential Information to third parties. You agree to use all reasonable efforts to prevent the unauthorized use, copying, publication or dissemination of the Product.

**WARRANTY.** Except for a Trial License, BMC warrants that the Software will perform in substantial accordance with the Documentation for a period of one year from the date of the order. This warranty shall not apply to any problems caused by software or hardware not supplied by BMC or to any misuse of the Software.

**EXCLUSIVE REMEDY.** BMC’s entire liability, and Your exclusive remedy, for any defect in the Software during the warranty period or breach of the warranty above shall be limited to the following: BMC shall use reasonable efforts to remedy defects covered by the warranty or replace the defective Software within a reasonable period of time, or if BMC cannot remedy or replace such defective copy of the Software, then BMC shall refund the amount paid by You for the License for that Software. BMC’s obligations in this section are conditioned upon Your providing BMC prompt access to the affected Software and full cooperation in resolving the claim.

**DISCLAIMER. EXCEPT FOR THE EXPRESS WARRANTIES ABOVE, THE PRODUCT IS PROVIDED “AS IS.” BMC, ITS AFFILIATES AND LICENSORS SPECIFICALLY DISCLAIM ALL OTHER WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. BMC DOES NOT WARRANT THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR FREE, OR THAT ALL DEFECTS CAN BE CORRECTED.**

**DISCLAIMER OF DAMAGES. IN NO EVENT IS BMC, ITS AFFILIATES OR LICENSORS LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES RELATING TO OR ARISING OUT OF THIS AGREEMENT, SUPPORT, AND/OR THE PRODUCT (INCLUDING, WITHOUT LIMITATION, LOST PROFITS, LOST COMPUTER USAGE TIME, AND DAMAGE OR LOSS OF USE OF DATA), EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, AND IRRESPECTIVE OF ANY NEGLIGENCE OF BMC OR WHETHER SUCH DAMAGES RESULT FROM A CLAIM ARISING UNDER TORT OR CONTRACT LAW.**

**LIMITS ON LIABILITY. BMC’S AGGREGATE LIABILITY FOR DAMAGES IS LIMITED TO THE AMOUNT PAID BY YOU FOR THE LICENSE TO THE PRODUCT.**

**SUPPORT.** If Your order includes support for the Software, then BMC agrees to provide support (24 hours a day/7 days a week) (“**Support**”). You will be automatically re-enrolled in Support on an annual basis unless BMC receives notice of termination from You as provided below. There is a free support period during the one year warranty period.

(a) **Support Terms.** BMC agrees to make commercially reasonable efforts to provide the following Support: (i) For malfunctions of supported versions of the Software, BMC provides bug fixes, patches or workarounds in order to cause that copy of the Software to operate in substantial conformity with its then-current operating specifications; and (ii) BMC provides new releases or versions, so long as such new releases or versions are furnished by BMC to all other enrolled Support customers without additional charge. BMC may refuse to provide Support for any versions or releases of the Software other than the most recent version or release of such Software made available by BMC. Either party may terminate Your enrollment in Support upon providing notice to the other at least 30 days prior to the next applicable Support anniversary date. If You re-enroll in Support, BMC may charge You a reinstatement fee of 1.5 times what You would have paid if You were enrolled in Support during that time period.

(b) **Fees.** The annual fee for Support is 20% of the Software’s list price less the applicable discount or a flat capacity based annual fee. BMC may change its prices for the Software and/or Support upon at least 30 days notice prior to Your support anniversary date.

**VERIFICATION.** If requested by BMC, You agree to deliver to BMC periodic written reports, whether generated manually or electronically, detailing Your use of the Software in accordance with this Agreement, including, without limitation, the License Capacity. BMC may, at its expense, perform an audit, at your facilities, of Your use of the Software to confirm Your compliance with the Agreement. If an audit reveals that You have underpaid fees, You agree to pay such underpaid fees. If the underpaid fees exceed 5% of the fees paid, then You agree to also pay BMC’s reasonable costs of conducting the audit.

**EXPORT CONTROLS.** You agree not to import, export, re-export, or transfer, directly or indirectly, any part of the Product or any underlying information or technology except in full compliance with all United States, foreign and other applicable laws and regulations.

**GOVERNING LAW.** This Agreement is governed by the substantive laws in force, without regard to conflict of laws principles: (a) in the State of New York, if you acquired the License in the United States, Puerto Rico, or any country in Central or South America; (b) in the Province of Ontario, if you acquired the License in Canada (subsections (a) and (b) collectively referred to as the “**Americas Region**”); (c) in Singapore, if you acquired the License in Japan, South Korea, Peoples Republic of China, Special Administrative Region of Hong Kong, Republic of China, Philippines, Indonesia, Malaysia, Singapore, India, Australia, New Zealand, or Thailand (collectively, “**Asia Pacific Region**”); or (d) in the Netherlands, if you acquired the License in any other country not described above. The United Nations Convention on Contracts for the International Sale of Goods is specifically disclaimed in its entirety.

**ARBITRATION. ANY DISPUTE BETWEEN YOU AND BMC ARISING OUT OF THIS AGREEMENT OR THE BREACH OR ALLEGED BREACH, SHALL BE DETERMINED BY BINDING ARBITRATION CONDUCTED IN ENGLISH. IF THE DISPUTE IS INITIATED IN THE AMERICAS REGION, THE ARBITRATION SHALL BE HELD IN NEW YORK, U.S.A., UNDER THE CURRENT COMMERCIAL OR INTERNATIONAL, AS APPLICABLE, RULES OF THE AMERICAN ARBITRATION ASSOCIATION. IF THE DISPUTE IS INITIATED IN A COUNTRY IN THE ASIA PACIFIC REGION, THE ARBITRATION SHALL BE HELD IN SINGAPORE, SINGAPORE UNDER THE CURRENT UNCITRAL ARBITRATION RULES. IF THE DISPUTE IS INITIATED IN A COUNTRY OUTSIDE OF THE AMERICAS REGION OR ASIA PACIFIC REGION, THE ARBITRATION SHALL BE HELD IN AMSTERDAM, NETHERLANDS UNDER THE CURRENT UNCITRAL ARBITRATION RULES. THE COSTS OF THE ARBITRATION SHALL BE BORNE EQUALLY PENDING THE ARBITRATOR’S AWARD. THE AWARD RENDERED SHALL BE FINAL AND BINDING UPON THE PARTIES AND SHALL NOT BE SUBJECT TO APPEAL TO ANY COURT, AND MAY BE ENFORCED IN ANY COURT OF COMPETENT JURISDICTION. NOTHING IN THIS AGREEMENT SHALL BE DEEMED AS PREVENTING EITHER PARTY FROM SEEKING INJUNCTIVE RELIEF FROM ANY COURT HAVING JURISDICTION OVER THE PARTIES AND THE SUBJECT MATTER OF**

**THE DISPUTE AS NECESSARY TO PROTECT EITHER PARTY'S CONFIDENTIAL INFORMATION, OWNERSHIP, OR ANY OTHER PROPRIETARY RIGHTS. ALL ARBITRATION PROCEEDINGS SHALL BE CONDUCTED IN CONFIDENCE, AND THE PARTY PREVAILING IN ARBITRATION SHALL BE ENTITLED TO RECOVER ITS REASONABLE ATTORNEYS' FEES AND NECESSARY COSTS INCURRED RELATED THERETO FROM THE OTHER PARTY.**

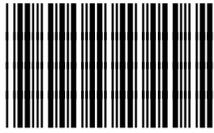
**U.S. GOVERNMENT RESTRICTED RIGHTS.** The Software under this Agreement is "commercial computer software" as that term is described in 48 C.F.R. 252.227-7014(a)(1). If acquired by or on behalf of a civilian agency, the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of this Agreement as specified in 48 C.F.R. 12.212 (Computer Software) and 12.211 (Technical Data) of the Federal Acquisition Regulations ("**FAR**") and its successors. If acquired by or on behalf of any agency within the Department of Defense ("**DOD**"), the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of this Agreement as specified in 48 C.F.R. 227.7202 of the DOD FAR Supplement and its successors.

**MISCELLANEOUS TERMS.** You agree to pay BMC all amounts owed no later than 30 days from the date of the applicable invoice, unless otherwise provided on the order for the License to the Products. You will pay, or reimburse BMC, for taxes of any kind, including sales, use, duty, tariffs, customs, withholding, property, value-added (VAT), and other similar federal, state or local taxes (other than taxes based on BMC's net income) imposed in connection with the Product and/or the Support. This Agreement constitutes the entire agreement between You and BMC and supersedes any prior or contemporaneous negotiations or agreements, whether oral, written or displayed electronically, concerning the Product and related subject matter. No modification or waiver of any provision hereof will be effective unless made in a writing signed by both BMC and You. You may not assign or transfer this Agreement or a License to a third party without BMC's prior written consent. Should any provision of this Agreement be invalid or unenforceable, the remainder of the provisions will remain in effect. The parties have agreed that this Agreement and the documents related thereto be drawn up in the English language. Les parties exigent que la présente convention ainsi que les documents qui s'y rattachent soient rédigés en anglais.

SW Click EULA 071102



# Notes



\*42116\*