MAINVIEW® AutoOPERATOR™

Basic Automation Guide

Version 6.2

March 15, 2002

bmcsoftware

Copyright © 2002 BMC Software, Inc., as an unpublished work. 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 registered trademarks or trademarks belong to their respective companies.

THE USE AND CONTENTS OF THIS DOCUMENTATION ARE GOVERNED BY THE SOFTWARE LICENSE AGREEMENT ENCLOSED AT THE BACK OF THIS DOCUMENTATION.

Restricted Rights Legend

U.S. GOVERNMENT RESTRICTED RIGHTS. UNPUBLISHED–RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in FAR Section 52.227-14 Alt. III (g)(3), FAR Section 52.227-19, DFARS 252.227-7014 (b), or DFARS 227.7202, as amended from time to time. Send any contract notices to Contractor/Manufacturer:

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

Contacting BMC Software

You can access the BMC Software Web site at **http://www.bmc.com**. From this Web site, you can obtain general information about the company, its products, special events, and career opportunities. For a complete list of all BMC Software offices and locations, go to **http://www.bmc.com/corporate/offices.html**.

USA 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 USA 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," below.

Support Web Site

You can obtain technical support from BMC Software 24 hours a day, seven days a week by accessing the technical support Web site at **http://www.bmc.com/support.html**. From this 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 via Telephone or E-mail

In the USA and Canada, if you need technical support and do not have access to the Web, call 800 537 1813. Outside the USA 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**.

Before Contacting BMC Software

Before you contact BMC Software, have the following information available so that a technical support analyst 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 program temporary fix (PTF)
  - system hardware configuration
  - serial numbers
  - related software (database, application, and communication) including type, version, and service pack or PTF
- 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

Part 1. Introducing MAINVIEW AutoOPERATOR 1

Chapter 1. Introduction to MAINVIEW AutoOPERATOR 3
How AutoOPERATOR Applications Are Organized 4
Operator Workstation Applications 4
Automation Applications 4
How to Access AutoOPERATOR 6
How the Documentation Library for AutoOPERATOR Is Organized 8

Chapter 2. Describing Common Application Functions and Display Fields 9
Requesting an AutoOPERATOR Application 9
Using Another Product from AutoOPERATOR: Product Line Transfers 10
Limitations 11
Using Another AutoOPERATOR Application from AutoOPERATOR: Application Transfers 12
Limitations 12
Using Product Line Transfers in Conjunction with Application Transfers 12
List of BBI Transfer Commands and Abbreviations for Application Names 13
Issuing Transfer Commands from CSM 20
Using Command Prefixes 20
Exiting from Product and Application Transfers: Using INIT and RETURN 21
Describing Common Application Fields and Functions 22
Describing the Short-Message Field 22
Describing the Primary Command Field 22
Describing the Target Input Field 22
Describing the INTERVAL Field: Refresh Mode Fields 23
Using Scroll Commands 23
Splitting the BBI Terminal Session Screen 24
Using BBI Control Commands 24
Invoking an AutoOPERATOR EXEC from the COMMAND Line 25
Sending BBI-SS PAS Journal Log Messages 25
Using Program Function (PF) Key Definitions in AutoOPERATOR 26
Assigning Commands to Program Function Keys 27
Assigning Labels to PF Keys 27

Part 2. Reacting to Automation Events Using Rules 29

Chapter 3. Automation with Rules: The Rule Processor Application 31
What the Rule Processor Is 31
What Events Are 32
What a Rule Is 32
Where Rules Are Kept 33
What Actions a Rule Can Take 34
How to Create a Rule 35
How to Begin Identifying Events for Automation 38
How to Manage and Implement Basic Automation with Rules 39
Grouping Rules into Rule Sets 39

What Automation Strategy Is . . . 39
Why Automation Strategy Is Important . . . 40
How to Qualify Displays in the Rule Processor Applications . . . 41
Where to Go from Here: Finding Information Quickly . . . 42

Chapter 4. Describing Events . . . 43
What Events Are . . . 43
Identifying Events for Automation with Rules: Using the Event Activity Statistics Application . . . 46
Accessing the Event Activity Statistics . . . 46
Describing Fields . . . 51
Describing Special Event Types: DB2, EXT, TIME, and VAR . . . 53
DB2 Events . . . 53
EXT Events . . . 53
TIME Events . . . 54
VAR Events . . . 54

Chapter 5. Creating Rules for Events: Using the Rule Processor Detail Control Panel . . . 55
What the Rule Processor Detail Control Panel Is . . . 55
Accessing the Rule Processor Detail Control Panel . . . 57
Describing the Rule Processor Detail Control Panel . . . 64
Using Primary Commands . . . 65
Describing Fields . . . 66
Using the Criteria Match Rate Fields . . . 70
Advanced Rule Processing Techniques . . . 71
Activating and Saving Modified Rules . . . 71

Chapter 6. Creating Rules for Events: Using the Rule Creation Panels . . . 73
Using Primary Commands . . . 73
Step 1. Using Selection Criteria Fields . . . 75
Describing Selection Criteria Fields . . . 76
Step 2. Specifying Variables as Additional Selection Criteria . . . 80
Using AND/OR Logic on the Variable Dependencies Panel . . . 81
Describing Variable Dependencies Fields . . . 82
Step 3. Using Action Specification Fields . . . 85
Describing Action Specification Fields . . . 86
Step 4. Creating ALERTs as Additional Action Specification . . . 93
Creating ALERTs for All Event Types Except ALRT Events . . . 94
Creating ALERTs for the ALRT Event Type . . . 95
Modifying an ALERT in a Rule: Using the MODIFY Function . . . 103

Chapter 7. Creating Rules: Examples . . . 105
Creating a Rule to Suppress a Message . . . 105
Example 1: Creating a Suppression Rule . . . 106
Example 2: Creating a Suppression Rule with Variables . . . 115
Creating TIME-Initiated Rules . . . 121
Example 1: Creating Simple TIME-Initiated Rules . . . 121
Example 2: Creating a TIME-Initiated Rule with ATSTART . . . 126
Creating More Complex Rules . . . 131
Example 1: Setting a Variable with a Rule . . . 131
Example 2: Creating a Rule That Fires When a Variable Value Is Met . . . 136
Using the Rules . . . 141
Creating ALERTs from Rules . . . 142

Example 1: Creating an ALERT Retained across BBI-SS PAS Restarts and MVS IPLs . . . 142
Example 2: Creating an ALERT Whose Priority Increases Over Time . . . 146
Creating a VAR-Initiated Rule . . . 151

Chapter 8. Creating More Powerful Rules . . . 155
Introduction . . . 155
Using Pattern Matching on Selection Criteria Panels . . . 156
Examples of Using Pattern Matching . . . 157
Using Variables in the Rule Processor . . . 161
Defining Variables . . . 161
Resolving Variables . . . 161
Using Variables in Rule Processor Application Fields . . . 164
Variable Modifiers . . . 165
List of AutoOPERATOR-Supplied EVENT Variables . . . 169
List of Variables for ALRT-initiated Rules . . . 174

Chapter 9. Managing Rules and Automation Using the Automation Control Panel . . . 177
Setting Automation Status/Strategy (Upper Portion) . . . 178
Using Primary Commands . . . 179
Viewing Automation Statistics (Middle Portion) . . . 182
Grouping Rules into Rule Sets (Bottom Portion) . . . 183
Using Line Commands . . . 185
Choosing an Automation Strategy . . . 186
Selecting FIRST . . . 186
Selecting ALL . . . 186
Selecting QUALIFIED . . . 187
Selecting INDIVIDUAL . . . 188
Rule Set Filter . . . 188
Filtering Events for Rule Sets . . . 188
Enabling Rule Set Filtering and Rule Set Match Rate . . . 190

Chapter 10. Using the Rule Set Overview Panel to Manage Rules in Rule Sets . . . 197
Using Primary Commands . . . 203
Using Line Commands . . . 203
Copying or Moving Rules . . . 205

Part 3. Managing Critical Applications Using CSM . . . 207

Chapter 11. Managing Critical Applications with Continuous State Manager . . . 211
What Continuous State Manager Is . . . 212
What the Benefits of CSM Are . . . 213
What CSM Requires . . . 213
Defining CSM Terms and Concepts . . . 214
CSM Terms and Concepts . . . 214
What an Object Is . . . 214
What Object Relationships Are . . . 215
What an Object State Is . . . 218
What a Schedule Is . . . 220
What Else CSM Must Know and Why . . . 222
How CSM Starts, Stops, and Cancels Objects . . . 222
How CSM Recognizes when an Object Starts or Stops . . . 222

How CSM Connects and Disconnects Objects. 223
How CSM Works. 224
When CSM Starts 224
Routine Scheduling 225
Recovering Failed Objects 225
Interfering with CSM Processing 226
Managing Objects Across BBI-SS PASs 227
Examples of CSM Processing 229
How CSM Automates the IPL Process. 230
How CSM Automates the Shutdown Process 232
How CSM Manages Routine Object Scheduling 235
How CSM Handles Global Calendar Overrides. 237
How to Effectively Plan for CSM Implementation 239
Determining Object Relationships 240
Planning Schedules. 240
Listing Start, Stop, Cancel and (Dis)Connect Commands 241
Choosing Start and Stop Text-IDs 241
Choosing Text-IDs for Abnormal Termination Events 241
Before You Begin 243
Setting Up the Shared Repository 243
Cross-System Object Management: Using Group 243
Repository Serialization 244
Securing Access to CSM 244
Backing Up the CSM Repository 244
Using CSM Panels 245
Accessing CSM for the First Time 247
How to Qualify (Mask) Panels in CSM 248
What the CSM Global Overview Panel Is 249
Viewing an Object's Documentation 249
Viewing Objects Whose States Do Not Match. 250
Filtering Groups Displayed on Global Overview. 251
Defining CSM Objects. 255
Creating a Group 256
Entering the Object's Information 259
Entering Command Specifications for an Object 262
Additional Fields to Complete 264
Specifying Conditional Command Specifications for an Object 265
Specifying Start and Stop Events: Event Specifications 1 267
Specifying Abnormal Termination Events: Event Specifications 2 268
Specifying User Notification 270
Associating an Object with a Group. 272
Moving an Object Between Groups 273
Defining the Object's Parents 274
Defining the Object's Servers. 275
Defining a Transient Object 278
Defining a Grouping Object 278
Comparing CSM Object Types. 278
Using the Repeat Line Command to Create Objects 279
Modifying Object Definitions 282
Adding Parents 282
Deleting Parents 286
Adding Servers 288
Deleting Servers 291
Modifying an Object's Routine Schedule 294

Determining the Schedule of the Object's Parent 294
Modifying the Object's Schedule 296
Modifying an Object's Schedule with CSM Schedule Tags 299
Dealing with Scheduling Conflicts at Modification 304
Overriding the Calendar for Special Situations 306
Enabling Cross-System Object Management 308
Routine Object Tasks 310
Starting an Object 311
Stopping an Object 314
Bouncing an Object 316
Resetting an Object 318
Canceling an Object 321
Moving an Object: Using the Moveable Attribute 323
Scheduling an EXEC Associated with a CSM ALERT 326
Displaying and Modifying Object States 327
What the Possible Desired States Are 327
What the Possible Actual States Are 330
Monitoring CSM Object States Manually 330
Using the Command Line Interface 335
Describing CSM Panels 338
CSM Global Overview Panel 338
Defining and Modifying Object Panels 349
Object Detail Control Panel 349
Command Specifications Panel 353
Conditional Command Specifications Panel 359
Event Specifications 1 Panel 361
Event Specifications 2 Panel 365
User Notification Panel 367
Object Groups Panel 369
Object Requisites Panel 371
Client/Server Relationships Panel 373
Object Scheduling Panel 376
Global Calendar Override Panel 379
Object Groups Panel 383
Group Detail Control 385

Part 4. Using Other Basic Automation Applications 389

Chapter 12. Managing by Exception: Using AutoOPERATOR ALERTs 391
Introduction 391
Overview to the ALERT Management Facility 393
What ALERTs Are 393
How ALERTs Are Kept 394
Where ALERTs Are Stored 394
How to Display ALERTs 394
What ALERTs Can Do 394
Where to Go from Here 395
Obtaining Statistical ALERT Information: ALERT Queue Statistics Application 396
Panel Description 397
Displaying ALERT Data Graphically: ALERT Overview Application 399
Using a Profile 402
Showing ALERTs in Text Format: Using the ALERT Detail Application 404
Using a Profile 408

Chapter 13. Monitoring Automation with the Automation Reporter 415
Introduction: What the Automation Reporter Application Does. 415
How to Activate the Automation Reporter 416
Activating the Automation Reporter with the Dynamic Parameter Manager 417
Step 1: Accessing the Automation Reporter Parms Panel in the DPM Application . . . 417
Step 2: Allocate a Data Set. 419
Step 3: Specifying the Data Set Disposition. 420
Step 4: Specifying Collection Intervals 421
Step 5: Specifying the Offload Interval 422
Step 6: Specifying an AutoOPERATOR ALERT Queue Name. 423
Step 7: Activating the Automation Reporter 424
Stopping the Automation Reporter. 426
Creating Automation Reporter Reports. 427
What Kind of Data Is Collected 427
How the Data Is Saved 429
What the Record Looks Like: Basic Data Record Design 431
Describing a Sample Data Record 434
What Record Types Are Available 435
Tracking Actions (ACTN) 437
Tracking AutoOPERATOR ALERTS (ALRT) 443
Tracking Events (EVENT) 445
Tracking AutoOPERATOR EXECs (EXEC). 446
Tracking AutoOPERATOR Rules Activity (RULE) 447
Tracking AutoOPERATOR Rule Sets (RULESET: AUTO) 448
Tracking AutoOPERATOR Rule Sets (RULESET: SET) 449
Tracking Resources (RES) 451
Tracking Tape Device Sharing Activity (TAPE: LOCAL). 453
Tracking Tape Device Sharing Activity (TAPE: PLEX) 455

Chapter 14. Managing EXECs Using the EXEC Management Application. 457
Introduction 457
Accessing the EXEC Management Application 458
Scrolling Right to Display Additional Information 459
Viewing All Currently Active EXECs: Using the EXPAND Command 459
Using Primary Commands 460
Describing Fields in the EXEC Status Area. 460
Describing Fields in the EXEC Description Area 462
Displaying Currently Active EXECs. 466
Using Line Commands. 466
Describing Fields 467
Documenting Your EXECs 468

Chapter 15. General Services 469
Creating Cyclic Displays: CYCLE SETUP. 469
Specifying Application Names in CYCLE SETUP 471
Saving a Refresh Cycle. 477
Viewing Current Operational Events: Log Display. 480
Displaying System Messages: MESSAGES 483
Using PF Keys: KEYS. 483
Requesting Time-Initiated EXECs: TIMEXEC Application. 484
Describing the TIMEXEC Application 485
Purging an Active CALLX Request. 489

Part 5. Using AutoOPERATOR Sample Solutions 491

Chapter 16. Introduction to Using Sample Solutions 493
Distributed Sample Solutions 493
Naming Conventions for EXECs 494
Standards for Sample Solutions 497

Chapter 17. Sample VTAM/NCP Solution 499
Sample VTAM Commands Solution 499
Setting Variables for the Sample VTAM Commands Solution 499
Invoking the Sample VTAM Commands Solution 499
Processing Flow for the Sample VTAM Commands Solution 500
EXECs for the Sample VTAM Commands Solution 500
Rules for the Sample VTAM Commands Solution 500

Chapter 18. Sample Monitoring Solution 501
Sample Operations Monitors Solution 501
Setting Variables for the Sample Operations Monitors Solution 501
Invoking the Sample Operations Monitors Solution 501
Processing Flow for the Sample Operations Monitors Solution 501
EXECs for the Sample Operations Monitors Solution 503
Rules for the Sample Operations Monitors Solution 503

Chapter 19. Sample CICSPlex Solution 505
Invoking the Sample CICSPlex Solution 505
EXECs 506
Rules 506

Chapter 20. Sample Info/System Management Solution 507
Invoking the Sample Info/System Management Solution 507
EXECs for the Sample Info/System Management Solution 507
Rules for the Sample Info/System Management Solution 508

Chapter 21. Activating the MVALARM Solution 509
Enabling the MVALARM Solution 509
Stopping the MVALARM Solution 510

Part 6. Appendixes 511

Appendix A. Diagnosing CSM Errors 513
Creating a Cross-Reference Report 513
Gathering Environmental Information 514
Generating CSM Exec Diagnostics for Support 514
Tracing Starts and Stops of CSM EXECs 516
Tracing All Activity in a CSM EXEC 516
Stopping a Trace 517
Creating CSM Start and Stop Rules 518
Creating CSM Start and Stop Rules for the APPC Object 518
Creating CSM Start and Stop Rules for the TCPIP Object 518
Common Questions and Answers 520
Starting/Stopping Errors 520
Restart-Only Objects 521

Defining Special Objects 521
CSMACT Usage 524
BBI Terminal Sessions (BBI-TS) 525
TSO/E or Environmental Problems 525
CSM Installation 526
Effective Implementation 526
Other Tips and Questions 527

Appendix B. MVS Sample Solutions 531

Appendix C. DBCTL Commands 533
Issuing Commands in a DBCTL Environment 533
Issuing DBCTL Commands from a Terminal Session 533
Issuing DBCTL Commands from an EXEC 534

Appendix D. Describing Rule Selection Criteria and Action Specifications 535
Selecting Criteria for ALRM-Initiated Rules 536
Specifying Actions for ALRM-Initiated Rules 538
Selecting Criteria for ALRT-Initiated Rules 541
Specifying Actions for ALRT-Initiated Rules 543
Selecting Criteria for CICS-Initiated Rules 546
Specifying Actions for CICS-Initiated Rules 548
Selecting Criteria for CMD-Initiated Rules 551
Specifying Actions for CMD-Initiated Rules 553
Selecting Criteria for DB2-Initiated Rules 556
Specifying Actions for DB2-Initiated Rules 558
Selecting Criteria for EXT-Initiated Rules 562
Specifying Actions for EXT-Initiated Rules 564
Selecting Criteria for IMS-Initiated Rules 567
Specifying Actions for IMS-Initiated Rules 569
Selecting Criteria for JES3-Initiated Rules 572
Specifying Actions for JES3-Initiated Rules 574
Selecting Criteria for JRNL-Initiated Rules 578
Specifying Actions for JRNL-Initiated Rules 579
Selecting Criteria for MSG-Initiated Rules 583
Specifying Actions for MSG-Initiated Rules 586
Selecting Criteria for Time-Initiated Rules 590
Examples 591
Specifying Actions for Time-Initiated Rules 593
Selecting Criteria for VAR-Initiated Rules 596
Specifying Actions for VAR-Initiated Rules 597

Part 7. Glossary and Index 601

Glossary 603

Index 615

Figures

1. AutoOPERATOR Applications Overview . . . 3
2. Example of MAINVIEW Selection Menu . . . 6
3. Example of AutoOPERATOR Primary Option Menu . . . 7
4. AutoOPERATOR Automation Menu . . . 7
5. Example 1 of Product Line Transfer . . . 10
6. Example 2 of Product Line Transfer . . . 10
7. Example of Stacking Product Line and Application Transfers . . . 12
8. Program Function Keys Panel . . . 26
9. PF Keys Label Display . . . 27
10. Automating Events . . . 33
11. Overview of Rules Panels . . . 35
12. Automation Menu . . . 46
13. Event Activity Statistics Panel . . . 47
14. Overview of Panels Required to Create a Rule . . . 56
15. Selecting an Event . . . 57
16. Rule Processor Detail Control Panel . . . 58
17. Selecting Option 2, Display/Modify Rules and Rule Sets . . . 58
18. Automation Control Panel . . . 59
19. Automation Control Panel: Creating a New Rule . . . 60
20. Adding a New Rule Set . . . 61
21. Rule Processor Detail Control Panel . . . 61
22. Rule Processor Detail Control Panel: Displaying Available Event Types . . . 62
23. Rule Processor Event Types Panel . . . 63
24. Top Portion: Rule Processor Detail Control Panel . . . 66
25. Middle Portion: Rule Processor Detail Control Panel . . . 67
26. Bottom Portion: Rule Processor Detail Control Panel . . . 68
27. Example of Confirm Rule Set Panel . . . 72
28. Variable Dependencies Panel . . . 80
29. Variable Dependencies Panel with AND/OR Logic . . . 81
30. ALERT Action(s) I: Primary Panel for All Events Except ALRT . . . 94
31. ALERT Action(s) Specification Panel for the ALRT Event Type . . . 95
32. ALERT Action(s) II: Secondary ALERT Creation Panel . . . 96
33. Creating a Suppression Rule (1): Automation Control Panel (Example 1) . . . 106
34. Creating a Suppression Rule (1): Automation Control Panel (Example 2) . . . 107
35. Creating a Suppression Rule (1): Rule Set Overview Panel (Example 1) . . . 107
36. Creating a Suppression Rule (1): Enter Rule Information on the Rule Processor Detail Control Panel . . . 108
37. Creating a Suppression Rule (1): Selection Criteria Panel . . . 109
38. Creating a Suppression Rule (1): Variable Dependencies Panel . . . 110
39. Creating a Suppression Rule (1): Action Specification Panel . . . 111
40. Creating a Suppression Rule (1): Alert Action(s) I Panel . . . 112
41. Creating a Suppression Rule (1): Alert Action(s) II Panel . . . 112
42. Creating a Suppression Rule (1): Rule Set Overview Panel (Example 2) . . . 113
43. Creating a Suppression Rule (1): Confirming Rule Set Modifications Panel . . . 114
44. Creating a Suppression Rule (2): Enter Rule Information on the Rule Processor Detail Control Panel . . . 115
45. Creating a Suppression Rule (2): Selection Criteria Panel . . . 116
46. Creating a Suppression Rule (2): Variable Dependencies Panel . . . 117
47. Creating a Suppression Rule (2): Action Specification Panel . . . 118
48. Creating a Suppression Rule (2): Alert Action(s) I Panel . . . 119
49. Creating a Suppression Rule (2): Alert Action(s) II Panel . . . 119

50. Creating a Suppression Rule (2): Rule Processor Detail Control Panel 120
51. Creating a TIME-Initiated Rule (1): Selection Criteria Panel 122
52. Creating a TIME-Initiated Rule (1): Variable Dependencies Panel 123
53. Creating a TIME-Initiated Rule (1): Action Specification Panel 123
54. Creating a TIME-Initiated Rule (1): Alert Action(s) I Panel 124
55. Creating a TIME-Initiated Rule (1): Alert Action(s) II Panel 124
56. Creating a TIME-Initiated Rule (1): Rule Processor Detail Control Panel 125
57. Creating a TIME-Initiated Rule (2): Rule Processor Detail Control Panel (Example 1). 126
58. Creating a TIME-Initiated Rule (2): Selection Criteria Panel 126
59. Creating a TIME-Initiated Rule (2): Variable Dependencies Panel 127
60. Creating a TIME-Initiated Rule (2): Action Specification Panel 128
61. Creating a TIME-Initiated Rule (2): Alert Action(s) I Panel 129
62. Creating a TIME-Initiated Rule (2): Alert Action(s) II Panel 129
63. Creating a TIME-Initiated Rule (2): Rule Processor Detail Control Panel (Example 2) 130
64. Setting a Variable with a Rule: Rule Processor Detail Control Panel (Example 1) 131
65. Setting a Variable with a Rule: Selection Criteria Panel. 132
66. Setting a Variable with a Rule: Variable Dependencies Panel 132
67. Setting a Variable with a Rule: Action Specification Panel 133
68. Setting a Variable with a Rule: Alert Action(s) I Panel 134
69. Setting a Variable with a Rule: Alert Action(s) II Panel. 134
70. Setting a Variable with a Rule: Rule Processor Detail Control Panel (Example 2) 135
71. Triggering a Rule with Variables: Rule Processor Detail Control Panel (Example 1) 136
72. Triggering a Rule with Variables: Selection Criteria Panel 137
73. Triggering a Rule with Variables: Variable Dependencies Panel 137
74. Triggering a Rule with Variables: Action Specification Panel 138
75. Triggering a Rule with Variables: Alert Action(s) I Panel 139
76. Triggering a Rule with Variables: Alert Action(s) II Panel 140
77. Triggering a Rule with Variables: Rule Processor Detail Control Panel (Example 2) 140
78. Modifying an ALERT to Survive MVS IPLs: Rule Processor Detail Control Panel (Example 1) 142
79. Modifying an ALERT to Survive MVS IPLs: Selection Criteria Panel 143
80. Modifying an ALERT to Survive MVS IPLs: Variable Dependencies Panel. 143
81. Modifying an ALERT to Survive MVS IPLs: Action Specification Panel. 144
82. Modifying an ALERT to Survive MVS IPLs: Alert Action(s) I Panel. 144
83. Modifying an ALERT to Survive MVS IPLs: Alert Action(s) II Panel 145
84. Modifying an ALERT to Survive MVS IPLs: Rule Processor Detail Control Panel (Example 2) 145
85. Escalating an ALERT: Rule Processor Detail Control Panel (Example 1). 146
86. Escalating an ALERT: Selection Criteria Panel 147
87. Escalating an ALERT: Variable Dependencies Panel. 147
88. Escalating an ALERT: Action Specification Panel. 148
89. Escalating an ALERT: Alert Action(s) I Panel. 148
90. Escalating an ALERT: Alert Action(s) II Panel 149
91. Escalating an ALERT: Rule Processor Detail Control Panel (Example 2). 150
92. Creating a VAR-Initiated Rule: Rule Processor Detail Control Panel (Example 1). 151
93. Creating a VAR-Initiated Rule: Selection Criteria Panel 151
94. Creating a VAR-Initiated Rule: Variable Dependencies Panel. 152

95. Creating a VAR-Initiated Rule: Action Specification Panel 153
96. Creating a VAR-Initiated Rule: Alert Action(s) I Panel 153
97. Creating a VAR-Initiated Rule: Alert Action(s) II Panel. 154
98. Creating a VAR-Initiated Rule: Rule Processor Detail Control Panel (Example 2) 154
99. Automation Control Panel 177
100. Automation Control Panel: Upper Portion 178
101. Automation Control Panel: Middle Portion. 182
102. Automation Control Panel: Bottom Portion 183
103. Automation Control Panel: Rule Set Filtering. 189
104. Automation Control Panel: Viewing a Rule Set Filter 190
105. Filter Criteria Panel: Viewing the Filter Crtieria 191
106. Variable Dependencies - Filter Panel. 193
107. Automation Control Panel: Viewing a Rule Set's Rules 194
108. Rule Set Overview Panel: Viewing a Rule Set's Rules 194
109. Rule Set Overview Panel—View 1 197
110. Rule Set Overview Panel—View 2 199
111. Rule Set Overview Panel—View 3 200
112. Rule Set Overview Panel—View 4 201
113. Rule Set Overview Panel—View 5 202
114. CSM Relationships 216
115. Example of IPL 230
116. Example of Shutdown 232
117. Example of Shutdown and IPL 234
118. Example of Scheduled Down Time 236
119. Example of Global Calendar Override: Add Down Time 237
120. Example of Global Calendar Override: Extend Down Time. 238
121. Sample JCL to Back Up CSM Shared Repository 245
122. Object Detail Control Panel. 246
123. AutoOPERATOR Automation Menu 247
124. Global Overview Panel 248
125. Global Overview Panel: Scrolled LEFT. 249
126. CSM Exceptions Panel 250
127. Pre-Version 6.2 Display of the Global Overview Panel 251
128. Using the PROFile Command to Display the Group Display Selection - Initial Visit. 252
129. Group Display Selection Panel Showing Groups to Exclude 253
130. Group Display Selection Panel Showing Change in Group Status 253
131. Filtered Global Overview Panel. 254
132. Global Overview Panel: Adding a Group 256
133. Object Groups Panel 256
134. Group Detail Control - Add Panel. 257
135. Group Detail Control - Add Panel (Completed) 258
136. Object Groups Panel: One Group Added 258
137. Object Detail Control Panel: Adding a New Object 260
138. CSM Object Types. 261
139. Command Specifications Panel 262
140. Conditional Command Specifications Panel: Start 265
141. Conditional Command Specifications Panel: Stop 265
142. Event Specifications 1 Panel 267
143. Event Specifications 2 Panel 268
144. User Notification Panel 270
145. Object Groups Panel 272
146. Object Groups Panel 272

147. Object Groups Panel: Creating a Moveable Object 273
148. Object Requisites for AAOCSM01 Panel. 274
149. Add Object Requisites for AAOCSM01 Panel. 274
150. Client/Server Relationships for AAOCSM01 Panel. 275
151. Add Client/Server Relationships for AAOCSM01 Panel 275
152. Object Scheduling for AAOCSM01 Panel 276
153. Object Detail Control Panel 276
154. CSM Global Overview Panel 277
155. Repeating an Object 279
156. Repeat Object Definition Panel. 280
157. Add New Object Panel: Repeat Line Command Example 2 280
158. Object Detail Control - Repeat 281
159. AAOCSM03 Object Created. 281
160. Global Overview Panel: Adding a Parent 282
161. Adding a Parent - Step 2 283
162. Adding a Parent - Step 3 283
163. Adding a Parent - Step 4 284
164. Adding a Parent - Step 5 284
165. Adding a Parent - Step 6 285
166. Adding a Parent - Step 7 285
167. Global Overview Panel: Deleting a Parent 286
168. Deleting a Parent: Step 2. 286
169. Deleting a Parent: Step 3. 287
170. Deleting a Parent: Step 5. 287
171. Global Overview Panel: Adding a Server. 288
172. Adding a Server - Step 3. 288
173. Adding a Server - Step 4. 289
174. Adding a Server - Step 5. 289
175. Adding a Server - Step 6. 289
176. Global Overview Panel: Server Added. 290
177. Clients of AAOCSM10 Panel 290
178. Global Overview Panel: Deleting a Server 291
179. Deleting a Server - Step 3 291
180. Deleting a Server - Step 4 292
181. Deleting a Server - Step 5 292
182. Global Overview Panel: Server Deleted 293
183. Clients of AAOCSM10 Panel 293
184. Checking the Schedule of a Parent 294
185. Checking the Schedule of a Parent: Step 2 295
186. Checking the Schedule of a Parent: Step 3 295
187. Modifying an Object Schedule - Step 1 296
188. Modifying an Object Schedule - Step 2 297
189. Modifying an Object Schedule - Step 3 297
190. Modifying an Object Schedule - Step 4 298
191. Global Overview Panel: Schedule Modified. 298
192. Using Object Schedule Tags 300
193. Using Object Schedule Tags - Step 3 300
194. Modifying an Object Schedule - Step 4 301
195. Using Object Schedule Tags - Step 5 301
196. Global OverviewPanel: Schedule Modified 302
197. Schedule Conflicts: Example 1 304
198. Schedule Conflicts: Example 2 304
199. Schedule Conflicts: Example 3 305
200. Global Calendar Override Panel 306

201. Example of Global Calendar Override for Object AAOCSM07 307
202. Editing the MVS Object to Enable Cross-System Management 308
203. Object Detail Control - Edit Panel for MVS 309
204. Global Overview Panel: Starting an Object 311
205. Object Start Confirmation Panel 311
206. Global Overview Panel: Object Started 312
207. Global Overview Panel: Stopping an Object 314
208. Object Stop Confirmation Panel 314
209. Global Overview Panel: Object Stopped 315
210. Global Overview Panel: Bouncing an Object 316
211. Confirm Bounce Panel 316
212. Global Overview Panel: Object Bounce 317
213. Global Overview Panel: Resetting an Object 318
214. Confirm Resetting an Object Panel 318
215. Global Overview Panel: Object Reset 320
216. Global Overview Panel: Canceling an Object 321
217. Confirm Cancel Panel 321
218. Global Overview Panel: Object Cancel 322
219. Global Overview Panel: Moving an Object 324
220. Confirm Move Panel 324
221. Global Overview Panel: Moving an Object 325
222. Global Overview Panel: Scheduling an EXEC 326
223. Command Prefix Settings for Group KMZ1 335
224. Global Overview Panel 338
225. Global Overview Panel: Scrolled LEFT 339
226. Object Detail Control Panel Description 349
227. Command Specification Panel Description 353
228. Conditional Command Specifications Panel: Start 359
229. Conditional Command Specifications Panel: Stop 359
230. Event Specifications 1 Panel Example 361
231. Event Specifications 2 Panel Example 365
232. User Notification Panel Example 367
233. Object Groups Panel Example 369
234. Object Requisites Panel Example 371
235. Client/Server Relationships Panel Example 373
236. Object Scheduling Panel Example 376
237. Global Calendar Override Panel Example 379
238. Object Groups Panel Example 383
239. Group Detail Control Example 385
240. ALERT Queue Statistics Panel 396
241. ALERT Overview Application Panel 399
242. ALERT Overview Profile Application Panel 402
243. ALERT Detail Application Panel 404
244. ALERT Detail Profile Panel 408
245. ALERT Help Information (Extended ALERT Text) Panel 410
246. ALERT Detail Information Panel 411
247. ALERT Detail Information Panel: Secondary Panel 411
248. Sample Data Record for ACTN, Subtype=ONLY 442
249. Sample Data Record for ALRT, Subtype=ONLY 444
250. Sample Data Record for EVNT, Subtype=ONLY 445
251. Sample Data Record for EXEC, Subtype=ONLY 446
252. Sample Data Record for RULE, Subtype=ONLY 447
253. Sample Data Record for RULESET, Subtype=AUTO 448
254. Sample Data Record for RULESET, Subtype=SET 450

255. Sample Data Record for RES, Subtype=ONLY 452
256. Sample Data Record for TAPE, Subtype=LOCAL 454
257. Sample Data Record for TAPE, Subtype=PLEX 456
258. EXEC Management Panel with EXEC Description—Documentation View 458
259. EXEC Management Panel with EXEC Statistics—Resource View 459
260. Example of the Status Area of the EXEC Management Panel 460
261. Bottom Portion of EXEC Management Panel—Documentation View 462
262. Bottom Portion of EXEC Management Panel—Resource View 464
263. EXEC Management Display of All Currently Active EXECs. 466
264. General Services Options on the Primary Option Menu 469
265. SERVICE REFRESH CYCLE Panel 469
266. Sample Cycle Setup Member 478
267. Define a Subset of Messages with the PROFILE Command 481
268. Include Messages from the Current Target 481
269. View Messages from the Current Target. 482
270. TIME-INITIATED EXEC REQUESTS Panel 484
271. Example of a Documentation Box 495
272. Global Overview (CSM Administration Panel): Generating a Cross-Reference Report 513
273. CSM Diagnostic Cross-Reference Panel 514
274. Selection Criteria Panel for ALRM-Initiated Rules 536
275. Action Specification Panel for ALRM-Initiated Rules. 538
276. Selection Criteria Panel for ALRT-Initiated Rules 541
277. Action Specification Panel for ALRT-Initiated Rules. 543
278. Selection Criteria Panel for CICS-Initiated Rules 546
279. Action Specification Panel for CICS-Initiated Rules 548
280. Selection Criteria Panel for CMD-Initiated Rules 551
281. Action Specification Panel for CMD-Initiated Rules 553
282. Selection Criteria Panel for DB2-Initiated Rules 556
283. Action Specification Panel for DB2-Initiated Rules. 558
284. Selection Criteria Panel for EXT-Initiated Rules 562
285. Action Specification Panel for EXT-Initiated Rules. 564
286. Selection Criteria Panel for IMS-Initiated Rules 567
287. Action Specification Panel for IMS-Initiated Rules 569
288. Selection Criteria Panel for JES3-Initiated Rules. 572
289. Action Specification Panel for JES3-Initiated Rules 574
290. Selection Criteria Panel for JRNL-Initiated Rules 578
291. Action Specification Panel for JRNL-Initiated Rules. 579
292. Selection Criteria Panel for MSG-Initiated Rules 583
293. Action Specification Panel for MSG-Initiated Rules 586
294. Selection Criteria Panel for Time-Initiated Rules. 590
295. Action Specification Panel for Time-Initiated Rules 593
296. Selection Criteria Panel for VAR-Initiated Rules 596
297. Action Specification Panel for VAR-Initiated Rules. 597

Tables

1. Transfer Commands for Full-Screen Applications 13
2. Application Transfer Commands 14
3. Description of Panels Used to Create a Rule 36
4. Product Requirements for Event Availability 43
5. Event Activity Statistics Panel Primary Commands 49
6. Event Activity Statistics Application Field Descriptions 51
7. Using Primary Commands in the Rule Processor 65
8. Rule Processor Detail Control Panel Field Descriptions: Top Portion 66
9. Rule Processor Detail Control Panel Field Descriptions: Middle Portion 67
10. Rule Processor Detail Control Panel Field Descriptions: Bottom Portion 69
11. Using Primary Commands in the Rule Processor 73
12. Variable Dependencies Panel Field Descriptions 82
13. Alert Action(s) I Panel Field Descriptions 97
14. Alert Action(s) II Panel Field Descriptions 100
15. Variables Available for Creating Rules for ALRT Events 101
16. Pattern Matching for Acct Info Field 159
17. Example of Pattern Matches 159
18. AutoOPERATOR-Supplied EVENT Variables 169
19. Variables Available for Creating Rules for ALRT Events 174
20. Shortcut Commands for Navigating CSM Panels 246
21. Comparison Between Three CSM Object Types 278
22. Possible Desired States an Object Can Have 328
23. Possible Actual States 330
24. Command Line Interface Commands and Parameters 336
25. Field Descriptions: CSM Global Overview Panel 339
26. Primary Commands: CSM Global Overview Panel 340
27. Line Commands: CSM Global Overview Panel 344
28. Field Descriptions: Object Detail Control Panel 350
29. Shortcut Commands: Object Detail Control Panel 352
30. Command Specification Panel Fields 354
31. Object Groups Panel Fields 360
32. Event Specifications 1 Panel Fields 362
33. Event Specifications 2 Panel Fields 365
34. User Notification Panel Fields 367
35. Object Groups Panel Fields 369
36. Primary Commands: Object Groups Panel 369
37. Line Commands: Object Groups Panel 370
38. Object Requisites Panel Fields 371
39. Primary Commands: Object Requisites Panel 371
40. Line Commands: Object Requisites Panel 372
41. Client/Server Relationships Panel Fields 373
42. Primary Commands: Client/Server Relationships Panel 374
43. Line Commands: Client/Server Relationships Panel 375
44. Object Scheduling Panel Fields 376
45. Primary Commands: Object Scheduling Panel 377
46. Line Commands: Object Scheduling Panel 378
47. Global Calendar Override Panel 380
48. Global Calendar Override Primary Commands 381
49. Global Calendar Override Line Commands 382
50. Object Groups Panel Fields 383
51. Primary Commands: Object Requisites Panel 384

52. Line Commands: Object Requisites Panel 384
53. Group Detail Control Panel Fields 386
54. Where to Find More Information about ALERTs. 393
55. Header Fields Description. 432
56. Service Select Codes for Refresh Cycle SERVICE Field. 471
57. TIMEXEC Parameters 487
58. MVS Sample Solutions Index 531
59. Selection Criteria for ALRM-Initiated Rules 536
60. Action Specifications for ALRM-Initiated Rules 538
61. Selection Criteria for ALRT-Initiated Rules 541
62. Action Specifications for ALRT-Initiated Rules. 543
63. Selection Criteria for CICS-Initiated Rules 547
64. Action Specifications for CICS-Initiated Rules 548
65. Selection Criteria for CMD-Initiated Rules 551
66. Action Specifications for CMD-Initiated Rules 553
67. Selection Criteria for DB2-Initiated Rules 557
68. Action Specifications for DB2-Initiated Rules 558
69. Selection Criteria for EXT-Initiated Rules 562
70. Action Specifications for EXT-Initiated Rules 564
71. Selection Criteria for IMS-Initiated Rules 567
72. Action Specifications for IMS-Initiated Rules 569
73. Selection Criteria for JES3-Initiated Rules. 572
74. Action Specifications for JES3-Initiated Rules. 574
75. Selection Criteria for JRNL-Initiated Rules 578
76. Action Specifications for JRNL-Initiated Rules 579
77. Selection Criteria for MSG-Initiated Rules 583
78. Action Specifications for MSG-Initiated Rules 586
79. Selection Criteria for TIME-Initiated Rules 590
80. Action Specifications for TIME-Initiated Rules 593
81. Selection Criteria for VAR-Initiated Rules 596
82. Action Specifications for VAR-Initiated Rules 597

How To Use This Book

The *MAINVIEW AutoOPERATOR Basic Automation Guide* is for operators and system programmers who need to perform basic automation tasks in the data center.

Use this book with the AutoOPERATOR product to learn about the basic automation tasks that you can accomplish with the base component of the AutoOPERATOR product.

Throughout this book, you will see the product referred to simply as AutoOPERATOR.

How This Book Is Organized

The book is organized into six parts:

- Part 1. Introducing MAINVIEW AutoOPERATOR

  Describes how to access AutoOPERATOR and how to use facilities and techniques common to all BBI-2 products.

- Part 2. Reacting to Automation Events Using Rules

  Describes AutoOPERATOR Rules and the AutoOPERATOR Rules applications, and how to use them to implement some basic automation tasks on your system.

- Part 3. Managing Critical Applications Using CSM

  Describes how to use the Continuous State Manager application to automate important processes such as IPLs and shutdowns.

- Part 4. Using Other Basic Automation Applications

  Describes the basic automation applications (such as ALERT Management Facility and Automation Reporter) and general services available with the AutoOPERATOR product that you can use to create basic automation tasks.

- Part 5. Using AutoOPERATOR Sample Solutions

  Describes how to use the AutoOPERATOR Sample Solutions to quickly perform some automation tasks after you have AutoOPERATOR installed.

- Part 6. Appendixes

  Contains:

  – An appendix that contains information about diagnosing CSM errors

  – An appendix for the MVS Sample Solutions

  – An appendix for the DBCTL commands

  – An appendix describing the Selection Criteria and Action Specifications panels for all the events supported by AutoOPERATOR Rules

- Part 7. Glossary and Index

  Contains the glossary and the index for this guide.

MAINVIEW AutoOPERATOR Product Library

MAINVIEW AutoOPERATOR is available with seven 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

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 for CLIST EXECs*
- *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*
- *MAINVIEW AutoOPERATOR Options User Guide*
- *MAINVIEW AutoOPERATOR for MQSeries Installation and User Guide*
- *MAINVIEW AutoOPERATOR Reference Summary*
- *MAINVIEW AutoOPERATOR Solutions Guide*

Related Reading

The following IBM documents are referenced in this guide:

- *CICS Supplied Transactions*, SC33-1686-2
- *CICS Operations and Utilities Guide*, SC33-1685
- *Supervisor Services and Macro Instructions*, GC28-1154
- *TSO Extensions Version 2 REXX Reference*, SC28-1883

and the following BMC Software documents:

- *MAINVIEW Common Customization Guide*
- *MAINVIEW Administration Guide*
- *Using MAINVIEW*
- *MAINVIEW Quick Reference*
- *OS/390 and z/OS Installer Giude*
- *Implementing Security for MAINVIEW Products*
- *MAINVIEW Alternate Access Implementation and User Guide*
- *MAINVIEW Alarm Manager User Guide*

What the Conventions Are

The following syntax notation is used in this manual. Do not enter the special characters.

- Brackets, [ ], enclose optional parameters or keywords.
- Braces, { }, enclose a list of parameters; one must be chosen.
- A vertical line, |, separates alternative options; one can be chosen.
- An *italicized* or <u>underlined</u> parameter is the default.
- AN ITEM IN CAPITAL LETTERS must be entered exactly as shown.
- Items in lowercase letters are values you supply.

Part 1. Introducing MAINVIEW AutoOPERATOR

This part contains chapters that describe how to access AutoOPERATOR and how to use facilities and techniques common to all MAINVIEW applications.

Chapter 1. Introduction to MAINVIEW AutoOPERATOR 3
How AutoOPERATOR Applications Are Organized 4
Operator Workstation Applications 4
Automation Applications 4
How to Access AutoOPERATOR 6
How the Documentation Library for AutoOPERATOR Is Organized 8

Chapter 2. Describing Common Application Functions and Display Fields 9
Requesting an AutoOPERATOR Application 9
Using Another Product from AutoOPERATOR: Product Line Transfers 10
Limitations 11
Using Another AutoOPERATOR Application from AutoOPERATOR:
Application Transfers 12
Limitations 12
Using Product Line Transfers in Conjunction with Application Transfers 12
List of BBI Transfer Commands and Abbreviations for Application Names 13
Issuing Transfer Commands from CSM 20
Using Command Prefixes 20
Exiting from Product and Application Transfers: Using INIT and RETURN 21
Describing Common Application Fields and Functions 22
Describing the Short-Message Field 22
Describing the Primary Command Field 22
Describing the Target Input Field 22
Describing the INTERVAL Field: Refresh Mode Fields 23
Using Scroll Commands 23
Splitting the BBI Terminal Session Screen 24
Using BBI Control Commands 24
Invoking an AutoOPERATOR EXEC from the COMMAND Line 25
Sending BBI-SS PAS Journal Log Messages 25
Using Program Function (PF) Key Definitions in AutoOPERATOR 26
Assigning Commands to Program Function Keys 27

Assigning Labels to PF Keys 27

Chapter 1. Introduction to MAINVIEW AutoOPERATOR

MAINVIEW AutoOPERATOR (also referred to simply as AutoOPERATOR) is a BMC Software online product that provides the tools, techniques, and applications to improve an operator's ability to provide high system availability. AutoOPERATOR helps the operator through error minimization, improved productivity, and problem determination and prevention.

Figure 1 illustrates the overall functions and features of AutoOPERATOR.

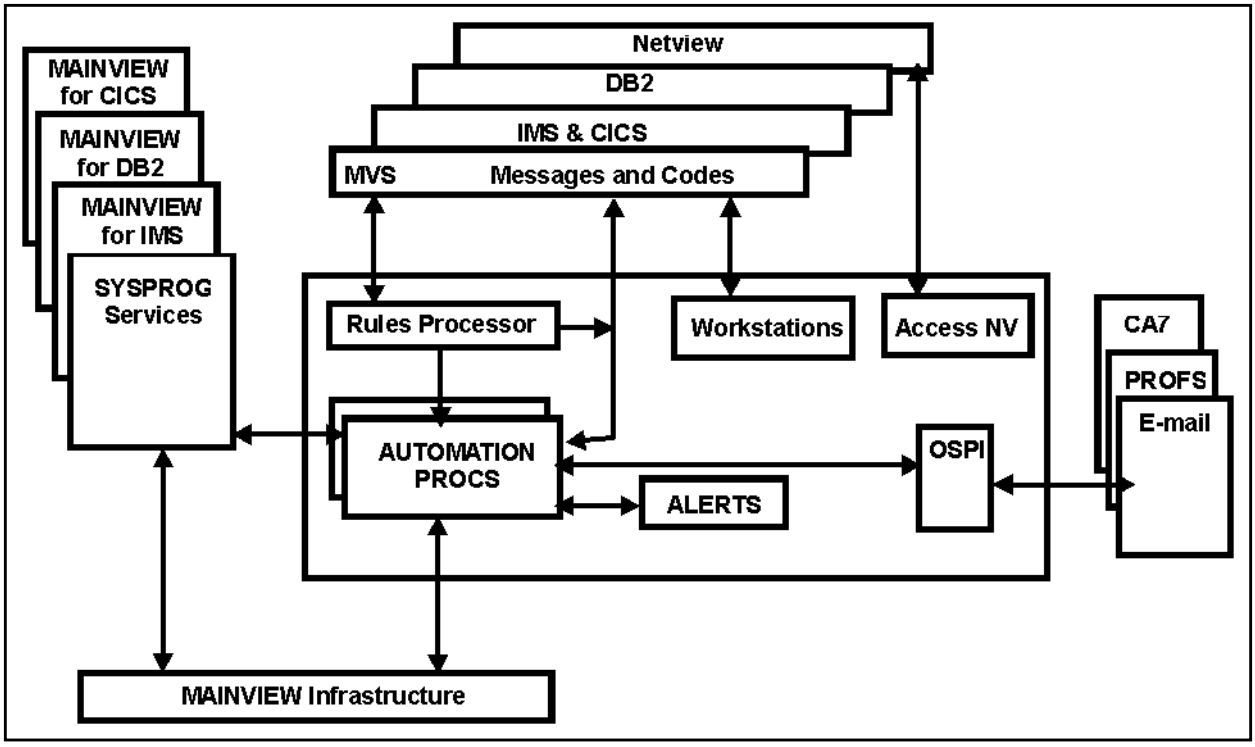

Figure 1. AutoOPERATOR Applications Overview

How AutoOPERATOR Applications Are Organized

AutoOPERATOR applications use an online ISPF-like dialog interface. The applications are divided into two general categories:

- Operator workstation applications
- Automation applications

Operator Workstation Applications

The operator workstation material consists of online menus and applications that allow you to:

- Manage ALERTs
- Modify and display CICS, IMS, MVS, and NetView resources and activity with simple line commands

For documentation about these applications, refer to the “Managing by Exception: Using AutoOPERATOR ALERTs” on page 391 for ALERT Management and the *MAINVIEW AutoOPERATOR Options User Guide* for managing CICS, IMS, MVS, and NetView resources.

Automation Applications

The automation applications are divided into three categories:

- Basic automation
- Advanced Automation
- Dynamic Parameter Manager

Basic Automation

The Basic Automation applications include the Rule Processor applications, and the Continuous State Manager (CSM) and Automation Reporter applications.

Rule Processor: The Rule Processor applications provide a panel-driven method you can use to create Rules that will automate events on your system. Part 2. “Reacting to Automation Events Using Rules” on page 29 contains chapters that explain the Rule Processor applications:

- Introduction to Rules
- What are events
- Identifying events for automation
- Creating Rules
- Managing Rules and automation
- Managing Rules in Rule Sets

Continuous State Manager: This application introduces the philosophy of managing workloads on a system from the object perspective. Continuous State Manager (also referred to as CSM) uses object dependency definitions and a calendar application to determine when restarts are necessary. "Managing Critical Applications with Continuous State Manager" on page 211 explains the Continuous State Manager application and its benefits.

Automation Reporter: The Automation Reporter application allows you to gather data about specific automation activities over user-specified time intervals.

Advanced Automation

The Advanced Automation applications include anything that involves procedural code such as the EXEC Management application, the EXEC Testing Facility, the Time-Initiated EXECs (TIMEXEC) application, and the Open System Procedural Interface (OSPI) application.

EXEC Management application: The EXEC Management application shows you the EXECs you have written to your SYSPROC concatenation, including information about how many EXECs you have running, schedules, CPU time, and abends. "Managing EXECs Using the EXEC Management Application" on page 457 explains the EXEC Management application.

The EXEC Testing Facility is accessed through this application and allows you to select an EXEC to interactively debug.

TIMEXEC: The Time-Initiated EXECs application manages time-initiated EXEC requests by using the CALLX service to invoke your specified EXEC based on time. "Requesting Time-Initiated EXECs: TIMEXEC Application" on page 484 explains the Time-Initiated EXECs application.

The Open System Procedural Interface application is described in the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* and the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*. This interface allows you to capture screen images and keystrokes in an EXEC so that you can programmatically interact with any VTAM application.

Dynamic Parameter Manager

The Dynamic Parameter Manager allows you to dynamically modify the AutoOPERATOR BBPARM members:

AAOALSxx	Contains AutoOPERATOR ALERT parameters
AAOARPxx	Contains AutoOPERATOR Automation Reporter parameters
AAOEXPxx	Contains AutoOPERATOR EXEC parameters
AAOGMExx	Contains General Messages Exchange parameters
AAOTSPxx	Contains TapeSHARE parameters

This application is described in the *MAINVIEW AutoOPERATOR Customization Guide*.

How to Access AutoOPERATOR

The MAINVIEW Selection Menu is displayed, as shown in Figure 2, if you are accessing MAINVIEW from a 3270 display. (This menu is not supported in MAINVIEW Explorer PLEX mode.)

From this menu and its related set of submenus, you can access any installed MAINVIEW product. Products that work together to provide similar solutions are grouped together in submenus. You also have access to a full set of MAINVIEW functions that work with all your MAINVIEW products to help you solve your performance problems. (See *Using MAINVIEW* for a description of these functions.)

To access the AutoOPERATOR submenu, shown in Figure 3 on page 7, select option **A** (Automated Operations) from the MAINVIEW Selection Menu.

```
-------------------------- MAINVIEW Selection Menu --------------------------
 OPTION  ===>                                              DATE   -- 01/09/26
                                                           TIME   -- 10:47:08
      0    Parameters and Options                          USERID -- BMVDID3
      E    Alerts and Alarms                               MODE   -- ISPF 4.8
      P    PLEX Management (PLEXMGR)
      U    Utilities, Tools, and Messages

   Solutions for:
      A    Automated Operations
      C    CICS
      D    DB2
      I    IMS
      L    Linux
      N    Network Management
      S    Storage Management
      T    Application Management and Performance Tuning
      W    WebSphere and MQSeries
      Z    OS/390, z/OS, and USS

      Enter X to Terminate

                       Copyright BMC Software, Inc. 2001
```

Figure 2. Example of MAINVIEW Selection Menu

Figure 3 shows the AutoOPERATOR Primary Option Menu.

```
---------------------------- MAINVIEW AutoOPERATOR ----------------------------
OPTION  ===>                                                   DATE   -- 01/10/03
                                                               TIME   -- 15:12:05
   Operator Workstations                                       USERID -- BAONIS2
      1 ALERTS       ALERT Management                         MODE   -- ISPF 5.0
      2 MVS          OS/390 or z/OS Resources
      3 CICS         CICS Resources
      4 IMS          IMS Resources
      5 NETVIEW      NetView Resources
      6 TAPESHARE    Tape Drive Management
      7 MQSERIES     WebSphere MQSeries

   Administration
      8 AUTOMATION   Basic and Advanced Automation
      9 PARMS        Dynamic Parameter Manager

   General Services
      C CYCLE        Service Refresh Cycle Setup
      L JOURNAL      Display Journal Log
      M MESSAGES     Display Messages and Codes

                      Copyright BMC Software, Inc. 2001
```

Figure 3. Example of AutoOPERATOR Primary Option Menu

From this Primary Option Menu, you can access the operator workstations applications, the automation applications (both basic and advanced), and the Dynamic Parameter Manager application.

Accessing Option 8, Basic and Advanced Automation, displays the following Automation Menu:

```
BMC Software ---------------- AUTOMATION MENU -------------------- AutoOPERATOR
OPTION  ===>                                                   DATE   -- 01/02/01
                                                               TIME   -- 12:24:35




    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 2001, BMC Software, Inc. All rights reserved.
```

Figure 4. AutoOPERATOR Automation Menu

How the Documentation Library for AutoOPERATOR Is Organized

The following table describes how the documentation library for AutoOPERATOR is organized. This table is included because descriptions for the applications from the Primary Option Menu are located in different books within the AutoOPERATOR product library.

For documentation about....	**See...**
Option 1 (`ALERT MANAGEMENT`)	"Managing by Exception: Using AutoOPERATOR ALERTs" on page 391 in this manual.
Options 2, 3, 4, and 5 (`MVS Resources, CICS Resources, IMS Resources, and NetView Resources`)	The *MAINVIEW AutoOPERATOR Options User Guide*.
Option 8: Using the basic automation applications: • Event Activity Statistics • Display/Modify Rules and Rule Sets • Continuous State Manager	"Reacting to Automation Events Using Rules" on page 29 and "Managing Critical Applications with Continuous State Manager" on page 211 in this manual.
Option 8: Using the advanced automation applications: • Display/Modify EXEC Status • Time-Initiated EXEC Requests • Open Systems Procedural Interface (OSPI)	"Managing EXECs Using the EXEC Management Application" on page 457 and "Requesting Time-Initiated EXECs: TIMEXEC Application" on page 484 in this manual. See "Using OSPI" in: • *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* • or *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*
Option 9 (`Dynamic Parameter Manager`)	The *MAINVIEW AutoOPERATOR Customization Guide*.
Using EXECs in AutoOPERATOR	See: • *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or • *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*

Chapter 2. Describing Common Application Functions and Display Fields

This chapter describes functions and display fields that are common to all AutoOPERATOR panels and applications. Use this chapter to learn how to:

- Navigate around the AutoOPERATOR product, its options, and any other products you might have
- Use to transfer to another application or product from an application in a product
- Issue commands from AutoOPERATOR to other products
- Read the upper portion of all AutoOPERATOR panels
- Use program function keys

Requesting an AutoOPERATOR Application

To select an application, enter the application number in the OPTION field of the AutoOPERATOR Primary Option Menu or the AutoOPERATOR Automation Menu and press ENTER. A panel for that application is displayed. If a submenu of options is displayed, more panels are displayed one at a time as you make each selection.

Here are two shortcuts:

- **To skip intervening panels**, concatenate the option numbers.

  For example, to proceed directly from the AutoOPERATOR Primary Option Menu to the Event Activity Statistics application, enter:

  ```
  OPTION ===> 8.2
  ```

  on the Primary Option Menu instead of entering `8` and then selecting Option 2 from the Automation Menu.

- **To proceed directly from one** AutoOPERATOR **application to another** AutoOPERATOR **application**, enter an equal sign (=) and the application's option number in the COMMAND field and press ENTER.

  For example, to proceed directly from the Event Activity Statistics application to the ALERT Management Facility, enter:

  ```
  COMMAND ===>  =1
  ```

  on the Event Activity Statistics panel and press ENTER to display the ALERT Management Facility (Option 1 on the Primary Option Menu).

  To return to the Event Activity Statistics application, use the PF3/END key to exit from the ALERT Management Facility.

Using Another Product from AutoOPERATOR: Product Line Transfers

You can display and access some product's applications from other applications with product line transfers. Use product line transfer commands to access another BMC Software product line you have installed without exiting from your current product application session.

To access another product, enter a valid product line transfer name in the COMMAND field. Valid product line transfer names are:

Command	Product Accessed
AO	MAINVIEW AutoOPERATOR
CAO	MAINVIEW AutoOPERATOR for CICS option
IAO	MAINVIEW AutoOPERATOR for IMS option
MAO	MAINVIEW AutoOPERATOR for OS/390 option
CICS	MAINVIEW for CICS
DB2	MAINVEW for DB2
IMS	IMS Management Facilities

To return to the initial product line, type either of the following commands on the COMMAND line:

Command	Description
INIT	Returns to the initial menu displayed when the TS was invoked.
RETURN	Returns to the Primary Option Menu of the application most currently displayed.

The INIT and RETURN commands are discussed on page 21. These commands can also be assigned to PF keys (see “Using Program Function (PF) Key Definitions in AutoOPERATOR” on page 26).

Example 1: You can transfer from the CICS SYSTEM STATUS display to the MAINVIEW for DB2 Analyzer Service — BFRPL (buffer Pool) as shown in Figure 5.

```
BMC Software ---------------- CICS SYSTEM STATUS -------------- AutoOPERATOR
COMMAND ===> DB2;P;BFRPL                                  TGT ===> CICSA
INTERVAL ==> 1                                            DATE --- 01/03/02
STATUS   --- INPUT                                        TIME --- 17:09:00
```

Figure 5. Example 1 of Product Line Transfer

Example 2: You can also transfer from the ALERT DETAIL panel to the SYSTEM STATUS panel of the MAINVIEW AutoOPERATOR for OS/390 option by typing:

```
BMC Software ---------------- CICS SYSTEM STATUS -------------- AutoOPERATOR
COMMAND ===> MAO;STA                                      TGT ===> SYSB
INTERVAL ==> 1                                            DATE --- 01/03/02
STATUS   --- INPUT                                        TIME --- 17:09:00
```

Figure 6. Example 2 of Product Line Transfer

To return to the ALERT DETAIL panel, press the PF3/END key until you return. Other methods are:

- Enter `INIT` on the COMMAND line of the SYSTEM STATUS panel and press ENTER to return to the AutoOPERATOR Primary Option Menu.
- Enter `RETURN` on the COMMAND line of the SYSTEM STATUS panel and press ENTER to return to the MAINVIEW AutoOPERATOR for OS/390 Primary Option Menu.

  Entering `RETURN` a second time displays the AutoOPERATOR Primary Option Menu.

Limitations

You can transfer to as many product lines as you want until you run out of storage.

If all product lines are not installed into common BMC Software libraries, you must ensure that the other product line libraries are concatenated in the BBI-TS CLIST and BBI-SS PAS JCL. The following library DDNAMEs are affected:

STEPLIB
BBILOAD
PNLLIB
MSGLIB

Using Another AutoOPERATOR Application from AutoOPERATOR: Application Transfers

Application transfer commands allow you to temporarily transfer from one application to another **within the same product line**. Enter the application name (or the appropriate abbreviation) in the COMMAND field and press the ENTER key. Refer to “List of BBI Transfer Commands and Abbreviations for Application Names” on page 13 for application name abbreviations.

A transfer **within** an application also counts as an application transfer. For example, if you display the ALERT OVERVIEW PROFILE panel from the ALERT OVERVIEW panel, this counts as an application transfer.

Use the PF3/END key to exit out of each application that you transferred to. When the original application is displayed, it is not refreshed; press ENTER to refresh the data.

Limitations

Only 10 transfers can be executed; this includes transfers between applications and **within** applications.

Using Product Line Transfers in Conjunction with Application Transfers

Product transfer commands can be stacked with application transfer commands when you want to transfer from an application in one product line to an application in another product line. Use a semi-colon to stack commands; for example:

```
BMC Software -------------- CICS SYSTEM STATUS -------------- AutoOPERATOR
COMMAND ===> DB2;AT                                        TGT ===> SYSB
INTERVAL ==> 1                                             DATE --- 01/03/02
STATUS   --- INPUT                                         TIME --- 17:09:00
```

Figure 7. Example of Stacking Product Line and Application Transfers

These commands, separated by a semi-colon, transfer from the MAINVIEW AutoOPERATOR for CICS SYSTEM STATUS application (in the MAINVIEW AutoOPERATOR for CICS option) to the Active Timer Requests application (of the MAINVIEW for DB2 product line).

List of BBI Transfer Commands and Abbreviations for Application Names

The following tables list the BBI transfer commands and the valid abbreviations for application names.

Table 1. Transfer Commands for Full-Screen Applications

Application Transfer Command	Description
COD|CODES	Messages and Codes List
CYC|CYCLE	Service Refresh Cycle
FOC|FOCAL	FOCAL POINT Overview Display
JOU|JOURNAL	LOG Display
KEY|KEYS	Program Function Keys
LOG	LOG Display
MSG	Messages and Codes List
REF|REFRESH	Service Refresh Cycle
TI	Time Initiated EXEC Requests

Using the information in Table 2, enter the Product Line Transfer command listed in the first column, a semicolon, and an Application Transfer Command from the second column on any COMMAND line. You will get the panel listed in the third column if the product listed in the fourth column is installed.

Table 2. Application Transfer Commands

Product Line Transfer Command	Application Transfer Command	Description	Product
MAINVIEW AutoOPERATOR Base			
AO	ALE|ALERTS	ALERT Overview	MAINVIEW AutoOPERATOR
AO	DPM	Dynamic Parameter Manager	MAINVIEW AutoOPERATOR
AO	EMA	EXEC Management	MAINVIEW AutoOPERATOR
AO	EAS	Event Activity Statstics	MAINVIEW AutoOPERATOR
AO	EXEC	EXEC Management	MAINVIEW AutoOPERATOR
AO	MAS	Event Activity Statistics	MAINVIEW AutoOPERATOR
AO	MSGS|MSGSTAT	Event Activity Statistics	MAINVIEW AutoOPERATOR
AO	OSPI	OSPI Script Development	MAINVIEW AutoOPERATOR
AO	NV	NetView OPERATOR WORKSTATION	MAINVIEW AutoOPERATOR
AO	RUL|RULES	Automation Control	MAINVIEW AutoOPERATOR
AO	SOF	Shared Object Facility	MAINVIEW AutoOPERATOR
AO	TI	Time-Initiated EXECs	MAINVIEW AutoOPERATOR
AO	XAL|XALRTS	Alert Detail	MAINVIEW AutoOPERATOR
CICS Operator Workstation			
CAO	BROA|BROADCAST	CICS Broadcast	MAINVIEW AutoOPERATOR
CAO	STA|STATUS	CICS System Status	MAINVIEW AutoOPERATOR

Table 2. Application Transfer Commands (Continued)

Product Line Transfer Command	Application Transfer Command	Description	Product
CAO	DAT|DATABASE	Database	MAINVIEW AutoOPERATOR for IMS
CAO	PR|PROGRAM	Program	MAINVIEW AutoOPERATOR for IMS
CAO	REG|REGION	IMS Regions	MAINVIEW AutoOPERATOR for IMS
MAINVIEW for CICS			
CICS	CMRTOOLS	MAINVIEW for CICS Tools Menu	MAINVIEW for CICS
CICS	CT	Current Traces	MAINVIEW for CICS
CICS	DM	Display Monitors	MAINVIEW for CICS
CICS	DW	Display Warnings	MAINVIEW for CICS
CICS	EX|EXEC|VIEW svc parm1, parm2	Execute a MAINVIEW for CICS service with defaults or passed parameters	MAINVIEW for CICS
CICS	HIST|HISTORY	MAINVIEW for CICS History Selection	MAINVIEW for CICS
CICS	HT	History Traces	MAINVIEW for CICS
CICS	PUT	MAINVIEW for CICS PUT Level	MAINVIEW for CICS
CICS	SD	Statistics and Defaults	MAINVIEW for CICS
CICS	SM	Start Monitor	MAINVIEW for CICS
CICS	ST	Start Trace	MAINVIEW for CICS
CICS	UGR|UGRAPH	User Defined Graph Selection	MAINVIEW for CICS
MAINVIEW for DB2			
DB2	AN	Analyzer Display Services	MAINVIEW for DB2
DB2	AT	Active Timer Requests	MAINVIEW for DB2
DB2	CT	View Current Traces	MAINVIEW for DB2
DB2	DM	Display Monitors	MAINVIEW for DB2
DB2	DW	Display Warnings	MAINVIEW for DB2
DB2	GC	General Commands	MAINVIEW for DB2
DB2	GT	Graph Thread History	MAINVIEW for DB2
DB2	EX|EXEC svc parm1, parm2	Execute a MAINVIEW for DB2 service with defaults or passed parameters	MAINVIEW for DB2

Table 2. Application Transfer Commands (Continued)

Product Line Transfer Command	Application Transfer Command	Description	Product
DB2	HT	HISTORY Traces	MAINVIEW for DB2
DB2	IO	I/O Analysis Options	MAINVIEW for DB2 (Release 3.1 and above)
DB2	CTIO	Current I/O Traces	MAINVIEW for DB2 (Release 3.1 and above)
DB2	HTIO	History I/O Traces	MAINVIEW for DB2 (Release 3.1 and above)
DB2	MN	Data Collection Monitors	MAINVIEW for DB2
DB2	PM	DB2 System Status	MAINVIEW for DB2
DB2	SD	Display Statistics and Defaults	MAINVIEW for DB2
DB2	SM	Start Monitors	MAINVIEW for DB2
DB2	ST	Start Application Trace	MAINVIEW for DB2
DB2	VT	View Current Traces	MAINVIEW for DB2
IMS Operator Workstation			
IAO	DAT|DATABASE	Database	MAINVIEW AutoOPERATOR for IMS
IAO	LINE	BTAM Lines	MAINVIEW AutoOPERATOR for IMS
IAO	LT|LTERM	LTERMS	MAINVIEW AutoOPERATOR for IMS
IAO	ISC	ISC Links	MAINVIEW AutoOPERATOR for IMS
IAO	REG|REGION	IMS Regions	MAINVIEW AutoOPERATOR for IMS
IAO	NO|NODE	VTAM nodes	MAINVIEW AutoOPERATOR for IMS
IAO	STA|STATUS	Status/Exception	MAINVIEW AutoOPERATOR
IAO	TR|TRANSACTION	Transaction	MAINVIEW AutoOPERATOR

Table 2. Application Transfer Commands (Continued)

Product Line Transfer Command	Application Transfer Command	Description	Product
IAO	PR|PROGRAM	Program	MAINVIEW AutoOPERATOR for IMS
MAINVIEW for IMS			
Note: MAINVIEW for IMS applies to both MAINVIEW for IMS and MAINVIEW for DBCTL.			
IMS	AN	Analyzer Display Services	MAINVIEW for IMS
IMS	AR	Data Entry Database Areas	MAINVIEW AutoOPERATOR for IMS
IMS	AT	Active Timer Requests	MAINVIEW for IMS
IMS	CT	View Current Traces	MAINVIEW for IMS
IMS	DB	Database	MAINVIEW AutoOPERATOR for IMS
IMS	DE	Data Entry Databases	MAINVIEW AutoOPERATOR for IMS
IMS	DM	Display Monitor Requests	MAINVIEW for IMS
IMS	DW	Display Warnings	MAINVIEW for IMS
IMS	EXEC svc parm1, parm2	Execute a service with passed parameters	MAINVIEW for IMS
IMS	EX	Status/Exception	MAINVIEW AutoOPERATOR for IMS
IMS	GC	General Commands	MAINVIEW for IMS
IMS	HT	HISTORY Traces	MAINVIEW for IMS
IMS	MAINVIEW for IMS	MAINVIEW for IMS Performance Management	MAINVIEW for IMS
IMS	LINE	BTAM Lines	MAINVIEW AutoOPERATOR for IMS
IMS	MN	Data Collection Monitors	MAINVIEW for IMS
IMS	MS	Main Storage Databases	MAINVIEW AutoOPERATOR for IMS

Table 2. Application Transfer Commands (Continued)

Product Line Transfer Command	Application Transfer Command	Description	Product
IMS	PD	MAINVIEW for IMS Performance Management	MAINVIEW for IMS
IMS	PM	MAINVIEW for IMS Performance Management	MAINVIEW for IMS
IMS	RC	Fast Path Routing Codes	MAINVIEW AutoOPERATOR for IMS
IMS	SD	Display Statistics and Defaults	MAINVIEW for IMS
IMS	SM	Start Monitors	MAINVIEW for IMS
IMS	ST	Start Trace	MAINVIEW for IMS
IMS	STA|STAT	IMS SYSTEM STATUS	MAINVIEW for IMS
IMS	VT	View Current Traces	MAINVIEW for IMS
MVS Operator Workstation			
MAO	DA	Address Spaces	MAINVIEW AutoOPERATOR for OS/390for OS/390
MAO	DASD	DASD Status/Control	MAINVIEW AutoOPERATOR for OS/390
MAO	DISP|DISPLAY	Address Spaces	MAINVIEW AutoOPERATOR for OS/390
MAO	ENQ|ENQUEUES	Enqueue/Reserve	MAINVIEW AutoOPERATOR for OS/390
MAO	OPE|OPERATOR	Operator Requests	MAINVIEW AutoOPERATOR for OS/390
MAO	OR	Operator Requests	MAINVIEW AutoOPERATOR for OS/390
MAO	REQ|REQUESTS	Operator Requests	MAINVIEW AutoOPERATOR
MAO	RES|RESERVES	Enqueue/Reserve	MAINVIEW AutoOPERATOR for OS/390

Table 2. Application Transfer Commands (Continued)

Product Line Transfer Command	Application Transfer Command	Description	Product
MAO	STA|STATUS	System Status	MAINVIEW AutoOPERATOR for OS/390
MAO	TAP|TAPE	Tape Status/Control	MAINVIEW AutoOPERATOR for OS/390
MAO	MAJ|MAJNODE	VTAM Major Nodes	MAINVIEW AutoOPERATOR for OS/390
MAO	APPL	VTAM Applications	MAINVIEW AutoOPERATOR for OS/390
MAO	CDRM	VTAM CDRMs	MAINVIEW AutoOPERATOR for OS/390
MAO	CDRS|CDRSC	VTAM CDRSCs	MAINVIEW AutoOPERATOR for OS/390
MAO	LINE	VTAM Lines	MAINVIEW AutoOPERATOR for OS/390
MAO	CLS|CLSTR	VTAM Clusters	MAINVIEW AutoOPERATOR for OS/390
MAO	TERM|TERMINAL	VTAM Terminals	MAINVIEW AutoOPERATOR for OS/390

Issuing Transfer Commands from CSM

To use transfer commands from within the CSM application, you must prefix the command with an equal sign (=). For example:

```
BMC Software ---------------    CSM Status Display    ------------ AutoOPERATOR
COMMAND ===>  =XAL                                   TGT ===> SYSB
Primary Commands: Locate    SYNChronize  CALendar
                  EXCeption SORT/UNsort  Xref
⋮
```

Entering `=XAL` as shown above displays the ALERT Detail Panel from the CSM Status Display.

Using Command Prefixes

There are several special characters you can use as command prefixes to issue commands. The following table lists the special characters, the hex equivalents, and a description of what kind of command is generated.

Special Character	Hex Equivalent	Description
%	x'6C'	When used with an EXEC name, submits an EXEC.
¢	x'4A'	When used with an EXEC name, submits an EXEC.
#	x'7B'	When used with an MVS command, issues an MVS command.
.	x'4B'	When used with a BBI control command, issues a BBI command.
!	x'5A'	When used with an IMS command, issues an IMS transaction.
$	x'5B'	Sends a message to an IMS Lterm.
/	x'6I'	Submits an IMS command or DBCTL command. When the command is for a DBCTL-only system, the DBCTL target must be specified in the BBI Journal target field.
?	x'6F'	Invokes a SYSPROG command when issued from a BBI-TS command line.

Exiting from Product and Application Transfers: Using INIT and RETURN

Use the primary commands INIT and RETURN in the COMMAND line to exit from your product or application transfers and return to the original application you started with.

The commands are:

Command	Description
INIT	Returns to the first menu displayed when the BBI-TS was invoked.
RETURN	Returns to the Primary Option Menu of the application most currently displayed. For example, if you have stacked the MAINVIEW AutoOPERATOR for OS/390 SYSTEM STATUS application over the AutoOPERATOR ALERT application, using RETURN displays the Primary Option Menu for the MAINVIEW AutoOPERATOR for OS/390 option. Use RETURN again to return to the Primary Option Menu for AutoOPERATOR. All intermediate panels are bypassed.

If the request is made on the Primary Option Menu, the application from which the transfer was requested is displayed.

These commands can be assigned to PF keys (see “Program Function Keys Panel” on page 26).

Describing Common Application Fields and Functions

The first few lines of all AutoOPERATOR panels are similar and look like this:

```
BMC Software --------------- CICS SYSTEM STATUS -------------- AutoOPERATOR
COMMAND ===>                                                TGT ===> CICSA
INTERVAL ==> 1                                              DATE --- 01/03/15
STATUS   --- INPUT                                          TIME --- 17:09:00
```

The center of the top line always shows the name of the panel or application you are currently using; CICS SYSTEM STATUS in this example.

On all AutoOPERATOR panels, input fields are shown with: `===>`. This arrow means that data can be entered in the field.

Display-only fields that show the date, the time, and other status-related information are shown with three dashes: `---`.

The following sections describe the various areas of the preceding figure.

Describing the Short-Message Field

Short messages, such as edit messages, are displayed on the right side of the top line of the panel. In AutoOPERATOR, when there are no short messages, the area reads AutoOPERATOR or `GENERAL SERVICES`.

To see additional text for a short message, type `MSG` or `CODE` on the COMMAND line to access the Messages and Codes display.

Describing the Primary Command Field

Primary commands such as AutoOPERATOR, MVS, JES2, IMS, CICS, DB2, and any application-specific primary commands, can be entered in the `COMMAND ===>` field of a panel. Each application's description includes information about application-specific commands.

Under certain circumstances, you may want to issue a series of commands. This is called command stacking. To stack primary commands, type the commands on the COMMAND line and separate them by semi-colons; for example:

```
1;1;LOG
```

This sequence stacks three commands.

Describing the Target Input Field

The current target is always shown on the right side of the panel, on the second line. To change targets, enter the name of the new target in the `TGT` field. When you enter a command, the command takes effect in the target shown. In an MSNF (multisystem network function) environment, the target can be remote or on the same system on which your BBI terminal session is running. Ask your AutoOPERATOR administrator for a list of valid targets and their defaults.

Describing the INTERVAL Field: Refresh Mode Fields

The `INTERVAL ==>` field is an input field where you can specify how often you want a display refreshed when the application is in Refresh mode. Valid values for `INTERVAL` are 1 to 99 seconds.

When the application is initially started, the INTERVAL value is determined by the INTERVAL keyword in BBPROF member BBITSP00. If no value is specified in BBITSP00, the screen refreshes every 3 seconds. For more information about BBPARM member BBITSP00, refer to the *MAINVIEW Common Customization Guide*.

The `STATUS ---` field shows the mode of the application:

INPUT Indicates the application is accepting input.

To enter refresh mode on the application after entering input, press the PF6/PF18 key or enter GO in the COMMAND field.

RUN/RUNNING Indicates the application is running, the panel is in Refresh mode, and no commands are accepted until the application is stopped and `STATUS --- INPUT` is displayed.

To stop Refresh mode on an SNA terminal, press the ATTN Key. To stop Refresh mode on a non-SNA terminal, press RESET and then press PA1.

Using Scroll Commands

AutoOPERATOR provides scroll commands that operate similarly to the way IBM ISPF scroll commands operate. Use these scroll commands in AutoOPERATOR applications and General Services.

Specify the amount you want to scroll in the COMMAND line. Then press the UP key (PF7/19) or DOWN key (PF8/20).

COMMAND field:	Press PF7/19 to:	Press PF8/20 to:
M or MAX (Maximum)	Scroll to the top	Scroll to the bottom
1 to 9999 (number)	Scroll x lines up	Scroll x lines down
H (Half)	Scroll half page up	Scroll half page down
P (Page)	Scroll full page up	Scroll full page down
Note: These commands do not work with the PF keys PF10/22 (scroll right) and PF11/23 (scroll left). The amount scrolled is not displayed.		

You can also scroll to the top or the bottom of the data by entering the TOP command or BOTTOM command and pressing ENTER:

```
COMMAND ===> TOP
COMMAND ===> BOT
```

In some applications, for example, the General Service LOG DISPLAY, you can scroll to the left and right. Enter the number of columns you want to scroll in the COMMAND field. To scroll that many columns left, press PF10/22. Scrolling to the left displays the origin or point of entry for each message on the Journal log. To scroll right, press PF11/23.

Splitting the BBI Terminal Session Screen

During a BBI Terminal Session (BBI-TS), you can split your current terminal display into two logical screen displays by pressing the SPLIT (PF2/14) key. The position of the cursor determines the position of the split.

By splitting the display, two panels can be displayed on one screen, allowing you to access two applications or menus. Splitting an application or a tutorial puts the current application on one side of the split and the Primary Option Menu (or ISPF Primary Option Menu if invoked under ISPF) on the other.

The active screen is indicated by the location of the cursor. To switch to the other screen, press the SWAP (PF9/PF21) key. To unsplit the screen and return to only one screen, end one of the applications by pressing the END (PF3/15) key.

Using BBI Control Commands

BBI control commands affect the current BBI-SS PAS and its resources. Refer to the *MAINVIEW Common Customization Guide* for information about BBI and BBI control commands.

Enter BBI commands in the COMMAND field of any AutoOPERATOR application in this format:

```
.command keyword parameter
```

Each BBI control command must be entered with a period (.) prefix. Delimiters in the control command statement can be a blank, comma, or equal sign.

The following examples are the most frequently used BBI control commands. The command abbreviations are shown in bold type.

Example	Description
.C E 2109	Is the same as .**C**ANCEL **E**XEC eid and cancels an EXEC that is either queued or running; eid is the EXEC ID number as shown in DISPLAY EXEC and is always numeric.
.D A	Is the same as .**D**ISPLAY **A**CTIVE ALL and writes the status of the BBI-SS PAS and its resources to the Journal log. Use LOG DISPLAY to see this information online.
.D E HIGH	Is the same as .**D**ISPLAY **E**XEC HIGH and displays the high-priority EXECs.
.D V	Is the same as **D**ISPLAY **V**POOL and displays the EXEC statistics for the SHARED variable pools.

.H E	Is the same as **H**ELP R**E**SET and displays an explanation of the BBI RESET control command.
.HELP	Displays information on all the BBI control commands.
.D RULES	Indicates whether the Rule Processor is active.

Invoking an AutoOPERATOR EXEC from the COMMAND Line

You can invoke an AutoOPERATOR EXEC from the COMMAND line of any AutoOPERATOR panel. This is called a user-initiated EXEC. For more information about AutoOPERATOR EXECs and other ways to invoke EXECs, refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.

To invoke EXECs, enter the EXEC name prefixed with a % (the percentage sign) or a 4 (the cent sign) in the COMMAND field.

For example:

```
COMMAND ===> %PAYROLL START
```

invokes a general payroll EXEC named PAYROLL and passes a positional parameter value of START.

Sending BBI-SS PAS Journal Log Messages

The BBI-SS PAS Journal log includes all messages and commands issued by AutoOPERATOR, as well as the system responses to these messages and commands.

You can send messages to the BBI-SS PAS Journal log by prefixing the message with an asterisk (*); for example:

```
COMMAND ===> *SHIFT CHANGE AT 1600
```

To view the BBI-SS PAS Journal log messages, use the General Service LOG DISPLAY.

Using Program Function (PF) Key Definitions in AutoOPERATOR

A set of program function (PF) key definitions is maintained for each BMC Software product line and the General Service LOG DISPLAY application. This makes the PF key usage unique to each product.

You can modify the PF key definitions by using the General Service KEYS option or by entering KEYS in the command line of the current application. Selecting KEYS displays an input panel that shows the current default values, as shown in Figure 8.

```
BMC Software ---------- PROGRAM FUNCTION KEYS ------- GENERAL SERVICES
COMMAND ===>

The current PF Key assignments for BBI are as follows:
PF1  ===> HELP
PF2  ===> SPLIT
PF3  ===> END                                          To save changes
PF4  ===> PRINT                                        into profile,
PF5  ===> EXPAND                                       enter SAVE on
PF6  ===> GO                                           the command line.
PF7  ===> UP
PF8  ===> DOWN
PF9  ===> SWAP
PF10 ===> LEFT
PF11 ===> RIGHT
PF12 ===> RETRIEVE

When executing under ISPF, these labels may be displayed using PFSHOW
PF1  LABEL ===>        PF2  LABEL ===>       PF3  LABEL ===>
PF4  LABEL ===>        PF5  LABEL ===>       PF6  LABEL ===>
PF7  LABEL ===>        PF8  LABEL ===>       PF9  LABEL ===>
PF10 LABEL ===>        PF11 LABEL ===>       PF12 LABEL ===>

Press ENTER to display keys 13 - 24.  Enter END command to EXIT.
```

Figure 8. Program Function Keys Panel

In the above figure, the initial values as distributed by BMC Software are displayed. Twenty-four keys can be defined. This first panel displays the current values for the first 12 keys. Press ENTER to display the current values for keys 13 through 24, as instructed on the screen.

Assigning Commands to Program Function Keys

Each set of 24 PF key definitions can be changed for a single terminal session, or they can be saved across multiple sessions. Any valid system command or BMC Software product command can be assigned to a PF key. Any input that can be entered in the first input field of a display, such as the COMMAND or OPTION field, can also be assigned as a PF key value. PF keys can also be combined with command line input. For example, if you enter a 5 in the COMMAND field and press PF8 (DOWN), you generate a DOWN 5 command.

To change a PF key assignment, select the KEYS option (K) or enter the KEYS command and:

- Enter the new value over the displayed value
- Assign NOP to disable a PF key
- Blank out the current value to restore the default definition

The changes remain in effect until you end the terminal session. Enter SAVE on the COMMAND line to save the your PF key assignments in your BBPROF data set. After you save this information, it is named AOPFK in your BBIPROF data set.

Assigning Labels to PF Keys

Enter the ISPF command `PFSHOW` on the COMMAND line to display labels assigned to the PF keys. The PF key definitions or labels then display on the last two lines of your current panel, as shown in Figure 9.

```
PF1=HELP     2=SPLIT     3=END     4=PRINT     5=LOG      6=GO
PF7=UP       8=DOWN      9=SWAP   10=LEFT     11=RIGHT   12=RETRIEVE
```

Figure 9. PF Keys Label Display

If there is no label assignment, the first eight characters of the key value are shown.

To assign labels to keys:

1. Use the KEYS command or K Option from the Primary Option Menu to display the input panel shown in Figure 8.
2. Enter the value in the LABEL field of the input panel.
3. Press ENTER to alternate between the input panel for PF keys 1 through 12 and 13 through 24.

Changes are in effect until the end of the terminal session. Save your changes in your BBPROF data set by using the SAVE command.

To turn off the PF key display, enter the command `PFSHOW OFF`.

Part 2. Reacting to Automation Events Using Rules

This part contains chapters that describe AutoOPERATOR Rules and the AutoOPERATOR Rules applications, and how to use them to create a reaction to an automation event.

Chapter 3. Automation with Rules: The Rule Processor Application 31
What the Rule Processor Is 31
What Events Are 32
What a Rule Is 32
Where Rules Are Kept 33
What Actions a Rule Can Take 34
How to Create a Rule 35
How to Begin Identifying Events for Automation 38
How to Manage and Implement Basic Automation with Rules 39
Grouping Rules into Rule Sets 39
What Automation Strategy Is 39
Why Automation Strategy Is Important 40
How to Qualify Displays in the Rule Processor Applications 41
Where to Go from Here: Finding Information Quickly 42

Chapter 4. Describing Events 43
What Events Are 43
Identifying Events for Automation with Rules: Using the Event Activity Statistics Application 46
Accessing the Event Activity Statistics 46
Describing Fields 51
Describing Special Event Types: DB2, EXT, TIME, and VAR 53
DB2 Events 53
EXT Events 53
TIME Events 54
VAR Events 54

Chapter 5. Creating Rules for Events: Using the Rule Processor Detail Control Panel 55
What the Rule Processor Detail Control Panel Is 55
Accessing the Rule Processor Detail Control Panel 57
Describing the Rule Processor Detail Control Panel 64
Using Primary Commands 65
Describing Fields 66
Using the Criteria Match Rate Fields 70
Advanced Rule Processing Techniques 71
Activating and Saving Modified Rules 71

Chapter 6. Creating Rules for Events: Using the Rule Creation Panels 73
Using Primary Commands 73
Step 1. Using Selection Criteria Fields 75
Describing Selection Criteria Fields 76
Step 2. Specifying Variables as Additional Selection Criteria 80
Using AND/OR Logic on the Variable Dependencies Panel 81
Describing Variable Dependencies Fields 82
Step 3. Using Action Specification Fields 85
Describing Action Specification Fields 86
Step 4. Creating ALERTs as Additional Action Specification 93

Creating ALERTs for All Event Types Except ALRT Events 94
Creating ALERTs for the ALRT Event Type 95
Modifying an ALERT in a Rule: Using the MODIFY Function 103

Chapter 7. Creating Rules: Examples 105
Creating a Rule to Suppress a Message 105
Example 1: Creating a Suppression Rule 106
Example 2: Creating a Suppression Rule with Variables 115
Creating TIME-Initiated Rules 121
Example 1: Creating Simple TIME-Initiated Rules 121
Example 2: Creating a TIME-Initiated Rule with ATSTART 126
Creating More Complex Rules 131
Example 1: Setting a Variable with a Rule 131
Example 2: Creating a Rule That Fires When a Variable Value Is Met 136
Using the Rules 141
Creating ALERTs from Rules 142
Example 1: Creating an ALERT Retained across BBI-SS PAS Restarts and MVS IPLs 142
Example 2: Creating an ALERT Whose Priority Increases Over Time 146
Creating a VAR-Initiated Rule 151

Chapter 8. Creating More Powerful Rules 155
Introduction 155
Using Pattern Matching on Selection Criteria Panels 156
Examples of Using Pattern Matching 157
Using Variables in the Rule Processor 161
Defining Variables 161
Resolving Variables 161
Using Variables in Rule Processor Application Fields 164
Variable Modifiers 165
List of AutoOPERATOR-Supplied EVENT Variables 169
List of Variables for ALRT-initiated Rules 174

Chapter 9. Managing Rules and Automation Using the Automation Control Panel 177
Setting Automation Status/Strategy (Upper Portion) 178
Using Primary Commands 179
Viewing Automation Statistics (Middle Portion) 182
Grouping Rules into Rule Sets (Bottom Portion) 183
Using Line Commands 185
Choosing an Automation Strategy 186
Selecting FIRST 186
Selecting ALL 186
Selecting QUALIFIED 187
Selecting INDIVIDUAL 188
Rule Set Filter 188
Filtering Events for Rule Sets 188
Enabling Rule Set Filtering and Rule Set Match Rate 190

Chapter 10. Using the Rule Set Overview Panel to Manage Rules in Rule Sets 197
Using Primary Commands 203
Using Line Commands 203

Copying or Moving Rules 205

Chapter 3. Automation with Rules: The Rule Processor Application

This chapter:

- Introduces basic terms and concepts for Rules you want to use
- Describes how to find information about Rules quickly in this part

Read this chapter to learn basic concepts and terms that will help you write Rules that perform automation on your system. For information about creating Rules for MQSeries events, refer to the *MAINVIEW AutoOPERATOR for MQSeries Installation and User Guide*.

What the Rule Processor Is

The Rule Processor applications provide panels that you can use to create Rules to perform automation tasks such as:

- Suppress messages that appear on operator consoles
- Provide automatic replies to certain events
- Automatically send AutoOPERATOR ALERTs based on events

To best understand Rules and how Rules can help perform automation on your system, you must understand the following basic concepts:

- Events, pages 32 and 43
- Rules, pages 32 and 73
- Selection criteria, pages 32 and 76
- Action specifications, pages 32 and 86
- Automation strategy, page 39.

What Events Are

Events

Events are occurrences in the system, such as messages, commands, and ALERTs, that are eligible to be automated.

In general, events are differentiated depending on where they originated in the system. Some examples of AutoOPERATOR-supported events are MVS write-to-operator messages (WTOs), CICS transient data queue (TDQ) messages, BBI-SS PAS Journal messages, and MVS commands. AutoOPERATOR regards each of these events as a different event type, and AutoOPERATOR supports a total of 12 system-based events.

Each event also has a unique set of attributes. These attributes are used by the Rule Processor to recognize an event and perform automation against it. To support all event types, AutoOPERATOR provides separate panels for each event type where only the applicable attributes are displayed.

AutoOPERATOR also supports an event called TIME. With TIME events, you can cause automation to occur based not on a system event but on the time of day. For example, you can create a Rule to automate a task every 10 minutes from 1:00 pm to 3:00 pm.

Which event types are available for automation on your system depends on your configuration of AutoOPERATOR products. Refer to “Describing Events” on page 43 for more complete information about events.

What a Rule Is

A Rule is basically a two-part conditional statement: when the conditions of the first part of the statement are met, the actions of the second part of the statement are performed. The two parts of the statement are called:

- Selection criteria
- Action specifications

Selection criteria

Selection criteria are user-specified event attributes the Rule uses to recognize an event that will be automated.

When you create a Rule, you must choose specific attributes of the event that the Rule will try to match to incoming events. When the specified selection criteria match that of an incoming event, the Rule fires and the automation action will be taken (see ***Action Specifications*** below).

Each event type has its own version of a selection criteria panel, which contains only the applicable attributes for that event. For more information about selection criteria, refer to “Step 1. Using Selection Criteria Fields” on page 75.

Action specifications

The second part of a Rule is a set of **user-specified automation actions** (also called action specifications).

When a Rule’s selection criteria matches an incoming event and fires, the user-specified automation actions are performed. This is also called **handling** the event.

Some examples of automation actions are suppressing a message from appearing on the console or issuing an ALERT that notifies an operator of an exceptional situation that requires operator intervention.

Each event type has its own version of an action specification panel that contains all the applicable actions a Rule can perform for the event. For more information about action specifications, refer to “Step 3. Using Action Specification Fields” on page 85.

In summary, a Rule screens incoming events for matches to user-specified selection criteria and, when the attributes of an event match the selection criteria, the Rule fires and performs the user-specified automation actions.

Where Rules Are Kept

Rule Sets

The Rule Processor keeps groups of Rules in **Rule Sets**. A Rule Set is simply a way to organize and categorize groups of Rules. The Rule Processor supports multiple Rule Sets. Each Rule Set has its own name and is a member of a partitioned data set (PDS) that resides in storage and is loaded every time the BBI-SS PAS is cold-started (or when the .SET RULE or .RESET RULES command is issued).

An entire Rule Set can be enabled (active) or disabled (inactive). When a Rule Set is disabled, none of the individual Rules within the Rule Set can perform any automation tasks.

Individual Rules also can be enabled or disabled, so, even when a Rule Set is enabled, a Rule within the Rule Set may be disabled and thus not performing any automation tasks.

For more information about Rule Sets, refer to “Grouping Rules into Rule Sets” on page 39 and Chapter 9, “Managing Rules and Automation Using the Automation Control Panel” on page 177.

Figure 10 shows how the Rule Processor (represented by the rulers) automates events.

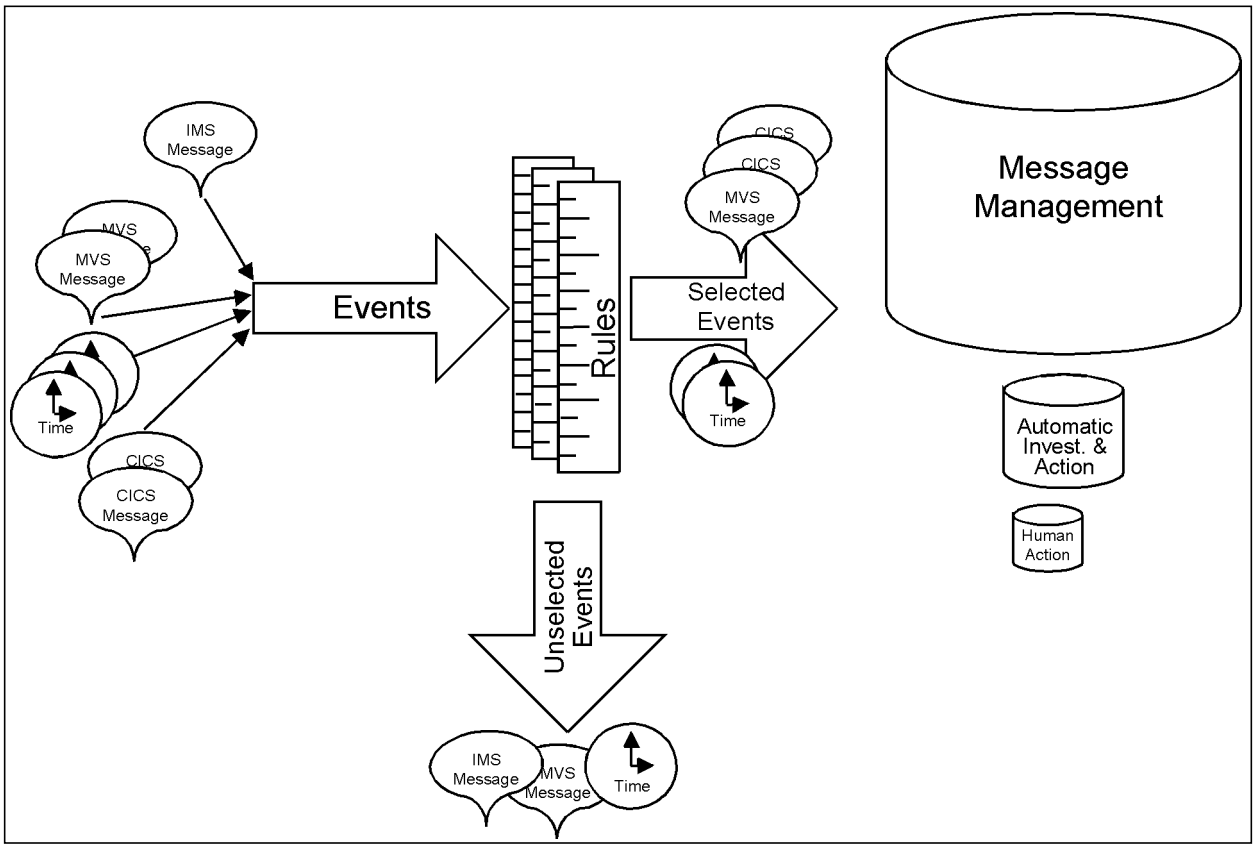

Figure 10. Automating Events

What Actions a Rule Can Take

Some of the actions a Rule can take are:

- Suppress a message from appearing on the console
- Create ALERTs in response to a system event
- Execute MVS, IMS, BBI-SS PAS, or CICS commands at predefined times with TIME-inititated Rules
- Set SHARED variables
- Send messages to TSO users
- Issue WTO messages
- Reword messages or ALERTs
- Route messages to a specific console ID or name
- Schedule EXECs to run
- Notify an outboard pager
- Delete an operator message

How to Create a Rule

You can select events for which to create Rules by examining the flow of events on your system and selecting events or by selecting events from the AutoOPERATOR application — Event Activity Statistics. For more information about selecting events from the Event Activity Statistics application, refer to “Identifying Events for Automation with Rules: Using the Event Activity Statistics Application” on page 46.

Regardless of how you select events for automation, Rules are created using panels in the Rule Processor. A series of six panels are displayed; refer to “Overview of Rules Panels” on page 35 which illustrates the sequence of panels.

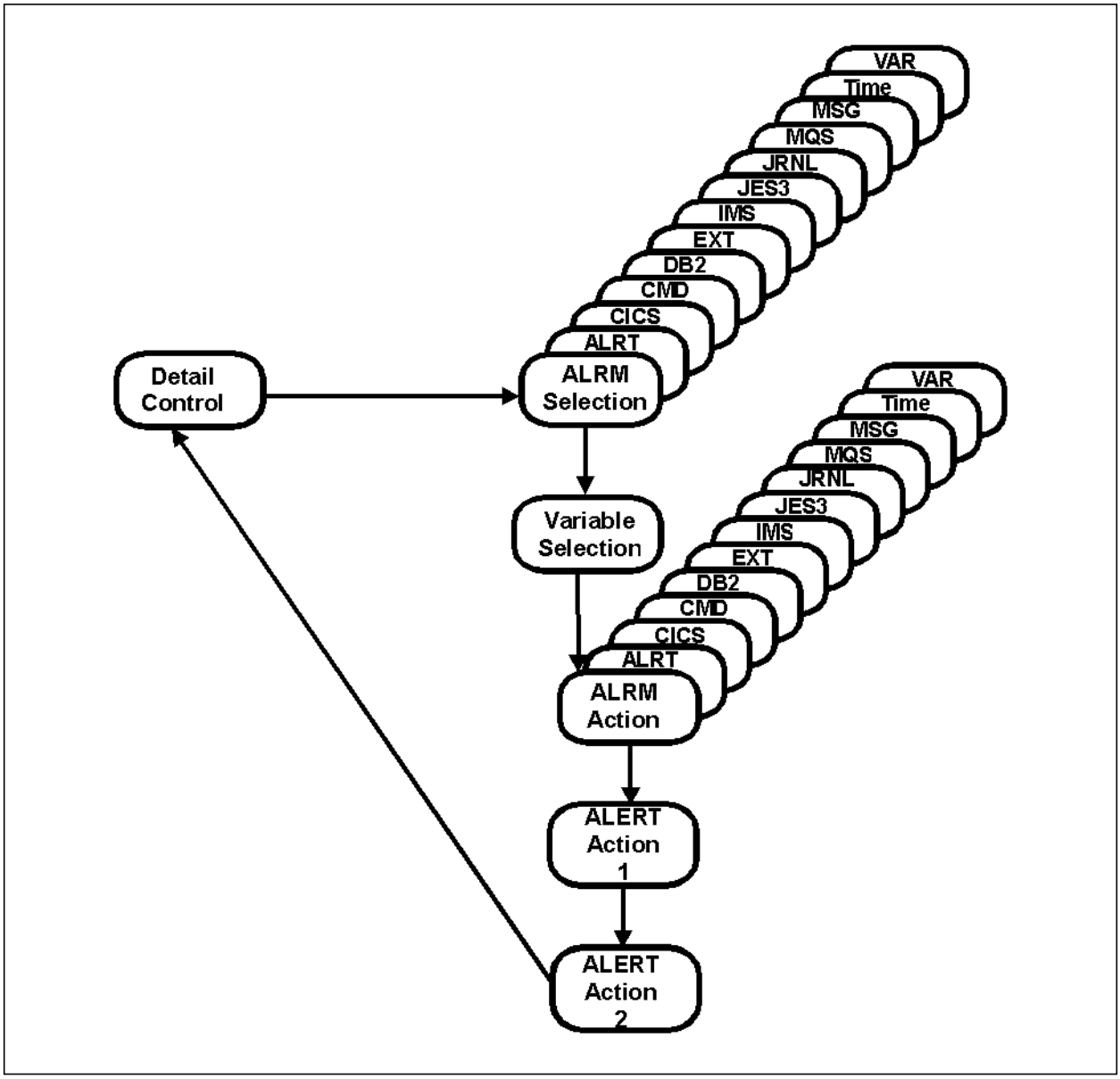

Figure 11. Overview of Rules Panels

All Rule creation begins on the Rule Processor Detail Control panel. One of the things you specify on the Rule Processor Detail Control panel is the event type that you are creating a Rule for.

This is important because, for each event type, **there are different versions** of the Selection Criteria and Action Specification panels where only the applicable attributes and actions for each event type are listed.

On the Selection Criteria panels, you specify information about the event that must match with a Rule before the Rule is fired. On the Action Specification panels, you specify the actions the Rule should take when it is fired.

Optionally, you may use a second (additional) selection criteria panel called the Variable Dependencies panel or a second (additional) set of action specification panels called the ALERT Action(s) panels.

In other words, depending on what event you create a Rule for, different selection criteria items and different actions are displayed on the Selection Criteria panel, Action Specification panel.

Table 3 provides a brief synopsis of each of the 6 panels you can use while creating a Rule.

Table 3. Description of Panels Used to Create a Rule

Panel name	**Allows you to...**	**See page...**
Rule Processor Detail Control	Assign an ID to the Rule, specify the event type, set the maximum rate at which you will allow the Rule to fire, and so on.	55
Selection Criteria	Specify the conditions and attributes the event must match before the Rule will fire. Although there are some criteria that are common to all events, some selection criteria apply only to certain events. Therefore, each event type has a different version of the Selection Criteria panel that contains only the attributes that apply for that specific event type.	75
Variable Dependencies	**Optional** Specify the conditions of additional variables that must exist and be met before the Rule can fire. Conditions on this panel are used as secondary selection criteria for an event. For a Rule to fire, the conditions on both the Selection Criteria panel and the Variable Dependencies panel must be met. This panel is identical for all event types.	82

Table 3. Description of Panels Used to Create a Rule (Continued)

Panel name	Allows you to...	See page...
Action Specification	Specify what actions the Rule takes once it is fired. Although there are some actions that are common to all events, some actions apply only to certain events. Therefore, each event type has a different version of the Action Specification panel that contains only the attributes that apply for a specific event type.	85
Alert Action(s) I and Alert Action(s) II	**Optional** Create an AutoOPERATOR ALERT from a Rule. The Rule Processor has two panels that you can use to create an AutoOPERATOR ALERT from a Rule. When an ALERT is specified as part of a Rule's definition and the Rule fires, it performs the actions on both the Action Specification panel and issues an ALERT using information filled in on the Alert Action(s) panels. Some events have a slightly different set of Alert Action(s) panels. Refer to "Step 4. Creating ALERTs as Additional Action Specification" on page 93 for more information.	93

How to Begin Identifying Events for Automation

The Rule Processor provides the Event Activity Statistics application that displays most of the events eligible for automation detected on your system.

You can use this application as a starting point to identify and select events to write Rules for or when you want to see the frequency of events on your system. Refer to “Identifying Events for Automation with Rules: Using the Event Activity Statistics Application” on page 46 for more information.

The operators and system programmers might also be able to readily identify system events that they regularly have to perform standard actions for which would be good candidates for automation with AutoOPERATOR Rules.

For these events, you immediately can begin creating Rules with the Rule Processor Detail Control panel. Refer to Chapter 5, “Creating Rules for Events: Using the Rule Processor Detail Control Panel” on page 55 for more information.

How to Manage and Implement Basic Automation with Rules

There are two phases to creating automation with Rules. The more basic method is to use the panels to create Rules to attain automation based on certain system events. However, you also need to think about your automation from a wider point of view. Do you want automation to suppress a large number of messages? Do you want to reroute messages? Create ALERTs based on system events which might require operator intervention?

This section introduces how you can manage and control your automation once you have Rules and Rule Sets defined.

Grouping Rules into Rule Sets

The Rule Processor keeps groups of Rules in **Rule Sets**. A Rule Set is simply a way to organize and categorize groups of Rules. The Rule Processor supports multiple Rule Sets. Each Rule Set has its own name and is a member of a partitioned data set (PDS) that resides in storage and is loaded every time the BBI-SS PAS is cold-started (or when the .SET RULESET or .RESET RULES command is issued).

An entire Rule Set can be enabled (active) or disabled (inactive). When a Rule Set is disabled, none of the individual Rules within the Rule Set can perform any automation tasks.

Individual Rules also can be enabled or disabled, so, even when a Rule Set is enabled, a Rule within the Rule Set may be disabled and thus not performing any automation tasks.

When you plan which events you want to automate, you may want to consider grouping Rules that perform similar functions together in the same Rule Set. For example, BMC Software distributes a sample Rule Set that contains Rules whose primary automation action is to suppress messages.

Or you may decide to group Rules that react to the same event type in their own Rule Set. You also need to define a unique name for each Rule Set.

Therefore, you should first consider what categories your Rules might fall under and how you will name the Rule Sets. Then, your site should determine a standard for creating and using Rule Sets.

Once you have developed your Rules and Rule Sets, you can start thinking about **automation strategy**.

What Automation Strategy Is

You could have a situation where an event matches the selection criteria for **more than one Rule** and you might not want all those Rules to fire in response to a single event. Your **Automation Strategy** determines whether all the Rules in all the Rule Sets that match the event are fired to handle that event. The Automation Strategy can also be set so that not all the Rules fire.

Refer to “Choosing an Automation Strategy” on page 186 for a complete discussion about how to set Automation Strategy.

Why Automation Strategy Is Important

Automation Strategy is important because regardless of how many Rules or how few Rules you create for automation, the strategy you set will determine how effective the Rules are. The following sections describe the applications and panels that allow you to manipulate Rules and Automation Strategy.

Using the Automation Control Application: The Automation Control application displays a list of the Rule Sets you have created and displays an overview of how your automation is set up. This application shows:

- Whether automation is active (you can activate/deactivate automation from this panel)
- What automation strategy is set
- Statistics regarding how many events are eligible for automation, how many were already handled by Rules, what the arrival rate of events is, and so on
- A list of all the Rule Sets and their status

Use this application to manage the Rule Sets and basic automation using Rules.

To read more about...	See page...
Using the Automation Control panel to see how well automation is running on your system	177

Using the Rule Set Overview Application: Use the Rule Set Overview application to:

- See all of the Rules within a specific Rule Set and their status
- Set the INDIVIDUAL Automation Strategy for a Rule Set

To read more about...	See page...
Using the Rule Set Overview panel to manage and modify Rules within a Rule Set	197

Once Rules and Rule Sets are created, they are kept in extended common storage. You can save Rules to disk so that they can be recalled after subsystem cold starts.

How to Qualify Displays in the Rule Processor Applications

Throughout the panels in the Rule Processor applications, you will see column headings that are underlined; for example, in the Event Activity Statistics panel:

```
BMC Software ------------ Event Activity Statistics ------------- AutoOPERATOR
COMMAND ===>                                                TGT ===>  SYSA
Activated at: 08-FEB-01  10:06:24  Total Events:        52   DATE --- 01/02/08
Rule Set to be Updated ===> AAORUL00 Sort Criterion: NONE    TIME --- 10:11:47
Line CMDS - (S)elect
LC    Count Handled  Type Text-ID/Description                   Unique-    510
                     ____ ____________________
⋮
```

The underlined areas under `Type` and `Text-ID/Description` are input masking fields where you can enter a text string that will qualify (or mask) the display to show specific information.

These input areas appear in many of the Rule Processor panels. For example, in the Event Activity Statistics panel to see all the events with a type of CMD, enter the characters `CMD` in the field under `Type`:

```
⋮
LC    Count Handled  Type Text-ID/Description                   Unique-    510
                     CMD
                     ---- ----------------
⋮
```

and press ENTER. The display will show only those events with a Type of CMD. Use this technique to mask any display of data in any of the Rule Processor panels where the column headings have input areas.

In the input areas, you also can use the wildcard characters asterisk (*) or plus sign (+). Refer to “Using Pattern Matching on Selection Criteria Panels” on page 156 for more information about how to use wildcard characters.

Where to Go from Here: Finding Information Quickly

The chapters in this part are organized sequentially, based on the following major subjects:

To...	See...
Read about and learn what events are	"Describing Events" on page 43 This chapter defines what events are and which AutoOPERATOR product components you need to have for each event type.
See a display of all the events eligible for automation	"Identifying Events for Automation with Rules: Using the Event Activity Statistics Application" on page 46 This chapter describes how you can use the Event Activity Statistics application when you want to see all the events eligible for automation on your system and select an event right off the panel to automate.
Create Rules with the Rule Processor applications	Although there are basic principles for creating a Rule, actually creating Rules means that you will have to specify information on several panels. For this reason, there are several chapters to cover this material. • Chapter 5, "Creating Rules for Events: Using the Rule Processor Detail Control Panel" on page 55 This chapter describes how to use the initial panel in creating a Rule. • "Example 1: Creating a Suppression Rule" on page 106 This chapter describes the additional panels you can use to create Rules. • Chapter 7, "Creating Rules: Examples" on page 105 This chapter describes the Selection Criteria and Action Specification panels for each event type, sorted alphabetically by event type. • Chapter 8, "Creating More Powerful Rules" on page 155 This chapter describes how to use pattern matching and variables to create more powerful and flexible Rules.
Managing Your Rules and Automation	• Chapter 9, "Managing Rules and Automation Using the Automation Control Panel" on page 177 • Chapter 10, "Using the Rule Set Overview Panel to Manage Rules in Rule Sets" on page 197 These chapters describe how you can implement and control your Rules and Rule Sets.

Chapter 4. Describing Events

This chapter describes:

- What an event is
- Which AutoOPERATOR option is required for certain events
- How to identify events for automation with the Event Activity Statistics application
- Special events

What Events Are

Events are occurrences in the system, such as messages, commands, and ALERTs, that are eligible to be automated.

In general, events are differentiated based on where they originated in the system. Some examples of AutoOPERATOR-supported events are MVS write-to-operator messages (WTOs), CICS transient data queue (TDQ) messages, BBI-SS PAS Journal messages, and MVS commands. AutoOPERATOR regards each of these events as a different event type and AutoOPERATOR supports a total of 13 system-based events.

Which event types are available for automation on your system depends on your configuration of AutoOPERATOR options and other BMC Software products. To help you determine which events might be available on your system, Table 4 describes:

- The event
- The origin or cause of the event
- The product option required for the event type to be eligible for automation
- Whether or not statistics are available for the event in the Event Activity Statistics application

Note that statistics are kept only when the product or option is installed and operational on your system.

For information about special circumstances that can affect event types, refer to “Describing Special Event Types: DB2, EXT, TIME, and VAR” on page 53

Table 4. Product Requirements for Event Availability

Event Type	Caused by...	Product or Option Required	Event Statistics Available?
ALRT	The creation of an AutoOPERATOR ALERTs. AutoOPERATOR ALERTs can be created by an AutoOPERATOR EXEC or by the Rule Processor.	Any AutoOPERATOR	Yes
ALRM	Alarm messages created by the MAINVIEW Alarm Manager.	Any AutoOPERATOR	Yes

Table 4. Product Requirements for Event Availability (Continued)

Event Type	Caused by...	Product or Option Required	Event Statistics Available?
CICS	CICS transient data queue messages. Messages intercepted from the CICS transient data queue (TDQ). This consists of messages that are seen by the AutoOPERATOR transient data exit in CICS. **Restrictions:** CICS can generate multiline messages queued to the transient data queue (application programs typically cannot do this). When any line of a multiline message is suppressed in the TDQ (XTDOUT) exit, all subsequent lines of the multiline message are discarded and not presented to the XTDOUT exit. This restriction is due to the design of the CICS XTDOUT exit. This means that if a Rule fires that handles a CICS message and has 'Display at Destination = NO' specified, all remaining lines of the multiline message are discarded and not seen by the Rules Processor. For example, you cannot suppress a part of any multiline CICS message and still expect to receive the rest of the multiline message in a Rule.	MAINVIEW AutoOPERATOR for CICS	Yes
CMD	Commands processed by the MVS subsystem interface.	MAINVIEW AutoOPERATOR for OS/390	Yes
DB2	DB2 generated messages. Messages sent from a DB2 subsystem. Refer to “Describing Special Event Types: DB2, EXT, TIME, and VAR” on page 53 for more information about this event type.	MAINVIEW for DB2	Yes
EXT	Externally generated messages. External messages directed at the BBI-SS PAS using SUBSYS= parameter on a DD statement in JCL. Refer to “Describing Special Event Types: DB2, EXT, TIME, and VAR” on page 53 for more information about this event type.	Any AutoOPERATOR	Yes

Table 4. Product Requirements for Event Availability (Continued)

Event Type	Caused by...	Product or Option Required	Event Statistics Available?
IMS	IMS MTO generated messages. System messages sent to the IMS Master Terminal, and IMS commands and responses. **Note:** IMS does not route messages sent to the master LTERM by application programs through the AOIEXIT. Therefore, these messages are not input to the Rule Processor and are not eligible for automation.	MAINVIEW AutoOPERATOR for IMS	Yes
JES3	JES3 generated messages.	MAINVIEW AutoOPERATOR for OS/390	Yes
JRNL	BBI Journal generated messages. These messages come from BBI products and the IMFEXEC MSG statement.	Any AutoOPERATOR	Yes
MSG	WTOs and WTORs. Messages that originated as MVS WTOs (write-to-operator) and WTORs (write-to-operator with response).	MAINVIEW AutoOPERATOR for OS/390	Yes
MQS	Events that originated as MQSeries events. For more information, refer to the *MAINVIEW AutoOPERATOR or MQSeries Installation and User Guide*.	MAINVIEW AutoOPERATOR for MQSeries	Yes
TIME	Time initiated event. These Rules are fired based on the time of day.	Any AutoOPERATOR	No
VAR	The creation, deletion, or change in value of a variable in the SHARED variable pool.	Any AutoOPERATOR	No

As this table shows, AutoOPERATOR keeps statistics for most event types that you can view with the Event Activity Statistics application. Use the Event Activity Statistics application to see easily which events occur more frequently and therefore, may be good candidates for automation with Rules. For more information, refer to “Identifying Events for Automation with Rules: Using the Event Activity Statistics Application” on page 46.

Identifying Events for Automation with Rules: Using the Event Activity Statistics Application

This section describes:

- How to access and use the Event Activity Statistics Application
- Event types

The Event Activity Statistics application is a facility that monitors and displays AutoOPERATOR event traffic. Use this application when you want to display events available for automation by Rules. You can select any of the events shown on this panel and write a Rule for it.

Accessing the Event Activity Statistics

Choose the Event Activity Statistics application, Option 1, from the Automation Menu (shown in Figure 12).

```
BMC Software ---------------- AUTOMATION MENU -------------------- AutoOPERATOR
OPTION  ===>                                                DATE   -- 01/01/09
                                                            TIME   -- 17:48:18




    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 2000, BMC Software, Inc. All rights reserved.
```

Figure 12. Automation Menu

Figure 13 shows an example of the Event Activity Statistics application.

```
BMC Software ------------ Event Activity Statistics ------------- AutoOPERATOR
COMMAND ===>                                                 TGT ===> AO41
Activated at: 13-FEB-01  09:00:19  Total Events:    17,065   DATE --- 01/02/13
Rule Set to be Updated ===> AAORUL00 Sort Criterion: NONE    TIME --- 10:48:38
Line CMDS - (S)elect
LC    Count Handled  Type Text-ID/Description                   Unique-    510
                     ____ __________________
_        21      21  MSG  PM0010I  BBI/PM INITIALIZATION COMPLETE           AO4
_        27       0  ALRT PM0010I  BBI/PM INITIALIZATION COMPLETE           AO4
_         1       0  JRNL PM2302I  BBI JOURNAL(2) SELECTED FOR USE
_         1       0  JRNL PM3403I  SS IMAGE LOGGING INACTIVE - NO DD STATEMENT
_         1       0  JRNL PM0215I  ATTEMPTING TO ESTABLISH BBI CONNECTIONS
_         1       0  JRNL XS4201E  BBI/XS OPEN FAILED FOR VTAM ACB
_         1       0  JRNL XS4201I  VTAM APPLID=$AO41   ,ACBERFLG=5A
_         5       0  JRNL PM0503W  BBI/CONNECTION FAILED    FROM $AO41    TO $A
_         1       0  JRNL PM0216I  BBI CONNECTIONS COMPLETED FOR 000 LINK(S)
_         1       0  JRNL EM0110I  SYSTSIN DATASET WILL BE DYNAMICALLY ALLOCATE
_         1       0  JRNL EM0109I  SYSTSPRT DATASET WILL BE DYNAMICALLY ALLOCAT
_         1       0  JRNL DP1113I  SHARED OBJECT FACILITY INITIALIZED
_         1       0  JRNL SM2042I  NETVIEW COMMUNICATIONS INTERFACE IS INITIALI
_         1       0  JRNL AA0229W  UNABLE TO LOAD JES2MAO CSECT
_         1       0  JRNL AU4224E  RULESCAN= FOUND IN BBISSP00 IS OBSOLETE, IGN
_         1       0  JRNL AMTIN1I RESOLVE PLUS INITIALIZATION, RELEASE 3.1.0
_         1       0  JRNL AA1113I  AUTOOPERATOR RELEASE 4.1.B INITIALIZED
_         1       0  JRNL SM0200W  CMRLOAD  NOT FOUND
_         1       0  JRNL SS0975I  BBI/SS -AO41- COLD START COMPLETE
_         1       1  JRNL PM0010I  BBI/PM INITIALIZATION COMPLETE
```

Figure 13. Event Activity Statistics Panel

Statistical information about events shown on this panel is accumulated since the last BBI-SS PAS cold start or since the last time the BBI command:

```
.RESET STATS
```

was issued to reset statistics. For more information about statistics, refer to the field description of the field `UNIQUE` on page 52.

The Count column: Note the `Count` column shows how many times the event has occurred since the last BBI-SS PAS cold start (or since the last time the BBI command `.RESET STATS` command was issued). Use the numbers in this column to help you decide which events (based on their occurrence frequency) might be good candidates for automation with Rules.

The Handled column: If you have any automation already enabled, the `Handled` column shows how many times a Rule fired and handled the event.

Masking the display: You can sort the display with the Type and Text-ID/Description fields to selectively show events by event type or text-ID. To do this:

1. Enter a text-string under either the `Type` column or `Text-ID/Description` column.
2. Press ENTER.
3. The display is sorted to show entries that match the text-string you entered.

To see all the available data on this display, you can scroll both left and right as well as up and down on this panel.

Refer to “Using Primary Commands” on page 49 for more information about using primary commands; refer to “How to Qualify Displays in the Rule Processor Applications” on page 41 for more information about masking.

Viewing events: The events on this panel can be sorted with primary commands to show information such as which events make up most of your event traffic.

For example, if you enter:

```
SHOWPER|SP 50
```

and press ENTER on the Event Activity Statistics panel, the display will show only the events that make up 50% of the event traffic. The following shows an example of the Event Activity Statistics after `SP 50` was entered.

```
BMC Software ------------ Event Activity Statistics ------------- AutoOPERATOR
COMMAND ===> SP 50                                          TGT ===> AO62
Activated at: 13-FEB-01  09:00:19  Total Events:    17,065   DATE --- 01/02/13
Rule Set to be Updated ===> AAORUL00 Sort Criterion: NONE    TIME --- 10:48:38
Line CMDS - (S)elect
LC    Count Handled  Type Text-ID/Description                   Unique-    510
                     ____ _________________
_       522       0  ALRT IST314I  END
_       394       0  ALRT DSI064A OPENACB FAILED, ACBOFLG = X'68', ACBERROR = X
_       317       0  ALRT IEF196I EW1023I  ESTABLISHING REMOTE ELAN CONNECTION,
_       234       0  ALRT DFS826I    TSTDBDV2 .NOFND  X17H
_       231       0  ALRT FT197I   WAITING FOR MAINVIEW FOR CICS SUBTASKS TO END
                          OL
_       217       0  ALRT IST663I  CDINIT     REQUEST FROM VTAME    FAILED  , S
_       191       0  ALRT IST889I  SID = CD9F01E6DCE4BF00
_       186       0  ALRT IST664I  REAL  OLU=USBOOL01.EXTSTN     ALIAS DLU=USBO
_       152     152  MSG  DFS826I   151 DBD ERRORS SENT TO JOB LOG  X17H
_       119       0  ALRT ICH408I USER(BMVJJC1 ) GROUP(MVSSUP  ) NAME(CHAO JUNG
_       101       0  ALRT IEA989I SLIP TRAP ID=X33E MATCHED
_        82       0  ALRT EAO002I 09.31.55 CCIS HEARTBEAT
```

You may find that only 1 or 12 events make up 50% of your event traffic. By creating Rules for these events, you could automate as much as 50% of the events.

Using Line Commands: Selecting an Event

Once you have decided on an event to automate: issue the S line command (for `(S)elect`) in the `LC` column.

The Rule Processor Detail Control panel is displayed. The Rule that you create will be added to the Rule Set named in the `Rule Set to be Updated ===>` field of the Event Activity Statistics application. Proceed to “Creating Rules for Events: Using the Rule Processor Detail Control Panel” on page 55 for more information about creating a Rule.

Using Primary Commands

Table 5 describes the primary commands in the COMMAND field of the Event Activity Statistics panel.

Table 5. Event Activity Statistics Panel Primary Commands

Command	Parameters	Description
FIND|F	text-id	Finds an exact match of a specified text-ID. Use this command when you are searching for a specific text-ID; for example: `F $HASP00` This command can be repeated with the RFIND command.
LOCATE|L	text-id or partial text-ID	Locates the closest match to a specific text-ID. Use this command when you are searching for occurrences of text-IDs and you want to use a text string to perform the search; for example: `L $HASP` This command cannot be repeated with the RFIND command.
RFIND	N/A	Repeats the previous FIND command and searches for the same text-ID.
SHOWALL|SA	N/A	Shows all messages; use this command to redisplay all messages after a SM, SP, or ST command that has been entered previously.
SHOWMESSAGE|SM	text-id or text string	Displays only those messages that match a specific text-ID or text pattern; for example: `SM $HASP` lists only those messages that begin with a message ID of $HASP. To return the display to its initial, issue the SHOWALL|SA command.
SHOWPER|SP	nn	Displays entries (sorted by occurrence) whose combined counts constitutes a given percentage (nn) of the total message traffic. The value nn is equal to the percentage. For example: `SP 80` displays those messages that make up 80% of the total message traffic. To return the display to its initial, issue the SHOWALL|SA command.

Table 5. Event Activity Statistics Panel Primary Commands (Continued)

Command	Parameters	Description
SHOWTOP|ST	nn	Applies the percentage (nn) to the total message traffic to calculate a minimum count. All entries whose count exceeds the calculated minimum count are displayed. For example: `ST 30` displays all messages that occurred more frequently than 30% of the total message traffic. To return the display to its initial, issue the SHOWALL|SA command.
SORT	[COUNT|HANDLED|Type|Text-ID] A|D	Sorts the display, by column heading, in ascending or descending order. Descending order (the default) means the largest value is shown at the top of the display. For example `SORT COUNT` sorts all messages by the number of times they have been encountered in descending order The event encountered most often is shown at the top. You can sort on all of the column headings **except** on the column heading `Description`.

Describing Fields

The input fields under the `Type` and `Text-ID/Description` column headings can be used to qualify the display of events and show only certain event types or events with certain text-IDs. Refer to "How to Qualify Displays in the Rule Processor Applications" on page 41 for more information.

Following is a description of the fields on this panel.

Table 6. Event Activity Statistics Application Field Descriptions

Field	Description
Activated at	Is the date and time of either: • The last cold start of the AutoOPERATOR subsystem • The last time the BBI command `.RESET STATS` was entered
Total Events	Is the total number of events collected since: • The last cold start of the AutoOPERATOR subsystem • The last time the command `.RESET STATS` was entered
Rule Set to be Updated	Is the name of the enabled Rule Set to which a Rule is added if you select an event from the display to be automated with a Rule (using the Select|S line command). The default Rule Set name is AAORUL00. You also can enter a new Rule Set name in this field.
Sort Criterion	Is the column title by which this display is sorted. Valid values are COUNT, HANDLED, Text-ID, and NONE. NONE displays the events in the order in which they first occurred.
LC	Is the field in which you can enter the line command Select|S. When you enter an S and press ENTER to select an event, the Rule Processor Detail Control panel is displayed and you can begin to create a Rule for the event with this panel.
Count	Is the number of times that this event occurred since: • The last cold start of the AutoOPERATOR subsystem • The last time the command `.RESET STATS` was entered
Handled	Is the number of times a Rule was fired to handle the event.
Type	Is the event origin; possible values for the Event Activity Statistics application are: ALRM ALRT CICS CMD DB2 EXT IMS JES3 JRNL MSG MQS

Table 6. Event Activity Statistics Application Field Descriptions (Continued)

Field	Description
Text-ID/ Description	Is a field containing the Text-ID and Description of the event. The maximum length of the text-ID is 16 characters and the Text-ID is highlighted so that it stands out from the remainder of the message text. Only the first 100 characters of the message are displayed. Scrolling left and right affects only the description portion of this field. To scroll this display, use the LEFT or RIGHT commands from the command line or use the assigned PF keys.
UNIQUE	Is the number of unique events collected by the Event Activity Statistics application since the last BBI-SS PAS cold start or the last time the BBI command `.RESET STATS` was entered. An event is considered unique by WORD1 of the event's text. New events are added to the statistics table in the order they occur after an AutoOPERATOR cold start or the last time the `.RESET STATS` command was entered. The default maximum value is 510. When the maximum (510) is reached, new text-IDs are no longer added to the table but the count for events already in the table continues to be updated. Because the Event Activity Statistics collects many event types, you might want to increase the MCTSIZE. To change the default maximum, change the MCTSIZE parameter in BBISSP00 (see the *MAINVIEW Common Customization Guide* for more information).
Note: Events that do not show up on this display are still eligible for automation but you must write Rules for them using the Display/Modify Rules and Rule Sets option (refer to Chapter 9, “Managing Rules and Automation Using the Automation Control Panel” on page 177).	

Describing Special Event Types: DB2, EXT, TIME, and VAR

This section describes event types:

- DB2
- EXT
- TIME
- VAR

which are special cases of event types.

DB2 Events

DB2 messages originate from the MAINVIEW for DB2 product and these messages are sent to the MVS console as Write-to-Operator messages (WTOs). Normally, you need the MAINVIEW AutoOPERATOR for OS/390 option for the Rule Processor to capture MVS traffic and automate events from it.

However, by supporting the DB2 event type, the Rule Processor allows customers who own MAINVIEW for DB2 but do not own the MAINVIEW AutoOPERATOR for OS/390 option to access the MVS DB2 messages. Therefore, DB2 messages are events supported by the Rule Processor with their own event type of DB2 rather than the MSG event type.

Customers who own MAINVIEW AutoOPERATOR for OS/390 have access to DB2 messages as both event types. For simplicity, these customers should automate DB2 using the MSG event type.

If you do not have the MAINVIEW AutoOPERATOR for OS/390 option, you must define the DB2 target in BBPARM member BBJNT00. In addition, you must specify `PRODUCT=DMR` in BBPARM member BBISSP00 and `LOG=YES` in BBPARM member DMRBEX00. For more information about these BBPARM members and parameters, refer to the *MAINVIEW Common Customization Guide*.

EXT Events

The event type EXT (external) is used to capture externally generated messages.

An external message is a message directed at the BBI-SS PAS using SUBSYS= parameter on a DD statement in JCL.

To make EXT events available to the Rule Processor, you must code

```
SUBSYS=xxxx
```

where `xxxx` is the AutoOPERATOR subsystem ID (SSID) on the DD statement to which the messages you want to capture are written. You can have only one SUBSYS=xxxx parameter specification per jobstep.

The AutoOPERATOR subsystem must have been started at least once since the last IPL or the job containing this DD will get a JCL error. The AutoOPERATOR subsystem does not have to actually be active at the time the external job is submitted.

Each line written to the DD will be treated as a separate event in the Rule Processor. The messages can be captured using the EXT event type.

TIME Events

Where most Rules are triggered to fire based on a system message or command, you can create Rules that are triggered to fire on **time**. These events never appear in the Event Activity Statistics application.

For example, you can specify that a Rule fires:

- At 10:00 am every weekday
- Every Wednesday from 1:00 pm to 3:00 pm at 5-minute intervals
- At BBI-SS PAS startup
- At 1:00 pm and fires every 10 minutes until 2:00 pm

All these Rules are TIME-initiated Rules and have an event type of TIME.

A Rule must be enabled at least 121 seconds (two minutes and one second) before the next start interval that it is scheduled to be fired. If it is not enabled, the following interval is used as the start time.

For example, suppose a Rule specifies a start time of 10:00:00 with an interval of one hour. If it is enabled at 10:58:01, it will fire for the first time at 12:00:00. If it is enabled at 10:57:59, it will fire for the first time at 11:00:00.

Refer to “Creating TIME-Initiated Rules” on page 121 for an example of creating a TIME-initiated Rule.

VAR Events

The VAR event is an event where a variable in the SHARED pool is created, deleted, or its value was changed. You can create Rules that are triggered to fire based on this event. These events never appear in the Event Activity Statistics application.

Chapter 5. Creating Rules for Events: Using the Rule Processor Detail Control Panel

There are two ways to select an event for Rule creation: one is to select events from the AutoOPERATOR Event Activity Statistics application (Option 1 from the Automation Menu). The other is to research and prepare a list of events and select Option 2, Display/Modify Rules and Rule Sets and begin creating Rules from the Automation Control panel.

Once you are ready to begin creating Rules, you will begin from the Rule Processor Detail Control panel. The following sections describe the Rule Processor Detail Control panel and the role it plays during Rule creation.

What the Rule Processor Detail Control Panel Is

The Rule Processor Detail Control panel plays a central role in Rule creation; on this panel you enter important information such as:

- The Rule ID of the Rule
- The event type
- The initial mode of the Rule when AutoOPERATOR starts
- Whether the Rule is controlled from over-firing when events flood the system
- Rule documentation information

Also, five additional panels used for Rule creation are accessed through the Rule Processor Detail Control panel and they are (by default) displayed in this order:

1. Selection Criteria
2. Variable Dependencies
3. Action Specification
4. Alert Action(s) I
5. Alert Action(s) II

The Rule Processor Detail Control panel is re-displayed.

“Overview of Panels Required to Create a Rule” on page 56 illustrates the sequence of Rule creation panels.

This chapter describes:

- How to access the Rule Processor Detail Control panel, page 57 through page 63
- How to fill in Rule-related information, page 64 through page 69
- Using the Criteria Match Rate Fields, page 70
- How to save a Rule, page 71

"Example 1: Creating a Suppression Rule" on page 106 describes the subsequent five panels accessed from the Rule Processor Detail Control panel used in Rule creation.

The following figure shows the sequence of panels used to create a Rule.

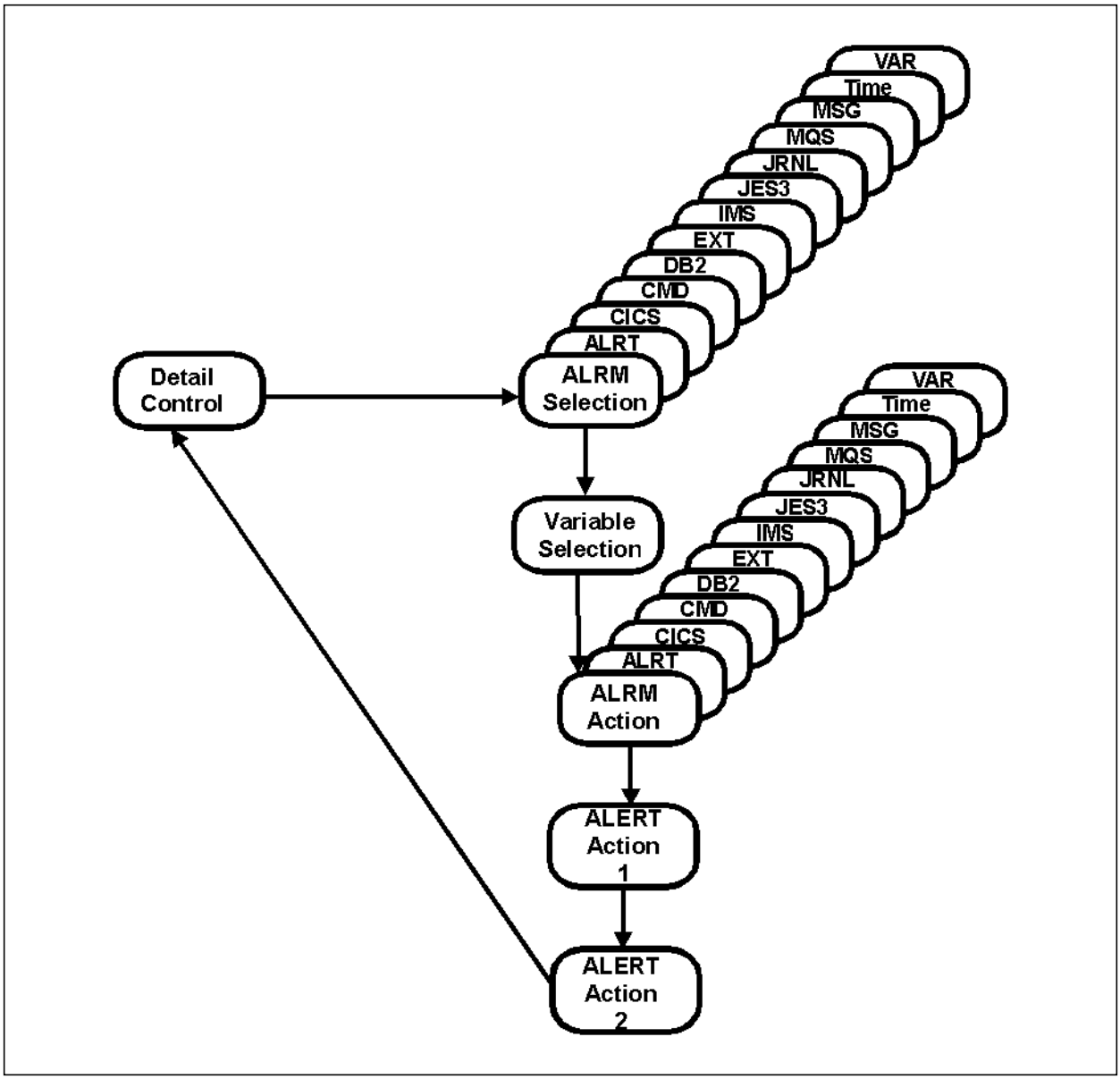

Figure 14. Overview of Panels Required to Create a Rule

Accessing the Rule Processor Detail Control Panel

You can access the Rule Processor Detail Control panel in one of two ways:

- From the Event Activity Statistics application when you select an event on the display to be automated by a Rule
- From the Automation Menu

The following describes how to access the Rule Processor Detail Control panel from either application.

1. To access the Rule Processor Detail Control panel from the Event Activity Statistics panel:

   a. Enter S (for the `(S)elect` line command) in the `LC` field next to the event that you want to write a Rule for. See Figure 15.

```
BMC Software ---------- Event Activity Statistics ------------- AutoOPERATOR
COMMAND ===>                                                TGT ===> AO41
Activated at: 14-FEB-01  14:15:53  Total Events:    19,095   DATE --- 01/02/14
Rule Set to be Updated ===> AAORUL00 Sort Criterion: NONE    TIME --- 17:23:12
Line CMDS - (S)elect
LC    Count Handled  Type Text-ID/Description                   Unique-    510
                     ____ _________________
_        54       0  ALRT IST264I  REQUIRED RESOURCE      CXTSTN   NOT ACTIVE
_         5       4  JRNL IM9100I  COMMAND ACCEPTED
_       943       0  ALRT DSI064A OPENACB FAILED, ACBOFLG = X'68', ACBERROR = X
_        90       0  ALRT HAO002I  CCIS HEARTBEAT
_         4       4  JRNL IM9102I  BBI VERSION 260 STARTED ON 14-FEB-96 AT 14:1
_         2       0  ALRT EAO004C 17.04.24 CCIS HEARTBEAT INTERVAL = 60
_         4       4  JRNL IM9107I  USERS      DEFINED - 020, ACTIVE - 000
_         4       4  JRNL IM9160I  REJECTING VTAM TS LOGONS
S       340       0  ALRT EAO002I 17.04.09 CCIS HEARTBEAT
_         4       0  JRNL IM9327I  AUTOMATIC RESTART MANAGER REGISTRATION IS IN
_         4       4  JRNL AA0103I      IAO - ACTIVE
_         4       4  JRNL AA0104I      MAO - ACTIVE
_       517       0  ALRT IST314I  END
_         4       0  JRNL CF0200I  CMRLOAD  NOT ACTIVE
_        90       0  ALRT BBCSJ016W R0=00000010, R15=00000004, FDBK=100000, FDB
_         2       0  ALRT HAO000I  CCIS COMMAND RECEIVED OK
```

Figure 15. Selecting an Event

The Rule Processor Detail Control panel is displayed (Figure 16).

```
BMC Software ---------- Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                    TGT  --- AO41
                                                                TIME --- 17:25:41
  The following options are displayed in sequence, or may       DATE --- 01/02/14
  be selected by entering the two-character code

   S1 - Selection Criteria            A1 - Action Specification
   SV - Variable Dependencies         AA - Alert Actions(s) I
                                      AD - Alert Actions(s) II
Rule ID       ===> EAO002I
Event Type    ===> ALRT     Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status   ===>          (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===>          Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===>
Last Modified by          on          at

   Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

Figure 16. Rule Processor Detail Control Panel

2. To access the Rule Processor Detail Control panel from the Automation Menu:

   a. Select Option 2, Display/Modify Rules and Rule Sets from the Automation Menu. See Figure 17.

```
BMC Software ---------------- AUTOMATION MENU -------------------- AutoOPERATOR
OPTION  ===>                                                    DATE   -- 01/01/09
                                                                TIME   -- 17:48:18




    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 2000, BMC Software, Inc. All rights reserved.
```

Figure 17. Selecting Option 2, Display/Modify Rules and Rule Sets

The Automation Control panel is displayed (Figure 18).

```
BMC Software -------------- Automation Control ---------------- AutoOPERATOR
COMMAND ===>                                                    TGT ===> AO62
Primary commands: Add, Statshow, Cmdshow                        DATE --- 01/02/14
                                                                TIME --- 17:48:38
Automation Status     ===> ACTIVE         (Active, Inactive)
Automation Strategy   ===> INDIVIDUAL     (Individual, All, First, Qualified)
Honor MPF Suppression ===> NO             (NO/YES)

                        Automation Statistics
  Total Events              23,655   Display suppressed             88
  Events Handled             5,367   Hardcopy suppressed             0
  Current arrival rate    1  / sec   Rule generated Alerts       5,140
  Peak arrival rate      56  / sec   Rule invoked Execs            246
                         Automation Library
LC CMDS --- (S)elect, (E)nable, (D)isable, (T)est, (SA)ve
            (M)ove, (B)efore or (A)fter, (F)ilter Criteria
LC   Rule-Set Status   Rules     Fired  Filtered    Date      Time   Strategy
___  AAORUL00 ENABLED     14     5,140    23,655  14-FEB-01 14:15:54 FIRST
___  AAORULBC ENABLED     54       232    23,655  14-FEB-01 14:15:55 FIRST
___  AAORULCM ENABLED     52       124    23,655  14-FEB-01 14:15:55 FIRST
___  RULCICS  ENABLED     25         0    23,655  14-FEB-01 14:15:57 FIRST
___  AAORULBA DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULBB DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULBF DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULBS DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULCS DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULC1 DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULD1 DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULD2 DISABLED   N/A   N/A       N/A          N/A    N/A
```

Figure 18. Automation Control Panel

b. Select a Rule Set by entering S (for the `(S)elect` line command) next to the Rule Set name. See Figure 19 on page 60.

```
BMC Software ----------------- Automation Control ---------------- AutoOPERATOR
COMMAND ===>                                                  TGT ===> AO62
Primary commands: Add, Statshow, Cmdshow                      DATE --- 01/02/14
                                                              TIME --- 17:48:38
Automation Status     ===> ACTIVE         (Active, Inactive)
Automation Strategy   ===> INDIVIDUAL     (Individual, All, First, Qualified)
Honor MPF Suppression ===> NO             (NO/YES)

                        Automation Statistics
  Total Events              23,655   Display suppressed             88
  Events Handled             5,367   Hardcopy suppressed             0
  Current arrival rate    1  / sec   Rule generated Alerts       5,140
  Peak arrival rate      56  / sec   Rule invoked Execs            246
                         Automation Library
LC CMDS --- (S)elect, (E)nable, (D)isable, (T)est, (SA)ve
            (M)ove, (B)efore or (A)fter, (F)ilter Criteria
LC   Rule-Set Status   Rules     Fired  Filtered    Date      Time   Strategy
S    AAORUL00 ENABLED     14     5,140    23,655  14-FEB-01 14:15:54 FIRST
___  AAORULBC ENABLED     54       232    23,655  14-FEB-01 14:15:55 FIRST
___  AAORULCM ENABLED     52       124    23,655  14-FEB-01 14:15:55 FIRST
___  RULCICS  ENABLED     25         0    23,655  14-FEB-01 14:15:57 FIRST
___  AAORULBA DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULBB DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULBF DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULBS DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULCS DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULC1 DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULD1 DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULD2 DISABLED   N/A   N/A       N/A          N/A    N/A
```

Figure 19. Automation Control Panel: Creating a New Rule

c. Or, add a new Rule Set with the ADD primary command on the COMMAND line. See Figure 20 on page 61.

```
BMC Software -------------- Automation Control ---------------- AutoOPERATOR
COMMAND ===> ADD RULALRT                                      TGT ===> AO62
Primary commands: Add, Statshow, Cmdshow                      DATE --- 01/02/14
                                                              TIME --- 17:48:38
Automation Status     ===> ACTIVE         (Active, Inactive)
Automation Strategy   ===> INDIVIDUAL     (Individual, All, First, Qualified)
Honor MPF Suppression ===> NO             (NO/YES)

                        Automation Statistics
  Total Events              23,655   Display suppressed             88
  Events Handled             5,367   Hardcopy suppressed             0
  Current arrival rate    1  / sec   Rule generated Alerts       5,140
  Peak arrival rate      56  / sec   Rule invoked Execs            246
                         Automation Library
LC CMDS --- (S)elect, (E)nable, (D)isable, (T)est, (SA)ve
            (M)ove, (B)efore or (A)fter, (F)ilter Criteria
LC   Rule-Set Status   Rules     Fired  Filtered    Date      Time   Strategy
___  AAORUL00 ENABLED     14     5,140    23,655  14-FEB-01 14:15:54 FIRST
___  AAORULBC ENABLED     54       232    23,655  14-FEB-01 14:15:55 FIRST
___  AAORULCM ENABLED     52       124    23,655  14-FEB-01 14:15:55 FIRST
___  RULCICS  ENABLED     25         0    23,655  14-FEB-01 14:15:57 FIRST
___  AAORULBA DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULBB DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULBF DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULBS DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULCS DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULC1 DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULD1 DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULD2 DISABLED   N/A   N/A       N/A          N/A    N/A
```

Figure 20. Adding a New Rule Set

d. A new Rule Set named RULALRT is added; select the RULALRT Rule Set with the S (for `(S)elect`) line command to add Rules.

The Rule Processor Detail Control panel is displayed; see Figure 21.

```
BMC Software ---------- Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO41
                                                              TIME --- 17:25:41
  The following options are displayed in sequence, or may     DATE --- 01/02/14
  be selected by entering the two-character code

   S1 - Selection Criteria            A1 - Action Specification
   SV - Variable Dependencies         AA - Alert Actions(s) I
                                      AD - Alert Actions(s) II
Rule ID       ===> EAO002I
Event Type    ===> ALRT     Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status   ===>          (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===>          Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===>
Last Modified by          on          at

   Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

Figure 21. Rule Processor Detail Control Panel

3. Use the Rule Processor Detail Control panel to specify the following information:

   - **The unique Rule ID for the Rule**

     You should establish a guideline for assigning Rule IDs to Rules. For example, you may want to develop a naming scheme where the Rule ID is associated with a specific group of people or functions.

     If you do not assign a Rule ID to a Rule, AutoOPERATOR automatically assigns a Rule ID in the form RULnnnnn, where nnnnn is a number from 1 to 99999. The format RULnnnnn is reserved for AutoOPERATOR use only and cannot be used as a user-specified Rule ID.

     When you select an event for automation from the Event Activity Statistics application, the `Rule ID` and the `Event Type` fields are automatically filled in with data from the `Text ID` and `Type` fields in the Event Activity Statistics application. The first eight characters of the selected event's Text-ID are used as the Rule ID.

     For more information about the Event Activity Statistics, refer to “Identifying Events for Automation with Rules: Using the Event Activity Statistics Application” on page 46.

   - **The event type that you are writing a Rule for**

     You must specify an event type because the event type determines which primary Selection Criteria and Actions(s) Specification panels are displayed as you create or edit a Rule.

     For example, if you are creating a Rule for a CICS event, you will receive a different Selection Criteria and Action Specification panel than if you are creating a Rule for an IMS event.

4. To see a list of possible types, enter a question mark (?) in the `Event Type` field.

```
BMC Software  ------------- Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO41
                                                              TIME --- 17:25:41
  The following options are displayed in sequence, or may    DATE --- 01/02/14
  be selected by entering the two-character code

   S1 - Selection Criteria            A1 - Action Specification
   SV - Variable Dependencies         AA - Alert Actions(s) I
                                      AD - Alert Actions(s) II
Rule ID       ===> EAO002I
Event Type    ===> ?        Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status   ===>          (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===>          Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===>
Last Modified by          on          at

   Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

Figure 22. Rule Processor Detail Control Panel: Displaying Available Event Types

The following Rule Processor Event Types panel is displayed.

```
BMC Software  ------------- Rule Processor Event Types ----------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO41

MVS Events: (MAO Option Required)    IMS Events: (IAO Option Required)
 C CMD  -  Operator commands.         I IMS  - IMS MTO messages.
 L ALRM -  Alarm Manager events.
 M MSG  -  WTO or WTOR message.      CICS Events: (CAO Option Required)
 3 JES3 - JES3 events                 Q CICS - Messages from TD Queue.

DB2 Event: (MAO Option Required)     MQSeries Event:  (QAO Option Required)
 D DB2  - DB2 msgs (WTOs).            S MQS  - MQSeries instrumentation events
                                               and messages

AO Events: (Common to all options)
 A ALRT - Alerts being added to Queue.
 J JRNL - BBI Journal messages.
 T TIME - Time-initiated Rule.
 V VAR  - Event-Type(Var).
 X EXT  - External msgs directed to BBI SS.
```

Figure 23. Rule Processor Event Types Panel

This panel shows the 3- or 4-character event type name for all the event types supported by the Rule Processor. You can also use a 1-character event type abbreviation in the `Event Type` field.

For more information about event types, refer to “Describing Events” on page 43.

5. Once the Rule ID and the event type are filled in, press ENTER to proceed to the next panel and continue creating the Rule.

   “Example 1: Creating a Suppression Rule” on page 106 contains information about using the five other Rule creation panels.

“Describing the Rule Processor Detail Control Panel” on page 64 describes the other fields on the Rule Processor Detail Control panel that can be filled in at the beginning of Rule creation or at the end.

Describing the Rule Processor Detail Control Panel

You can fill in the following fields before you proceed to the other panels or after you have completed the other panels:

- **Initial Mode Rule**

  When a Rule Set is enabled, this setting determines whether individual Rules within that Rule Set are ENABLED, DISABLED, or in TEST mode:

  – If a Rule is ENABLED when the Rule Set is enabled, the Rule is ready to perform automation.

  – If a Rule is DISABLED, the Rule will not perform any automation.

  – If a Rule is in TEST mode, the Rule will fire but no actions are taken.

    Using TEST mode is helpful when you want to see that you have set the correct selection criteria to cause the Rule to fire. Note that if the Rule has an EXEC associated with it, the EXEC is scheduled and variable IMFRLSTA is set to TEST.

- Criteria to change the status of a Rule to SUSPEND, DISABLE, or NOACTION is based on the volume of qualifying events within the specified time interval.

  Possible uses of this criteria include:

  – Disable a Rule when its firing rate exceeds the set threshold, possibly allowing a subsequent Rule to then fire

  – Prevent subsequent duplicate or unproductive actions from being scheduled

  – Prevent AutoOPERATOR from being flooded with excessive actions when an unexpectedly large volume of events occur

  When the `If matched` count is exceeded within the interval of time specified by `in seconds`, the Rule status is changed to the `then status` value.

  Specifying a `then status` of Disable will disable the Rule. You can manually enable this Rule. Disabled Rules are ignored by the Rules processor.

  Specifying a `then status` of Suspend temporarily disables the Rule. AutoOPERATOR enables the Rule once the arrival rate of qualifying events falls below your specified threshold. Disabled Rules are ignored by the Rules processor.

  Specifying a `then status` of NOACTION disables all actions for this Rule. In this case, no actions are issued, which means EXECs will not be scheduled, ALERT actions will not occur, and so on. However this Rule is considered to have fired. Therefore, if the strategy is set to FIRST, subsequent Rules in this Rule Set will not be inspected.

  Refer to “Using the Criteria Match Rate Fields” on page 70 for more information about using the match rate.

- **Documentation information**

  This information provides data for the Rule Set Overview panel that you will use to view and manage the Rules within a specific Rule Set. For this reason, it is important for you to complete these fields, especially if you have many users adding and using Rules within a single Rule Set.

Using Primary Commands

You can enter the primary commands shown in Table 7 in the `COMMAND` field of the Rule Processor Detail Control panel (or any of the panels used in Rule creation) as a shortcut to a specific panel. The table shows the command, the panel displayed, and a brief description.

Table 7. Using Primary Commands in the Rule Processor

Command	**Displays...**	**Description**
blank (Default)	Displays the Selection Criteria panel when you press ENTER	When you do not specify primary command, pressing ENTER displays the Selection Criteria panel for the event type you specify in the `Event Type` field. By not using the primary commands in this table and only pressing ENTER you progress through the six Rule creation panels.
S1	Selection Criteria - *event type* panel	Use this panel to specify attributes of an event that a Rule must match before the Rule is fired. You must specify at least one criterion.
SV	Variable Dependencies - *event type* panel	Use this panel to set the values of variables when you want to use variables for any of your selection criteria. Use of this panel is optional.
A1	Action Specification - *event type* panel	When the selection criteria of a Rule matches an event and the Rule is fired, the actions specified on this panel are taken by the Rule.
AA	Alert Action(s) I - *event type* panel	Only some of the Rules you create will have ALERTs associated with them. For those Rules, you can use this panel to add or delete the ALERT to be issued as a result of the Rule being fired.
AD	Alert Action(s) II - *event type* panel	This panel is a continuation of the Alert Action(s) I panel and includes additional fields that you can use while creating an AutoOPERATOR ALERT from a Rule.

Describing Fields

The following sections describe the fields on the Rule Processor Detail Control panel.

Top portion of the panel: The following describes the fields in the top portion of the panel.

```
Rule ID       ===> IMSJRNL
Event Type    ===>          Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)
```

Figure 24. Top Portion: Rule Processor Detail Control Panel

Table 8. Rule Processor Detail Control Panel Field Descriptions: Top Portion

Field	Description
Rule ID	Is the 1- to 8-character unique name of the Rule. When you assign a Rule ID, you must remember to make it unique from previously created Rules because a duplicate Rule ID will override the original Rule. Valid characters that can be used in the Rule ID are letters A through Z, numbers 0 through 9, the pound sign (#), the at sign (@), and the dollar sign ($). If this field is left blank, the Rule Processor assigns a Rule ID in the format RULxxxxx, where xxxxx is a number from 0 to 99999. Therefore, when you assign a Rule ID, you should avoid the naming convention RULxxxxx. In addition, if you issue a BBI`.RESET RULESET` or `.SET RULESET=` command, the Rule Processor assigns new Rule IDs to the Rules that do not have user-defined Rule IDs. To prevent reassignment of a Rule ID, assign an ID when you create a new Rule.
Event Type	Is the event type that describes the source of the message. “Describing Events” on page 43 describes all the possible event types that the Rule Processor processes. You also can enter a question mark in the `Event Type` field and a list of the possible event types is displayed. You must enter an event type before you can proceed to the subsequent Rule creation panels.

Table 8. Rule Processor Detail Control Panel Field Descriptions: Top Portion (Continued)

Field	Description
Initial Mode	Is the initial mode of the Rule when its Rule Set is enabled. All Rules reside in Rule Sets which can be enabled (available to process events) or disabled (not available). When the Rule Set is enabled, you can specify the initial mode of individual Rules in this field; possible modes are: • ENABLED means that the Rule is available to process events. • DISABLED means that the Rule is not available to process events. • TEST means that the Rule will be fired in test mode. When a Rule fires in TEST mode, no actions of the Rule are taken, but if an EXEC is associated with the Rule, the EXEC is scheduled and the variable IMFRLSTA is set to TEST. Once the Rule Set is enabled, you can change the mode of an individual Rule on the Rule Set Overview panel.

Criteria match rate threshold: The following table describes the fields in the middle portion of the panel called `Criteria match rate threshold`. For more information about using these fields, refer to "Using the Criteria Match Rate Fields" on page 70.

```
Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status   ===>          (SUSPEND, DISABLE, NOACTION)
```

Figure 25. Middle Portion: Rule Processor Detail Control Panel

Table 9. Rule Processor Detail Control Panel Field Descriptions: Middle Portion

Field	Description
If matched	Specifies the number of times a Rule may be matched to an event within a certain interval of time. This value is also called match count target. Possible values range is 0 to 100 times. This value is used in conjunction with the `in seconds` and `then status` fields to define what event rate is used to prevent the Rule from matching too often. If value is 0, the Rule Set Criteria Match Rate is ignored. The Rule Set match rate is reset only when the Rule is updated or if the BBI-SS PAS is cold started. Refer to "Using the Criteria Match Rate Fields" on page 70 for more information.

Table 9. Rule Processor Detail Control Panel Field Descriptions: Middle Portion (Continued)

Field	Description
in seconds	Is the interval of time, set `in seconds`, over which the match count will be collected for an event. Possible intervals range is 1 to 99999 seconds. When a Rule is matched by the number of times set in the `If matched` field and within the time interval set in the `in seconds` field, the action specified in the `then status` field is taken. Refer to "Using the Criteria Match Rate Fields" on page 70 for more information.
then status	Specifies whether a Rule should be suspended or disabled when the number of times it is matched reaches the match count target within a specific time interval. Possible values are: **SUSPEND** Suspends the Rule. Event matches to the Rule are ignored until the actual match count drops below the match count target and the Rule ceases to fire. When the match rate falls below the threshold, the Rule will begin to fire again. **DISABLE** Disables the Rule. Event matches to Rule are ignored when the actual match count exceeds the specified match count target, and the Rule is disabled. You must manually re-enable the Rule. Refer to "Using the Criteria Match Rate Fields" on page 70 for more information. **NOACTION** Takes no action. When the match count for a Rule exceeds its match rate, the Rule is set for no action. The Rule will match and the fired count will increase, but no actions specified for that Rule will take place. The action is resumed when the match rate falls below the threshold.

Application Information Fields: The following table describes the Application information portion of the panel.

```
Application information:
Group         ===> IMS      Function      ===> JOURNAL  Code   ===> JN
Author        ===> DDH1     Description   ===> JOURNAL ALL IMS MESSAGES
Last Modified by          on          at
```

Figure 26. Bottom Portion: Rule Processor Detail Control Panel

These fields are used to document information about the Rule and this information appears on the Rule Set Overview panel. By filling in these fields, the Rule Set Overview panel will contain more complete information about each Rule.

Table 10. Rule Processor Detail Control Panel Field Descriptions: Bottom Portion

Field	**Description**
Group	Is the user-defined 1- to 8-character group name assigned to this Rule
Function	Is the user-defined 1- to 8-character function name assigned to this Rule
Code	Is the user-defined 2-character code assigned to this Rule
Author	Is the user-defined 1- to 8-character name of author of this Rule
Description	Is the user-defined 1- to 25-character description of this Rule
Last Modified	Is a display-only field that shows the date and time when the Rule was last modified and the user ID of whomever changed the Rule

Using the Criteria Match Rate Fields

In addition to specifying thresholds for each Rule, you can specify a default threshold for each Rule within a Rule Set by using the Criteria Match Rate settings in the filter of the Rule Set (see “Managing Rules and Automation Using the Automation Control Panel” on page 177).

Specifying a Rule Set match rate affects only the Rules that do not have a threshold specified. The Rule Set Match Rate can be used to ensure that every Rule has a match rate specified, thus preventing looping Rules or flooding the PAS with automation.

Some Rules might have a normally high match rate, such as a Rule to suppress a message. Before changing the Rule Set default Match Rate Criteria, Rules should be reviewed. Rules designed to have a high Match Rate can be coded with `If matched ===> 0` to ignore the default Rule Set value.

For every Rule you create, you should determine an appropriate firing rate. For example, sometimes a single event generates a flood of events in a very short amount of time. In this case, you might want the Rule for that event to fire only once, based on the first time the event occurs. By using the fields `If matched`, `in seconds`, and `then status`, you can resolve this problem.

The `If matched` and `in seconds` fields define the threshold matching rate for a Rule. When the match count of a Rule matches the value you set within the specified time interval, the `then status` field determines the status of the Rule. You can specify the status to be SUSPEND, DISABLE or NOACTION.

If you specify that the Rule is suspended, the Rule is automatically re-enabled when the matching rate falls below the specified threshold. If you specify that the Rule is disabled, you must manually re-enable the Rule. If you specify to take no action, the Rule will match and the fired count will increase, but no actions specified for that Rule will take place. The action is resumed when the match rate falls below the threshold.

Note: The difference between NOACTION and SUSPEND is that if the Rule Set is using the strategy FIRST, SUSPEND allows a subsequent Rule in the Rule Set to fire for this event whereas NOACTION will fire this Rule and suppress further checking in the Rule Set.

Example

This example shows that if a Rule matches an event 10 times within any 30 second interval, the Rule will be suspended.

```
Criteria match rate threshold:
If matched    ===> 10       (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===> 30       (Interval length, 1-9999 seconds)
then status   ===> SUSPEND  (SUSPEND, DISABLE, NOACTION)
```

Note: When the PAS is cold started, the Rules Status will be as coded in the Rule Set members. Any Rule status changes made as a result of the match rate specifications are discarded.

If a criteria match rate is specified in this Rule Set's Filter, every Rule will have a criteria match rate, either an explicit match rate coded in the Rule or implicit match rate inherited from the filter. With the Rule Set filter, you can limit the number of events and the types of events that are passed through a specific Rule Set. For more details on how to use Rule Set filtering with criteria match rate, see “Enabling Rule Set Filtering and Rule Set Match Rate” on page 190.

In order to bypass Rule Set criteria match rate checking, specify the criteria match rate fields for the Rule, which causes the Rule Set criteria match rate to be ignored. Only the criteria match rate for the Rule will apply.

Note: A value of 0 in the `If matched` field of the Rule Set Match Rate can also be used to ignore the Rule Set Match Rate. Rules that specify 0 will always fire.

Rules whose status has changed to SUSPEND, DISABLE, or NOACTION as a result of a high matching rate are enabled when manually enabled or when the subsystem is cold started. A new actual matching rate is calculated for the preceding interval every time the Rule matches an event.

These fields cannot be set for Time-initiated Rules.

Advanced Rule Processing Techniques

You can use the criteria match rate fields along with an Automation Strategy of FIRST for some advanced Rule Processing techniques. For example, as you create a Rule and specify a match rate to be met, you might want to create a second Rule to take some action when the first Rule is suspended or disabled.

In other words, you can use the criteria match rate fields with the FIRST Automation Strategy to develop a Rule that is disabled or suspended when the match rate is met **and** develop a second Rule to perform an appropriate automation task when the first Rule stops firing. Perhaps this second Rule will write a message to the Journal or perhaps it will send a message to a TSO user ID to notify an operator of the situation.

You can also use this technique to have a Rule fire every *n*th occurrence of an event. For example, you can create a Rule that does not perform any automation actions and specify that it is disabled or suspended when it matches an event 5 times within 30 minutes. Create a second a Rule for the same event and this time, have it take an automation action. You have, in effect, a Rule that fires and takes an automation action every 6th time the event occurs.

Activating and Saving Modified Rules

Saving a Rule occurs when the Rule Set the Rule belongs to is explicitly saved and written to disk.

To save a new Rule or changes made to an existing Rule, you must:

1. Press PF3/END from the Rule Processor Detail Control Panel.

   The Rule Set Overview panel is displayed and any changes you have made to the Selection Criteria, Variable Dependencies, Action Specification, or ALERT Action(s) panel are saved to the Rule Set.

2. Type the primary command `SAVE` on the COMMAND line of the Rule Set Overview panel.

   Typing `SAVE` writes the displayed Rule Set to disk. If you do not save your changes to disk, your changes will be lost at the next subsystem cold start or if you issue the BBI command to reset Rule Sets.

If you press PF3/END from the Rule Set Overview panel **before** you SAVE the Rule Set, Figure 27 is displayed.

```
BMC Software  ------------ Confirm Rule Set Modifications -------- INVALID INPUT
COMMAND ===>                                                TGT  --- BBSYSA


 +-------------------------------------------------------------------------+
 + WARNING! Changes made to Rule Set AAORUL02 have not been saved.  Those     +
 + changes were one or more of the following: +
 +                                                                         +
 +  o A Rule was changed.                                                  +
 +  o The status of a Rule was modified.                                   +
 +  o A Rule was added, deleted, inserted, or copied.                      +
 +  o A Rule was moved.                                                    +
 +  o The individual Rule Set strategy changed.                            +
 +                                                                         +
 +                                                                         +
 + Please do one of the following:                                         +
 +                                                                         +
 + - Enter SAVE to save AAORUL02 to the BBIPARM data set.                   +
 + - Enter NOSAVE  to exit WITHOUT saving AAORUL02 to the BBIPARM data set.  +
 + - Press END to return to Rule Set Overview.                             +
 +-------------------------------------------------------------------------+
```

Figure 27. Example of Confirm Rule Set Panel

If the BBI-SS PAS terminates while you are using the Rule Processor Detail Control panel, a short message `BBI-SS UNAVAILABLE` is displayed when you press PF3. If this occurs, enter the CANCEL command to exit the Rule Processor Detail Control panel. Any changes you might have made will be lost and you must make them again.

Chapter 6. Creating Rules for Events: Using the Rule Creation Panels

The following chapter describes the five panels accessed through the Rule Processor Detail Control panel that are used in Rule creation.

For every Rule you create, you will use a Selection Criteria panel and an Action Specification panel and for each event type, there are different versions of the Selection Criteria and Action Specification panels.

Optionally, you may use a second (additional) selection criteria panel called the Variable Dependencies panel or a second (additional) action specification panel called the Alert Action(s) panel. Certain event types use customized Alert Action(s) panels.

In other words, depending on what events you create a Rule for, different selection criteria items and different actions are displayed on the Selection Criteria panels, Action Specification panels, and ALERT Action(s) panels for that event type. Only the Variable Dependencies panel displays the same information for every event type.

For more information about Rules and events, refer to:

- "Describing Events" on page 43 describes the event types the Rule Processor supports
- "Creating Rules: Examples" on page 105 for examples that show how different types of Rules are created for different event types
- *MAINVIEW AutoOPERATOR for MQSeries Installation and User Guide* for more information about creating Rules for MQSeries events.

Using Primary Commands

You can enter the primary commands shown in Table 11 in the `COMMAND` field of the Rule Processor Detail Control panel (or any of the panels used in Rule creation) as a shortcut to a specific panel. The table describes the command, the panel displayed, and a brief description.

Table 11. Using Primary Commands in the Rule Processor

Command	Displays...	Description
blank (Default)	Displays the Selection Criteria panel when you press ENTER	When you do not specify primary command, pressing ENTER displays the Selection Criteria panel for the event type you specify in the `Event Type` field. By not using the primary commands in this table and only pressing ENTER you progress through the six Rule creation panels.
S1	Selection Criteria - *event type* panel	Use this panel to specify attributes of an event that a Rule must match before the Rule is fired. You must specify at least one criteria.

Table 11. Using Primary Commands in the Rule Processor

Command	Displays...	Description
SV	Variable Dependencies - *event type* panel	Use this panel to set the values of variables when you want to use variables for any of your selection criteria. Use of this panel is optional.
A1	Action Specification - *event type* panel	When the selection criteria of a Rule matches an event and the Rule is fired, the actions specified on this panel are taken by the Rule.
AA	Alert Action(s) I - *event type* panel	Only some of the Rules you create will issue AutoOPERATOR ALERTs as part of the actions the Rule takes. For those Rules, use this panel to add or delete the ALERT to be issued.
AD	Alert Action(s) II - *event type* panel	This panel is a continuation of the Alert Action(s) I panel and includes additional fields that can be used while creating an AutoOPERATOR ALERT from a Rule.

Step 1. Using Selection Criteria Fields

Each event type has its own selection criteria panel and when you create a Rule for an event, the first panel displayed is the `Selection Criteria - event type` panel. The `event type` field shows the name of event type you specified in the `Event Type` field on the Rule Processor Detail Control panel.

Not every selection criteria field can be used for every event type. The second column in the table below shows which event types can use a specific selection criteria field. Depending on which event type you create a Rule for, a panel for that event type appears and will contain a subset of the criteria listed below.

The following figure shows an example of the Selection Criteria panel for a MSG-initiated Rule.

```
BMC Software ------------- Selection Criteria - MSG  ------------ AutoOPERATOR
COMMAND ===>

       Rule-set === AAORUL00             Rule-id  === ICH70001

Text Description:
Text ID     ===> ICH70001                          Ignore leading plus: Y
Text String (Enter Below):


Issuer Identification:
Job name    ===>                                   Name of issuer
Type        ===>                                   (JOB, STC, or TSO)
Jobclass    ===>                                   Job class of issuer
Acct Info   ===>                                   Job accounting information
RACF User   ===>                                   RACF Userid
RACF Group  ===>                                   RACF Group name
Route Codes ===>
Desc Codes  ===>
Console ID  ===>                                   Destination Console
Console Name ==>                                   Destination Console Name

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Describing Selection Criteria Fields

The following table lists all of the available selection criteria fields for all the event types. Note that Time-initiated Rules do not use any of the selection criteria listed here. To see the selection criteria for Time-initiated Rules, refer to “Creating TIME-Initiated Rules” on page 121.

Selection Criteria Field	Event Type	Description
Acct Info	CMD EXT IMS JES3 MSG	Enter up to 33 bytes of accounting information for the issuer of the event. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Alarm ID	ALRM	Enter the Alarm ID associated with the Alarm. SHARED variables, EVENT variables, and pattern matching can be used in this field.
CICS TDQ	CICS	Is the 1- to 4-character name of the CICS Transient Data Queue where the event originated. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Console ID	CMD JES3 MSG	Specify the console ID of a specific MVS console associated with the message. This field is 33 characters long. Valid console ID numbers are 1-255. SHARED variables and EVENT variables can be used in this field.
Console Name	CMD JES3 MSG	Specify the console name of a specific MVS console associated with the message. This field is 33 characters long. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Context	ALRM	Enter the context associated with the Alarm. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Desc codes	JES3 MSG	Enter one or more MVS descriptor codes associated with the message. Separate each descriptor code by a blank. Valid codes are 1-128. Variables and pattern matching cannot be used in this field.
Ignore leading plus	MSG	Enter Y to ignore a leading plus **+** in the message. Messages issued from a non-authorized program will have a plus sign inserted at the beginning of the message. Using this option allows messages that are issued from both authorized and non-authorized programs to be trapped by a single rule. If you specify this option, take additional steps to ensure that the message is being issued from a trusted job, such as by verifying the user ID, job name, accounting code, or some other information provided by the Rule or EXEC. Note: Use this option carefully to ensure that your Rules do not fire accidentally.

Selection Criteria Field	Event Type	Description
Jobclass	CMD EXT IMS JES3 MSG	Specify a one byte identifier for the job class of a batch job issuer. Valid values are 0-9 and A-Z. SHARED variables and EVENT variables can be used in this field.
Job name	CMD DB2 EXT IMS JES3 MSG	Specify a 1- to 8-byte name of the address space that issued the event. This field is usually used with the `Type` field which specifies the type of address space issuing the event. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Key	ALRT	Specify the key used to uniquely identify an ALERT within a queue. Use this field to create Rules for ALERTs written with specific keys. SHARED variables, EVENT variables, and pattern matching can be used in this field. For more information about ALERT keys, refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.
Name	VAR	Specify the name of a SHARED pool variable so that, when its value changes, it will trigger a Rule to fire. SHARED variables can be used in this field.
Origin	JRNL	Enter the name of the address space that issued the message for the BBI-SS PAS Journal. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Priority	ALRM	Enter the priority associated with the Alarm. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Queue	ALRM ALRT	Specify the name of the queue to which the ALERT is targeted. Use this field to create Rules for ALERTs destined for specific queues. SHARED variables, EVENT variables, and pattern matching can be used in this field. For more information about ALERT queues, refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.

Selection Criteria Field	Event Type	Description
RACF Group	CMD EXT IMS JES3 MSG	A 1- to 8-byte field where you can enter the RACF group ID for the address space that issued the message. The RACF Group ID is taken from the GROUP keyword of the job card. SHARED variables, EVENT variables, and pattern matching can be used in this field.
RACF User	CMD EXT IMS JES3 MSG	A 1- to 8-byte field where you can enter the RACF user ID for the address space that issued the message. The RACF user ID is taken from the USERID keyword of the job card. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Route codes	JES3 MSG	Enter one or more MVS route codes associated with the message. Separate each route code by a blank. Valid codes are 1-128. Variables and pattern matching cannot be used in this field.
Scope	ALRM	Enter the scope associated with the Alarm. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Text ID	All event types	Enter the 1- to 16-byte Text ID associated with the message. The length of the Text ID is determined by the first blank character in the text. The Text ID for the message is the first word, delimited by a blank, or the first 16 characters if the first word is longer than 16 characters. To select on more information, use the `Text String` field to enter message text. SHARED variables, EVENT variables, and pattern matching can be used in this field. **Note:** For event type MSG: • If you journal the message, the entire message (including the reply ID) is written to the journal. • If you use IMFTEXT the reply ID is stripped from the message. • WORD1 is the first word after the reply ID. • TEXT STRING does not include the reply ID.

Selection Criteria Field	Event Type	Description
Text String	All event types	Specify message text to be matched in addition to the Text ID. In this field, you must enter the Text ID and additional information from the message text for the Rule to match before it fires. This field supports up to 75 bytes. SHARED variables, EVENT variables, and pattern matching can be used in this field. **Note:** For event type MSG: • If you journal the message, the entire message (including the reply ID) is written to the journal. • If you use IMFTEXT the reply ID is stripped from the message. • WORD1 is the first word after the reply ID. • TEXT ID is the first word after the reply ID. • TEXT STRING does not include the reply ID.
Type	ALRM CMD EXT IMS JES3 MSG	Specify a 1- to 3-byte name for the type of address space that issued the event. Valid values are JOB, STC, or TSO. This field provides additional information about the address space that issued the event. SHARED variables and EVENT variables can be used in this field.
User ID	ALRM	Enter the User ID associated with the Alarm. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Variable is	VAR	Specify the action that will fire the Rule; valid values are: **Created** The Rule will fire when the variable is created in the SHARED pool. **Deleted** The Rule will fire when the variable is deleted from the SHARED pool. **Upd** The Rule will fire when the value of the variable is updated in the SHARED pool.

Step 2. Specifying Variables as Additional Selection Criteria

The second panel displayed is the `Variable Dependencies` - event type panel. The Variable Dependencies panel is the same for all event types, but the `event type` field shows the name of event type you specified in the `Event Type` field on the Rule Processor Detail Control panel.

Use the Variable Dependencies panel to define the value of variables to be tested as part of the selection criteria for an event.

You can use any SHARED or LOCAL variable on this panel.

Note: Refer to the chapter describing variables in the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs* for general information regarding SHARED and LOCAL variables. Refer to "Using Variables in the Rule Processor" on page 161 for more information about using variables in the Rule Processor applications.

You also can use the variable WORD*n*, where *n* is the number of the word in the message. For example, WORD3 represents the third word in the message text.

Figure 28 shows an example of the Variable Dependencies panel.

```
BMC Software ------------ Variable Dependencies - MSG  ---------- AutoOPERATOR
COMMAND ===>

       Rule-set === AAORULXX             Rule-id  === ICH70001

                                                                        OR/
 Variable-name                    Op  Variable-Value                   AND

 QSMFID_____________________________ EQ  SYSB_________________________________  OR_
 QSMFID_____________________________ EQ  SYSC_________________________________  ___
 __________________________________ __  __________________________________  ___
 __________________________________ __  __________________________________  ___
 __________________________________ __  __________________________________  ___
 __________________________________ __  __________________________________  ___
 __________________________________ __  __________________________________  ___
 __________________________________ __  __________________________________  ___
 __________________________________ __  __________________________________  ___
 __________________________________ __  __________________________________  ___
 __________________________________ __  __________________________________  ___
 __________________________________ __  __________________________________  ___
 __________________________________ __  __________________________________

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 28. Variable Dependencies Panel

In Figure 28, the QSMFID variable is specified to equal the value SYSB or SYSC. This means that if you use the variable QSMFID on the Selection Criteria panel for an event, the Rule will fire only when the value of QSMFID matches SYSB or SYSC.

Note: You can code variables with or without the ampersand (`&`) on this panel. The panel displays the `Variable-name` field without the `&` even if it was coded previously.

Using AND/OR Logic on the Variable Dependencies Panel

You can use AND/OR logic on the Variable Dependencies panel to combine multiple variable evaluations.

The logic operates as follows:

- The use of `AND` implies that entries on the panel are grouped together in parenthesis.
- The use of `OR` separates the groups of information.

Figure 29 shows an example.

```
BMC Software ------------ Variable Dependencies - MSG  ---------- AutoOPERATOR
COMMAND ===>

       Rule-set === AAORULXX             Rule-id  === ICH70001

                                                                          OR/
 Variable-name                    Op  Variable-Value                     AND

 QSMFID____________________________ EQ  SYSA_________________________________  AND
 IMFDAY____________________________ EQ  TUE__________________________________  AND
 WORD3_____________________________ EQ  DUMP_________________________________  OR_
 QSMFID____________________________ EQ  SYSB_________________________________  AND
 IMFDAY____________________________ EQ  WED__________________________________  OR_
 QSMFID____________________________ EQ  SYSC_________________________________  ___
 __________________________________ __  _____________________________________  ___
 __________________________________ __  _____________________________________  ___
 __________________________________ __  _____________________________________  ___
 __________________________________ __  _____________________________________  ___
 __________________________________ __  _____________________________________  ___
 __________________________________ __  _____________________________________  ___
 __________________________________ __  _____________________________________

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 29. Variable Dependencies Panel with AND/OR Logic

This information translates into the following statements:

```
[(QSMFID=SYSA) AND (IMFDAY=TUE) AND (WORD3=DUMP)]
             OR
[(QSMFID=SYSB) AND (IMFDAY=WED)]
             OR
       (QSMFID=SYSC)
```

Describing Variable Dependencies Fields

Following is a description of the fields on this panel.

Table 12. Variable Dependencies Panel Field Descriptions

Field	Description		
Variable-Name	Is the name of the SHARED or LOCAL variable to be tested		
Op	Is the operand used in the variable evaluation; valid operands are as follows:		
	Operand	**Character or Hexadecimal**	**Description**
	EQ	Character	Specifies that the variable must equal a specified value.
	NE	Character	Specifies that the variable must not be equal to a specified value.
	GT	Character	Specifies that the variable must be greater in value than a specified value.
	LT	Character	Specifies that the variable must be less in value than a specified value.
	GE	Character	Specifies that the variable must be greater than or equal in value to a specified value.
	LE	Character	Specifies that the variable must be less than or equal in value to a specified value.
	IN	Character	Specifies that the queried variable includes the value specified. For example, use this operand to verify that the text MQRO_COD is contained in the value of the MQSeries variable IMFQ_MD_REPORT (Report Options) which resolves to MQRO_PASS_CORREL_ID MQRO_COD.
	EX	Hexadecimal	Specifies that the queried variable excludes the value specified. For example, use this operand to verify that the text MQRO_COA is excluded from the value of the MQSeries variable IMFQ_MD_REPORT (Report Options) which resolves to MQRO_PASS_CORREL_ID MQRO_COD.
	HE	Hexadecimal	Specifies that the queried variable is equal to the specified value. For example, the MQSeries variable IMFQ_MD_CORRELID (Correlation ID) contains the hex value x'80BE'. The value '80BE' is specified on the Variable Dependencies panel to the right of the operand. When the event occurs, the Rule fires (provided all other selection criteria are also matched) because the queried variable does contain the specified value x'80BE'.

Table 12. Variable Dependencies Panel Field Descriptions

Field	Description		
Op (cont.)	Is the operand used in the variable evaluation; valid operands are as follows:		
	Operand	**Character or Hexadecimal**	**Description**
	HN	Hexadecimal	Specifies that the queried variable is not equal to the specified value. For example, the MQSeries variable IMFQ_MD_CORRELID (Correlation ID) contains the value x'73C4'. The value '73C5' is specified on the Variable Dependencies panel to the right of the operand. When the event occurs, the Rule fires (provided all other selection criteria are also matched) because the queried variable does not contain the specified value x'73C5'.
	HG	Hexadecimal	Specifies that the queried variable is greater than the specified value. For example, the MQSeries variable IMFQ_MD_CORRELID (Correlation ID) contains the value x'80BF'. The value '80BE' is specified on the Variable Dependencies panel to the right of the operand. When the event occurs, the Rule fires (provided all other selection criteria are also matched) because the queried variable is greater than the specified value x'80BE'.
	HL	Hexadecimal	Specifies that the queried variable is less than the specified value. For example, the MQSeries variable IMFQ_MD_CORRELID (Correlation ID) contains the value x'2A20'. The value '2A20' is specified on the Variable Dependencies panel to the right of the operand. When the event occurs, the Rule does not fire because the queried variable is not less than the specified value x'2A20'.
	GX	Hexadecimal	Specifies that the queried variable is greater than or equal to the specified value. For example, the MQSeries variable IMFQ_MD_CORRELID (Correlation ID) contains the value x'2A21'. The value '2A21' is specified on the Variable Dependencies panel to the right of the operand. When the event occurs, the Rule fires (provided all other selection criteria are also matched) because the queried variable is greater than or equal to the specified value x'2A21'.

Table 12. Variable Dependencies Panel Field Descriptions

Field	Description		
Op (cont.)	Is the operand used in the variable evaluation; valid operands are as follows:		
	Operand	**Character or Hexadecimal**	**Description**
	LX	Hexadecimal	Specifies that the queried variable is less than or equal to the specified value. For example, the MQSeries variable IMFQ_MD_CORRELID (Correlation ID) contains the value x'0A21'. The value '0B21' is specified on the Variable Dependencies panel to the right of the operand. When the event occurs, the Rule fires (provided all other selection criteria are also matched) because the queried variable is less than or equal to the specified value x'0B21'.
	FO	Hexadecimal	Specifies the value to be used as a mask to test for bits set in the queried variable. For example, a SHARED variable, FLAG1, contains the value x'00C00080'. The value '00000080' is specified on the Variable Dependencies panel to the right of the operand. When the event occurs, the Rule fires (provided all other selection criteria are also matched) because bit 7 is set to on in both the variable and the value specified.
	FN	Hexadecimal	Specifies the value to be used as a mask to test for bits not set in the queried variable. For example, a SHARED variable, FLAG1, contains the value x'00C00080'. The value '000000A0' is specified on the Variable Dependencies panel to the right of the operand. When the event occurs, the Rule fires (provided all other selection criteria are also matched) because bit 5 is set to on in to the mask specified but not in the variable.
Variable-Value	Is the value the variable is compared against.		
AND/OR	Is the logical operand used to combine multiple values. Valid operands are as follows: **AND** Use AND when you want more than one condition to be true **OR** Use OR when any one of multiple conditions can be true		

Step 3. Using Action Specification Fields

The third panel displayed during Rule creation is the `Action Specification` - event type panel. The event type field shows the name of the event type you specified in the `Event Type` field on the Rule Processor Detail Control panel.

Not every action specification field can be used for every event type. The second column shows which event types can use a specific action specification field. Depending on which event type you create a Rule for, a panel for that event type appears and will contain a subset of the actions listed below.

The following figure shows an example of the Action Specification panel for a MSG-initiated Rule.

```
BMC Software  ------------- Action Specification - MSG  ---------- AutoOPERATOR
COMMAND ===>
       Rule-set === AAORUL00             Rule-id  === ICH70001
Display at dest. ===> YES       Journal ===>           SYSLOG Display ===>
EXEC Name/Parms  ===>
Send (TSO)       ===>
Cmd (Type   MVS) ===>

Set Variable     ===>                                    =
Reword Msg       ===>

Notify           ===>                                   Outboard Pager ID
Info             ===>                                   Info for Outboard

Route codes      ===>
Descriptor codes ===>
Console ID       ===>                                   Destination Console ID
Console Name     ===>                                   Dest. Console ID Name
Update Rout/Desc Codes ===>                             (ADD/REPL)
DOMID            ===>                                   Delete operator message
Issue WTO Msg    ===>

DOM MSG          ===>                                   Issue message when DOM
Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Describing Action Specification Fields

The following table lists all of the available actions you can select for all the event types.

Action Specification Field	Event Type	Description
Console ID	MSG	Specify a message's new destination console ID. This field is 33 characters long. SHARED variables are valid. Only one console ID can be specified. A value from 1-99 is valid.
Console Name	MSG	Specify a message's new destination console name. This field is 33 characters long. SHARED variables are valid. Only one console name can be specified. Console names are 1-8 characters long. Named consoles are available only on MVS/SP4.

Action Specification Field	Event Type	Description

Cmd (Type xxx)	All event types	Enter one or more commands to be issued where xxx represents a user-specified command type (up-to 3 characters). Possible values are as follows:
		
		Blank No command is issued
		
		BBI Allows you to specify BBI commands
		
		CIC Allows you to specify CICS commands or transactions. The CICS command must contain a target; for example:
		
		`TARGET:CICSTRAN command_text ‘parms’`
		
		The CICS transaction parms must be enclosed in single quotation marks; for example:
		
		`Cmd (type CIC) ===> TARGET:CICSTRAN transaction_name 'parm1 parm2'`
		
		For a list of CICS commands you can use, refer to the section "CICS Command Parameters" of the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.
		
		Note: When you use any of the listed commands in this field, do not prefix the command name with IMFEXEC CICS.
		
		Only CICS transactions that run terminal unattached are supported. For terminal attached transactions, use the OSPI application (see *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*).
		
		IMS Allows you to specify IMS commands
		
		MVS Allows you specify MVS console commands in this field; the address space that originated the event will issue the commands
		
		SS Allows you to specify MVS commands that originate from the BBI-SS PAS
		
		MQ Allows you to specify MQSeries commands
		
		In addition:
		
		• This field is 126 characters long
		• EVENT and SHARED variables can be used in this field
		• You can use multiple commands by separating them with double colon marks: ::
		• MQSeries commands must begin with the name of the queue manager that will process the command specified. For example: `queuemgrname:command text`

Action Specification Field	Event Type	Description
Descriptor codes	MSG	A 38-character field that allows you to add new MVS descriptor codes or replace the original MVS descriptor codes of the message. Separate each descriptor code entered with a blank. This field must be used in conjunction the `Update Rout/Desc Codes` field to specify either replacement or adding of new codes. Variables cannot be used in this field. •
Display at dest.	ALRT CICS DB2 IMS JRNL JES3 MSG	Specifies whether the message will be displayed at the destination it was originally sent to. Valid values are YES and NO. YES is the default. If you specify NO, the message is suppressed. Variables cannot be used in this field. **Note:** • If you journal the message, the entire message (including the reply ID) is written to the journal. • If you use IMFTEXT the reply ID is stripped from the message. • WORD1 is the first word after the reply ID. • TEXT ID is the first word after the reply ID. • TEXT STRING does not include the reply ID. **For CICS events**, When any line of a multiline message is suppressed in the TDQ (XTDOUT) exit, all subsequent lines of the multiline message are discarded (not presented to the XTDOUT exit). This means that if a Rule fires that handles a CICS message and has NO specified in this field, all remaining lines of the multiline message will be discarded (not presented to the Rules Processor). This restriction is due to the design of the CICS exit. Given this CICS behavior, you cannot suppress a part of any multiline CICS message and still receive the rest of the multiline message in a Rule.
DOM Id	All event types	Specify the name of a SHARED variable that contains the delete-operator-message ID of a previously issued WTO.
DOM Msg	All event types	Tells AutoOPERATOR to issue message DX9505I when the action message which caused this Rule to fire is deleted.

Action Specification Field	Event Type	Description
EXEC name/Parms	All event types	Allows you to specify the name of an EXEC the Rule can invoke and any parameters that you want to send to the EXEC. This field is 56 characters long. Separate the EXEC name and any parameters by a blank. Note that the first parameter specified in this field will be the first parameter passed to the EXEC. This also means the message ID and any message text will not be passed to the EXEC. To have the message ID and message text passed to the EXECs, use the variable IMFTEXT in this field. EVENT and SHARED variables can be used in this field.
Info	All event types	Specify a 1-80 alphanumeric character string of information to be displayed in the area of an outboard pager managed by AutoOPERATOR Elan. Use this field in conjunction with the `Notify` field. EVENT and SHARED variables can be used in this field.
Issue WTO Msg	All event types	Specify a 1-125 character message to be sent to an MVS console. The message is issued without routing or descriptor codes unless specified with the R (route) or D (descriptor code) keywords (or both), and followed by a colon (:). For example, to issue a message with Descriptor code 2 and Route code 11: `Issue WTO Msg ===> D(2) R(11): EMPMSG88 &WORD1 ISSUED` The message may be selected by another Rule. The default is blank, which means no message is issued. EVENT and SHARED variables are valid. **Note:** For event type MSG: • If you journal the message, the entire message (including the reply ID) is written to the journal. • If you use IMFTEXT the reply ID is stripped from the message. • WORD1 is the first word after the reply ID. • TEXT ID is the first word after the reply ID. • TEXT STRING does not include the reply ID.

Action Specification Field	Event Type	Description
Journal	ALRT CICS CMD DB2 EXT IMS JES3 MSG TIME VAR	Specifies whether the message will be recorded in the BBI-SS PAS Journal. Valid values are YES, NO, and REW. The default is NO. A value of REW copies a reworded message to the BBI-SS PAS Journal. Use REW with the `Reword Msg` field. **Note:** For event type MSG: • If you journal the message, the entire message (including the reply ID) is written to the journal. • If you use IMFTEXT the reply ID is stripped from the message. • WORD1 is the first word after the reply ID. • TEXT ID is the first word after the reply ID. • TEXT STRING does not include the reply ID.
Notify	All event types	Enter the telephone number of the AutoOPERATOR Elan operator ID to be notified using an outboard pager. This field is 33 characters long. EVENT and SHARED variables are valid. Specify the telephone number exactly as it must be dialed. The Elan operator ID must exist in the Elan Operator Profile. Use this field in conjunction with the `Info` field.
Reject Command	CMD	Specifies whether command will be issued or rejected.
Reword ALERT	ALRT	Use this field to change the ALERT text to be sent to its destination. Note that the reworded text is used for all subsequent Rule matching comparisons if you use an Automation strategy of ALL. The length of the reworded text cannot exceed 255 characters. EVENT and SHARED variables are valid in this field. The reworded text is used in subsequent Rule Selection Criteria comparisons.
Reword CMD	CMD	Enter a new command that replaces the original command. All attributes associated with the command such as console ID are retained. The maximum length of a command replacement is 126 characters. This parameter should not be coded for Rules that are used in a release prior to AutoOPERATOR 6.2. Doing so causes this field to be displayed as a message on the console and in syslog.

Action Specification Field	Event Type	Description
Reword Msg	IMS JES3 JRNL MSG CICS	Use this field to change the message text to be sent to its destination. Note that the reworded text is used for all subsequent Rule matching comparisons if you use an Automation strategy of ALL. For the IMS and JES3 event types, the length of the reworded text cannot exceed the length of the original message plus 20 characters. For example, if the length of the original text is 40 characters long, the reworded text cannot exceed 60 characters. For the JRNL and MSG event types, the length of the reworded text cannot exceed the length of the original message. For CICS Transient Data messages, the maximum length of the reworded message is limted to 212 characters. If variables are used, the maximum length is 255 characters (after the contents of the variables are resolved). Any resolved value greater than 255 characters will be truncated to 255 at the time the Rule fires. EVENT and SHARED variables are valid in this field. The reworded message text is used in subsequent Rule Selection Criteria comparisons.
Route codes	MSG	A 61 character field that allows you to add new MVS route codes or replace the original MVS route codes of the message. Separate each route code entered with a blank. This field must be used in conjunction the `Update Rout/Desc Codes` field to specify either replacement or adding of new codes. Variables cannot be used in this field.
Send (TSO)	All event types	Allows you to send a copy of the message using the MVS SEND command to 1-5 TSO user IDs. Separate each TSO user ID by a blank. This field is 40 characters long. EVENT and SHARED variables are valid in this field.
Set variable	All event types	Allows you to set, increment, or decrement the value of a SHARED variable. A variable's numerical value may be incremented or decremented by specifying `+n` or `-n`, where `n` is a number. The first field is 33 characters long and SHARED variables are valid. Use the first field to specify the variable to be modified. The second field is 17 characters long and SHARED variables and constants are valid. Use the second field to assign a new value (with a variable or an unsigned constant) or modify a numeric value (with a signed constant). **Note:** The variables set by using this field are set asynchronously to the firing of the Rule. In other words, there will be a slight delay before the value of the variables are set.

Action Specification Field	Event Type	Description
SYSLOG Display	MSG	Specifies whether the message will be recorded in the MVS SYSLOG. Valid values are YES and NO. YES is the default.
Update Rout/Desc Codes	MSG	A field that allows you add or replace route and descriptor codes specified in the `Route Codes` or `Desc Codes` fields. Valid values are ADD (to add new codes) or REPL (to replace old codes with new ones specified). Variables cannot be used in this field.

Step 4. Creating ALERTs as Additional Action Specification

There may be times when you want a Rule to generate an AutoOPERATOR ALERT as part of the actions a Rule takes. AutoOPERATOR ALERTs are special exception messages that AutoOPERATOR creates for operators to notify them of exceptional situations that might require their intervention and attention.

When you create an ALERT from a Rule, a set of `Alert Action(s) - event type` panels are always displayed after the Action Specification panel; these panels are:

- `Alert Action(s) I - event type`

  This is the primary ALERT action specification panel and, with the exception of the ALRT event type, it provides the same fields for all event types. The panel displayed for ALRT event type has an additional field; refer to "Creating ALERTs for the ALRT Event Type" on page 95 for more information.

- `Alert Action(s) II - event type`

  This is the secondary ALERT action specification panel that contains new functions added in AutoOPERATOR Version 4, Release 1.

  The `event type` field shows the name of event type you specified in the `Event Type` field on the Rule Processor Detail Control panel.

These panels contain the keywords that match the keywords you would use to create an ALERT from an IMFEXEC CLIST or REXX EXEC and all the functions that you can specify in an IMFEXEC CLIST or REXX EXEC are available on these panels.

The following sections describe the Alert Action(s) action specification panels.

Creating ALERTs for All Event Types Except ALRT Events

Figure 30 shows an example of the Alert Action(s) I Specification panel that is displayed for all event types **except ALRT**.

```
BMC Software ------------- Alert Action(s) I - MSG  ------------- AutoOPERATOR
COMMAND ===>                                                 TGT  --- AO62
       Rule-set === RULTEST1             Rule-id  === FT407IA
Function ===>                                 (ADD, DELETE, DELETEQ)
Key      ===>
Text     ===>


Queues   ===>                                   Alert Queue Name(s)
Priority ===>                                   (CRIT,MAJ,MIN,WARN,INFO,CLEAR)
Color    ===>                                   RED,PINK,YEL,DKBL,LTBL,GRE,WHI
PCMD     ===>

System   ===>                                   Return to target after PCMD
Exec     ===>                                   Follow-up EXEC
Help     ===>                                   Associated HELP Panel
Targets  ===>                                   Target System
Udata    ===>                                   User Data
Origin   ===>                                   Origin
User     ===>                                   Userid
Alarm    ===>                                   Sound Alarm (YES/NO)
Publish  ===>                                   Messages (ADD/REPLACE/NO)

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 30. ALERT Action(s) I: Primary Panel for All Events Except ALRT

Note that this panel shows three actions for the Function field:

```
Function ===>                                 (ADD, DELETE, DELETEQ)
```

Possible actions are `ADD`, `DELETE`, and `DELETEQ`. The `Function` field has an additional action (`Modify`) when you are creating a Rule for ALRT (see “Creating ALERTs for the ALRT Event Type” on page 95).

Creating ALERTs for the ALRT Event Type

The following figure shows an example of the ALERT Action(s) I specification panel when you are creating a Rule for the ALRT event type.

```
BMC Software ------------- Alert Action(s) I - ALRT ------------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
       Rule-set === RULCICS              Rule-id  === FT407IA
Function ===>                                  (ADD, DELETE, DELETEQ, MODIFY)
Key      ===>
Text     ===>


Queues   ===>                                  Alert Queue Name(s)
Priority ===>                                  (CRIT,MAJ,MIN,WARN,INFO,CLEAR)
Color    ===>                                  RED,PINK,YEL,DKBL,LTBL,GRE,WHI
PCMD     ===>

System   ===>                                  Return to target after PCMD
Exec     ===>                                  Follow-up EXEC
Help     ===>                                  Associated HELP Panel
Targets  ===>                                  Target System
Udata    ===>                                  User Data
Origin   ===>                                  Origin
User     ===>                                  Userid
Alarm    ===>                                  Sound Alarm (YES/NO)
Publish  ===>                                  Messages (ADD/REPLACE/NO)

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 31. ALERT Action(s) Specification Panel for the ALRT Event Type

The difference between this panel and the one displayed for all other event types is that for an ALRT-initiated Rule, you can choose **an additional keyword** for the `Function` keyword:

```
Function ===>                                (ADD, DELETE, DELETEQ, MODIFY)
```

The `MODIFY` action is available only when the event is ALRT. For more information, refer to “Modifying an ALERT in a Rule: Using the MODIFY Function” on page 103.

Figure 32 shows an example of the secondary panel you can use to create ALERTs for all event types.

```
BMC Software ------------ Alert Action(s) II - MSG -------------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62

       Rule-set === RULCICS     Alert    Rule-id  === FT407IA

Auto Delete          ===>                                       Yes/No
Retain               ===>                                       Yes/No

Escalate Direction   ===>                                       Up/Down
         Interval    ===>
                     ===>
                     ===>
                     ===>
                     ===>
                     ===>
         Disposition ===>                                       Keep/Delete

         Exec    ===>

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 32. ALERT Action(s) II: Secondary ALERT Creation Panel

This panel contains ALERT functions added in AutoOPERATOR Version 4, Release 1; they are:

Auto Delete Allows you to specify that when an action message is DOMed, the associated ALERT is also automatically deleted.

In other words, when you create Rules for action messages that issue ALERTs, you can specify that the ALERT is automatically deleted when the action message is DOMed.

This field appears only when you are creating ALERTs for the MSG event type.

If you do not want to delete operator action messages (perform a DOM), specify No in the the Auto Delete field.

Retain Allows you to specify an ALERT will be retained across BBI-SS PAS restarts (both cold and warm restarts) and MVS IPLs.

Escalate Allows you to specify an ALERT can increase or decrease in severity over user-specified periods of time.

Additional keywords used with Escalate are Interval, Disposition, and Exec.

For more information about these fields, refer to Table 14 on page 100 and for examples of how these fields work, refer to “Creating ALERTs from Rules” on page 142.

To read more about ALERTs, refer to:

- Chapter 12, “Managing by Exception: Using AutoOPERATOR ALERTs” on page 391 for information about using the ALERT Management Facility to handle ALERTs

- The section for the IMFEXEC ALERT command statement in the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs* manuals.

Describing Alert Action(s) I Fields

Following is a description of the fields on the first ALERT Action(s) I panel.

```
BMC Software ------------- Alert Action(s) I - MSG  ------------- AutoOPERATOR
COMMAND ===>                                                TGT  --- AO62
       Rule-set === RULTEST1             Rule-id  === FT407IA
Function ===>                                  (ADD, DELETE, DELETEQ)
Key      ===>
Text     ===>


Queues   ===>                                  Alert Queue Name(s)
Priority ===>                                  (CRIT,MAJ,MIN,WARN,INFO,CLEAR)
Color    ===>                                  RED,PINK,YEL,DKBL,LTBL,GRE,WHI
PCMD     ===>

System   ===>                                  Return to target after PCMD
Exec     ===>                                  Follow-up EXEC
Help     ===>                                  Associated HELP Panel
Targets  ===>                                  Target System
Udata    ===>                                  User Data
Origin   ===>                                  Origin
User     ===>                                  Userid
Alarm    ===>                                  Sound Alarm (YES/NO)
Publish  ===>                                  Messages (ADD/REPLACE/NO)

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Table 13. Alert Action(s) I Panel Field Descriptions

Field	Description
Function	Is the function to be performed. This field is used in conjunction with the `Queue` field where possible values are: **ADD** Adds an ALERT to the specified queue **DELETE** Deletes an ALERT from the specified queue **DELETEQ** Deletes the ALERT and the ALERT queue **MODIFY** Available with the ALRT event type only For the ALRT event type, using MODIFY allows you to change any ALERT specifications **except** Key, Queue, and Target. For more information, refer to “Modifying an ALERT in a Rule: Using the MODIFY Function” on page 103.
Key	Is the 64-alphanumeric ALERT identifier used to uniquely identify the ALERT in the queue. You must specify a unique key for every ALERT you create. If you create a second ALERT with the same key as an already existing ALERT in the queue, the second ALERT will overwrite the first ALERT. This field **cannot be modified** when using the MODIFY function for an ALRT-initiated Rule.

Table 13. Alert Action(s) I Panel Field Descriptions (Continued)

Field	Description
Text	Is the text of an ALERT message sent when the ALERT is generated. The text is shown on the panel of the ALERT Detail application. The text can use any shared or function variable, any &WORDn variables, or literals.
Queue(s)	Is a 33-byte field specifying the name(s) of the ALERT queue(s). This field is used in conjunction with the `Function` field where valid values are ADD (to add an ALERT to the specified queue), DELETE (to delete an ALERT from the specified queue) or DELETEQ (to delete the ALERT and the ALERT queue). This field can contain any shared or function variable. The default is MAIN. You can specify multiple queues by separating them with double colon marks: :: If a queue name is a variable, you must use a third colon; for example: `&IMFOJOB:::MAIN` This field **cannot be modified** when using the MODIFY function for an ALRT-initiated Rule.
Priority	Is the priority of the ALERT. Valid values are CRITICAL, MAJOR, MINOR, WARNING, INFORMATIONAL, and CLEARING. The default is INFO.
Color	Is the color of the alert text. This overrides the color assigned by Alert Priority. Valid colors are RED, PINK, YELLOW, DKBLUE, LTBLUE, GREEN, and WHITE.
PCMD	Is a 142-byte field specifying the primary command associated with the ALERT that can be executed from the ALERT Management Facility. (This field can contain variables which can expand to 252 bytes.)
System	Allows you to issue the SYSTEM command after the PCMD is issued. This passes control back to the target. Valid values are Yes and No; Yes is the default.
Exec	Is a 33-byte field specifying the name of the follow-up EXEC (and any parameters) associated with the ALERT and can be scheduled from the ALERT Management Facility.
Help	Is the name of the help panel associated with the ALERT that can be accessed from the ALERT Management Facility.
Targets	Is the target system the ALERT is sent to. You can specify multiple targets; separate each target name with a blank. This field **cannot be modified** when using the MODIFY function for an ALRT-initiated Rule.
Udata	Is any desired user data string.
Origin	Is a new origin to assign to the ALERT. The default is the SSID where the ALERT was issued.
User	Is the user ID assigned to the ALERT.

Table 13. Alert Action(s) I Panel Field Descriptions (Continued)

Field	Description
Alarm	Specifies whether to emit an audible alarm.
Publish	Specifies whether an ALERT is published and how it is published to connected PATROL Enterprise Manager workstations that have subscribed to receive ALERTs through the General Message Exchange (GME). Possible values are REPLACE (where the previous ALERT is deleted from the workstation before the new ALERT is sent), ADD (where a new ALERT is added to the workstation), and NO (where the ALERT is not written to the workstation). The default setting for Publish is set with the PUBLISH= parameter in BBPARM member AAOALSxx. For more information about setting the default setting for Publish, refer to the *MAINVIEW AutoOPERATOR Customization Guide* where the Dynamic Parameter Manager application is described or where BBPARM members for AutoOPERATOR are documented.

Describing Alert Action(s) II Fields

Following is a description of the fields on the second ALERT Action(s) Specification panel.

```
BMC Software ------------ Alert Action(s) II - MSG  ---------- AutoOPERATOR
COMMAND ===>                                                TGT  --- AO62

       Rule-set === RULCICS     Alert    Rule-id  === FT407IA

Auto Delete           ===>                                      Yes/No
Retain                ===>                                      Yes/No

Escalate Direction    ===>                                      Up/Down
         Interval     ===>
                      ===>
                      ===>
                      ===>
                      ===>
                      ===>
         Disposition  ===>                                      Keep/Delete

         Exec    ===>

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Table 14. Alert Action(s) II Panel Field Descriptions

Field	Description
Auto Delete	**Appears only when you are creating a Rule for MSG event types.** Allows you to specify that, when the original action message is DOMed, the ALERT is deleted automatically.
Retain	Specifies whether or not the ALERT should be created as a non-volatile ALERT. Possible values are YES|NO; the default is NO. Refer to the *MAINVIEW AutoOPERATOR Customization Guide* where the BBPARM member AAOALSxx is documented, and see the documentation for the ALERTNV= parameter for more information about retaining ALERTs. Non-volatile ALERTs are saved to disk and continue to exist across BBI-SS PAS restarts and MVS IPLs. Be aware that creating non-volatile ALERTs is **very expensive** as DASD I/Os are required to create and delete ALERTs. Therefore, non-volatile ALERTs should be used for very specific situations where it is absolutely necessary and meaningful for an ALERT to survive a BBI-SS PAS or MVS restart. This feature cannot be used in conjunction with the Escalate parameter: **they are mutually exclusive**.
Escalate Direction	Allows you to create ALERTs that will either increase or decrease in priority over user-specified periods of time. This parameter cannot be used in conjunction with the RETAIN parameter: **they are mutually exclusive**. Possible values for this parameter are: **Up** Specifies that the ALERT will be upgraded in severity when the time interval elapses. **Down** Specifies that the ALERT will be downgraded in severity when the time interval elapses. The default is Up. When you use the Escalate Direction field, you must specify **at least one time interval** in the Interval field.

Table 14. Alert Action(s) II Panel Field Descriptions (Continued)

Field	Description
Interval	Used with the Escalate direction field. Use this field to specify over what period of time (in minutes) the ALERT will change in priority. You can specify up to six separate intervals of time over which the priority of the ALERT can be changed. You must specify **at least one time interval** or the Escalate parameter will not work. When the final priority is reached, the action specified by the Disposition parameter is taken. In addition, when you want to have an ALERT change in priority, you must always code one interval more than the number of changes. No priority changes occur in the last interval. For example, if you want an ALERT to change from MAJOR to CRITICAL, you must code two interval periods. Refer to the section "Examples of ALERT Escalation" in the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs* for more information.
Disposition	Used with the Escalate direction field. Allows you to specify what will happen to ALERT when the ALERT reaches its final priority. Possible values are: **Keep** Specifies that the ALERT will be kept when the last (or only) interval expires. **Delete** Specifies that the ALERT will not be kept when the last (or only) interval expires. The default is Delete.
Exec	Used with the Escalate direction field. Allows you to specify the name of an EXEC that will be invoked when the ALERT reaches its final priority level. You also can specify parameters that will be passed to the EXEC when it is invoked.

The following table shows the variables available when you create Rules for the ALRT event type.

Table 15. Variables Available for Creating Rules for ALRT Events

Name	Contents	Length/Format
AMFKEY	Key of the ALERT	1-64 / Character
AMFTEXT	Text of the ALERT	0-255 / Character
AMFALARM	Alarm value of the alert	1 / Y (YES) or N (NO)
AMFCOLOR	Color of ALERT	6 / As specified by COLOR parameter

Table 15. Variables Available for Creating Rules for ALRT Events

AMFEDIR	Increase or decrease the priority of the ALERT when it is escalated	1 / Character (U or D)
AMFEDISP	Keep or delete the ALERT at the final escalation level	1 / Character (K or D)
AMFEEXEC	Name of EXEC and EXEC parameters scheduled at final escalation priority	0 to 255 / Character
AMFEINT1 AMFEINT2 AMFEINT3 AMFEINT4 AMFEINT5 AMFEINT6	Number (in minutes) from 0 to 9999	4 / Numeric (or null)
AMFEXEC	EXEC and EXEC parameters associated with the ALERT	0-256 / Character
AMFHELP	Extended Alert member name	8 / Character
AMFIDATE	Date ALERT was issued	9 / DD-MMM-YY
AMFITIME	Time ALERT was issued	8 / hh:mm:ss
AMFORGN	Origin of ALERT	1-8 / Character
AMFPCMD	Primary command specified in ALERT	0-256 / Character
AMFPRIOR	Priority of ALERT	13 / As specified in PRIORITY parameter
AMFPSYS	Value for SYSTEM keyword (could be either YES or NO)	1 / Character (Y or null)
AMFPUB	Value of the Publish keyword when an ALERT is created	2-7/ADD, REPLACE, or NO
AMFQUEUE	Name of queue for ALERT	8 / Character
AMFRTAIN	Specifies whether or not to retain an ALERT across BBI-SS PAS warm and cold starts	1 / Character (Y or N)
AMFSSID	System from which ALERT was issued	8 / Character
AMFTGT	Target to which ALERT was issued	1-8 / Character
AMFUDATA	User data string	0-256 / Character
AMFUSER	Name of the user ID that the ALERT is addressed to	8 / Character

Modifying an ALERT in a Rule: Using the MODIFY Function

For an ALRT event, you can use a Rule to change any of the original ALERT's settings (**except** Key, Queue, and Target) before it appears in an ALERT queue.

The ability to modify an ALERT makes it convenient for you to select already existing ALERTs with which to use the Retain or Escalate functions without having to rewrite the ALERTs. For example, any existing ALERT can be modified by a Rule using the MODIFY function on the Alert Action(s) panel I to:

- Survive a BBI-SS PAS cold start (with the Retain keyword).
- Increase or decrease in severity over user-specified intervals of time (with the Escalate keyword).
- Change any of the ALERT's existing values (except Key, Queue, and Target) to new values.

The following sections describe some of the effects of using the MODIFY function.

If you leave fields blank with MODIFY: If you use the MODIFY function and leave fields for the new ALERT blank, the original ALERT's specifications will be used.

If you enter new values in fields with MODIFY: When you use the MODIFY function and enter new values in a field, the new ALERT will contain the new values **except for the Key, Queue, and Target fields**. These three fields cannot be changed with MODIFY.

You also can enter a variable name where the variable is set to a null value. When you use such a variable, one of two things can happen to the field:

- The value of the field is deleted.
- The value of the field is set back to its default value.

For example, if the original ALERT issues an EXEC and you do not want the new ALERT to schedule an EXEC, enter a null variable in the Exec field. The new ALERT will not schedule an EXEC.

However, if you enter a null variable for the Alarm field while modifying an ALERT, the ALARM keyword uses the default value of NO.

The following table lists which field's values are deleted when a null variable is entered and which field's values are reset to their default setting.

Fields Whose Values Are Deleted	Fields Whose Values Are Reset to Default
Exec Help Interval PCMD Text Udata User	Alarm Color Dispose Escalate Function Origin Priority Retain System

If you change a field related to escalation: If any ALERT field related to escalation is modified, all escalation parameters must be respecified. Please note that the priority of the ALERT affects escalation. If the ALERT priority is changed without specifying new escalation parameters, the modified ALERT will have no escalation.

If you change an ALERT's color: If you MODIFY and specify a color for the priority of the ALERT, the ALERT will appear with the new color. However, if the priority of the ALERT is set to either increase or decrease with the Escalate field, the ALERT reverts to the default color associated with the new priority.

For more information: Refer to "Creating ALERTs from Rules" on page 142 for an example of how you can use the MODIFY function.

Chapter 7. Creating Rules: Examples

This chapter provides examples of how to create:

- Rules that suppress messages from appearing on the console
- TIME-initiated Rules
- Rules that issue AutoOPERATOR ALERTs
- Rules that use variables

The *MAINVIEW AutoOPERATOR for MQSeries Installation and User Guide* contains information about how to create Rules for MQSeries events.

Creating a Rule to Suppress a Message

The following provides two examples of Rules that suppress the `$HASP395 job ENDED` and IEF450I messages from the console. These messages were chosen because they often appear on the console and can be easily automated by Rules.

The first example shows suppressing the $HASP395 message from the console if it was issued for any job beginning with BAO.

The second example shows searching the IEF450I message only when it contains an abend code and a specific jobname. When the two conditions are met, a message will be sent to a TSO user ID.

Example 1: Creating a Suppression Rule

To write a Rule that suppresses the $HASP395 message:

1. From the Automation Control panel, enable the Rule Set you want to add the Rule to. You can add Rules only to enabled Rule Sets.

```
BMC Software ---------------- Automation Control ------------ RULMAGIE ENABLED
COMMAND ===>                                                  TGT ===> AO62
Primary commands: Add, Statshow, Cmdshow                      DATE --- 01/02/14
                                                              TIME --- 14:14:39
Automation Status     ===> ACTIVE         (Active, Inactive)
Automation Strategy   ===> INDIVIDUAL     (Individual, All, First, Qualified)
Honor MPF Suppression ===> NO             (NO/YES)

                       Automation Statistics
  Total Events              48,538   Display suppressed             67
  Events Handled            22,407   Hardcopy suppressed             0
  Current arrival rate    2  / sec   Rule generated Alerts      22,223
  Peak arrival rate      99  / sec   Rule invoked Execs            276
                        Automation Library
LC CMDS --- (S)elect, (E)nable, (D)isable, (T)est, (SA)ve
            (M)ove, (B)efore or (A)fter, (F)ilter Criteria
LC   Rule-Set Status   Rules     Fired  Filtered    Date      Time   Strategy
___  AAORUL00 ENABLED     15    22,223    48,573  23-FEB-97 08:18:59 FIRST
___  AAORULBC ENABLED     54       188    48,573  23-FEB-97 08:18:59 FIRST
___  AAORULCM ENABLED     52        59    48,573  23-FEB-97 08:18:59 FIRST
___  RULCICS  ENABLED     25         1    48,573  23-FEB-97 08:19:00 FIRST
e_   RULMAGIE DISABLED     8         0        38  23-FEB-97 14:14:38 FIRST
___  AAORULBA DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULBB DISABLED   N/A   N/A       N/A          N/A    N/A
         ********************** END OF DATA **********************
```

Figure 33. Creating a Suppression Rule (1): Automation Control Panel (Example 1)

The Rule Set is enabled.

2. Select the Rule Set with the S line command (for `(S)elect`) in the `LC` field next to the Rule Set name.

```
BMC Software ---------------- Automation Control ------------ RULMAGIE ENABLED
COMMAND ===>                                                  TGT ===> AO62
Primary commands: Add, Statshow, Cmdshow                      DATE --- 01/02/14
                                                              TIME --- 14:14:39
Automation Status     ===> ACTIVE         (Active, Inactive)
Automation Strategy   ===> INDIVIDUAL     (Individual, All, First, Qualified)
Honor MPF Suppression ===> NO             (NO/YES)

                        Automation Statistics
  Total Events              48,538   Display suppressed            67
  Events Handled            22,407   Hardcopy suppressed            0
  Current arrival rate    2  / sec   Rule generated Alerts      22,223
  Peak arrival rate      99  / sec   Rule invoked Execs           276
                         Automation Library
LC CMDS --- (S)elect, (E)nable, (D)isable, (T)est, (SA)ve
            (M)ove, (B)efore or (A)fter, (F)ilter Criteria
LC   Rule-Set Status   Rules     Fired  Filtered    Date      Time   Strategy
___  AAORUL00 ENABLED     15    22,223    48,573  23-FEB-97 08:18:59 FIRST
___  AAORULBC ENABLED     54       188    48,573  23-FEB-97 08:18:59 FIRST
___  AAORULCM ENABLED     52        59    48,573  23-FEB-97 08:18:59 FIRST
___  RULCICS  ENABLED     25         1    48,573  23-FEB-97 08:19:00 FIRST
s_    RULMAGIE ENABLED      8         0        38  23-FEB-97 14:14:38 FIRST
___  AAORULBA DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULBB DISABLED   N/A   N/A       N/A          N/A    N/A
          ********************** END OF DATA **********************
```

Figure 34. Creating a Suppression Rule (1): Automation Control Panel (Example 2)

The Rule Set Overview panel is displayed. Figure 35 shows an example of a Rule Set named RULMAGIE and the Rules it contains.

```
BMC Software ----------------- Rule Set Overview ---------------- AutoOPERATOR
COMMAND ===> add                                                 TGT  --- AO62
Rule Set ID: RULMAGIE    Ruleset Strategy ===> FIRST            DATE --- 01/01/01/02/23
Primary commands: Add, Save, Sort, Unsort, Reset, Filter        TIME --- 14:15:11
LC CMDS --- (S)elect, (E)nable, (D)isable, (T)est, (DE)lete, (I)nsert
            (C)opy/(CC)opy, (M)ove/(MM)ove, (B)efore or (A)fter, (R)epeat
Sort Criterion:                                                   right/left

LC  Rule-id Stat Text-id          Type   Fired EXEC        Changed      ID
    ________ _   _________________ ____         _________ ________ _____ ________
___ DUMMY001 DIS $HASP*           MSG        0           01/02/21 14:58 BAOMXY2
___ RUL00001 ENA *TEST            JRNL       0           01/02/21 15:00 BAOMXY2
___ RUL98888 ENA $HASP395         MSG        0           01/02/21 15:20 BAOMXY2
___ RUL98887 ENA IEF450I          MSG        0           01/02/21 14:07 MES1
___ RUL98886 ENA                  TIME       0           01/02/21 14:14 MES1
___ RUL98885 SUS                  TIME       0           01/02/21 14:15 MES1
___ *TEST    ENA *TEST            ALRT       0           01/02/21 14:43 MES1
___ TESTRULE ENA IEF403I          MSG        0           01/02/21 17:47 BAOMXY2
         ************************* END OF DATA **************************
```

Figure 35. Creating a Suppression Rule (1): Rule Set Overview Panel (Example 1)

3. To add a new Rule, use the ADD primary command on the COMMAND line.

The Rule Processor Detail Control panel is displayed.

```
BMC Software ------------ Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
                                                              TIME --- 14:23:18
  The following options are displayed in sequence, or may    DATE --- 01/02/23
  be selected by entering the two-character code

   S1 - Selection Criteria            A1 - Action Specification
   SV - Variable Dependencies         AA - Alert Actions(s) I
                                      AD - Alert Actions(s) II
Rule ID       ===>
Event Type    ===>          Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status   ===>          (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===>          Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===>
Last Modified by          on          at

  Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

4. Enter the name of the new Rule (the Rule ID) and the event type of the new Rule.

   This example shows adding a new Rule named NEWRULE1 with an event type of MSG. See Figure 36.

```
BMC Software ------------ Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
                                                              TIME --- 14:23:18
  The following options are displayed in sequence, or may    DATE --- 01/02/23
  be selected by entering the two-character code

   S1 - Selection Criteria            A1 - Action Specification
   SV - Variable Dependencies         AA - Alert Actions(s) I
                                      AD - Alert Actions(s) II
Rule ID       ===> NEWRULE1
Event Type    ===> MSG      Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status   ===>          (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===>          Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===>
Last Modified by          on          at

  Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

Figure 36. Creating a Suppression Rule (1): Enter Rule Information on the Rule Processor Detail Control Panel

5. Press ENTER; the Selection Criteria - MSG panel is displayed (Figure 37).

   Specify the criteria the Rule must match with the event to cause the Rule to fire.

```
BMC Software ------------- Selection Criteria - MSG  ------------ AutoOPERATOR
COMMAND ===>                                                TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === NEWRULE1

Text Description:
Text ID     ===>                                   Ignore leading plus: Y
Text String (Enter Below):


Issuer Identification:
Job name    ===>                                   Name of issuer
Type        ===>                                   (JOB, STC, or TSO)
Jobclass    ===>                                   Job class of issuer
Acct Info   ===>                                   Job accounting information
RACF User   ===>                                   RACF Userid
RACF Group  ===>                                   RACF Group name
Route Codes ===>
Desc Codes  ===>
Console ID  ===>                                   Destination Console
Console Name ==>                                   Destination Console Name

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 37. Creating a Suppression Rule (1): Selection Criteria Panel

6. Enter $HASP395 in the `Text-ID` field and BAO* in the `Job Name` field..

```
BMC Software ------------- Selection Criteria - MSG  ------------ AutoOPERATOR
COMMAND ===>                                                TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === NEWRULE1

Text Description:
Text ID     ===> $HASP395                         Ignore leading plus: Y
Text String (Enter Below):


Issuer Identification:
Job name    ===> BAO*                              Name of issuer
Type        ===>                                   (JOB, STC, or TSO)
Jobclass    ===>                                   Job class of issuer
Acct Info   ===>                                   Job accounting information
RACF User   ===>                                   RACF Userid
RACF Group  ===>                                   RACF Group name
Route Codes ===>
Desc Codes  ===>
Console ID  ===>                                   Destination Console
Console Name ==>                                   Destination Console Name

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

7. Press ENTER; the Variable Dependencies - MSG panel is displayed (Figure 38).

```
BMC Software ------------ Variable Dependencies - MSG  ---------- AutoOPERATOR
COMMAND ===>                                                TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === NEWRULE1

                                                                        OR/
 Variable-name                    Op  Variable-Value                   AND

 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________
```

Figure 38. Creating a Suppression Rule (1): Variable Dependencies Panel

8. This example does not make use of variables; press ENTER.

   The Action Specification - MSG panel is displayed. Specify the actions the Rule will take when it is fired.

```
BMC Software ------------- Action Specification - MSG  ---------- AutoOPERATOR
COMMAND ===>                                                TGT  --- AO62
       Rule-set === RULMAGIE             Rule-id  === NEWRULE1
Display at dest. ===> YES       Journal ===>           SYSLOG Display ===>
EXEC Name/Parms  ===>
Send (TSO IDs)   ===>
Cmd (Type  MVS ) ===>

Set Variable     ===>                                    ===>
Reword Msg       ===>

Notify           ===>                                   Outboard Pager ID
Info             ===>

Console ID       ===>                                   Destination Console ID
Console Name     ===>                                   Dest. Console ID Name
Route codes      ===>
Descriptor codes ===>
Update Rout/Desc Codes ===>                             (ADD/REPL)
DOM Id           ===>                                   Delete Operator Message
Issue WTO Msg    ===>

DOM Message      ===>                                   (Yes/No)
Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

9. To specify that the Rule is suppressed from the console, written to the BBI-SS PAS Journal, and a notification is sent to a TSO user ID:
   - Enter NO for `Display at dest.` field to suppress the message from appearing on the console.
   - Enter YES for `Journal` to write the message to the BBI-SS PAS Journal.

     Note: For event type MSG, if you journal the message, the entire message (including the reply ID) is written to the journal.
   - Enter the TSO user ID in the `Send (TSO IDs)` field to send a user ID a notification.

     See Figure 39.

```
BMC Software ------------- Action Specification - MSG  ---------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
       Rule-set === RULMAGIE             Rule-id  === NEWRULE1
Display at dest. ===> NO        Journal ===> YES       SYSLOG Display ===>
EXEC Name/Parms  ===>
Send (TSO IDs)   ===> BAOMES1
Cmd (Type  MVS ) ===>

Set Variable     ===>                                    ===>
Reword Msg       ===>

Notify           ===>                                   Outboard Pager ID
Info             ===>

Console ID       ===>                                   Destination Console ID
Console Name     ===>                                   Dest. Console ID Name
Route codes      ===>
Descriptor codes ===>
Update Rout/Desc Codes ===>                             (ADD/REPL)
DOM Id           ===>                                   Delete Operator Message
Issue WTO Msg    ===>

DOM Message      ===>                                   (Yes/No)
Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 39. Creating a Suppression Rule (1): Action Specification Panel

10. Press ENTER; the Alert Action(s) I - MSG panel is displayed (Figure 40).

```
BMC Software ------------- Alert Action(s) I - MSG  ------------- AutoOPERATOR
COMMAND ===>                                                TGT  --- AO62
       Rule-set === RULMAGIE             Rule-id  === NEWRULE1
Function ===>                                  (ADD, DELETE, DELETEQ)
Key      ===>
Text     ===>


Queues   ===>                                  Alert Queue Name(s)
Priority ===>                                  (CRIT,MAJ,MIN,WARN,INFO,CLEAR)
Color    ===>                                  RED,PINK,YEL,DKBL,LTBL,GRE,WHI
PCMD     ===>

System   ===>                                  Return to target after PCMD
Exec     ===>                                  Follow-up EXEC
Help     ===>                                  Associated HELP Panel
Targets  ===>                                  Target System
Udata    ===>                                  User Data
Origin   ===>                                  Origin
User     ===>                                  Userid
Alarm    ===>                                  Sound Alarm (YES/NO)
Publish  ===>                                  Messages (ADD/REPLACE/NO)

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 40. Creating a Suppression Rule (1): Alert Action(s) I Panel

11. This Rule will not issue an ALERT; press Enter and the Alert Action(s) II - MSG panel is displayed (Figure 41).

```
BMC Software ------------ Alert Action(s) II - MSG  ------------- AutoOPERATOR
COMMAND ===>                                                TGT  --- AO62

       Rule-set === RULMAGIE    Alert    Rule-id  === NEWRULE1

Auto Delete           ===>                                   Yes/No
Retain                ===>                                   Yes/No

Escalate Direction    ===>                                   Up/Down
         Interval     ===>
                      ===>
                      ===>
                      ===>
                      ===>
                      ===>
         Disposition  ===>                                   Keep/Delete

         Exec    ===>

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 41. Creating a Suppression Rule (1): Alert Action(s) II Panel

12. Press ENTER; the Rule Processor Detail Control panel is re-displayed.

```
BMC Software ------------ Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
                                                              TIME --- 16:09:31
  The following options are displayed in sequence, or may    DATE --- 01/02/23
  be selected by entering the two-character code

   S1 - Selection Criteria            A1 - Action Specification
   SV - Variable Dependencies         AA - Alert Actions(s) I
                                      AD - Alert Actions(s) II
Rule ID       ===> NEWRULE1
Event Type    ===> MSG      Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status   ===>          (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===>          Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===>
Last Modified by          on          at

  Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

13. Press END to apply changes.

    The Rule Set Overview panel is displayed (Figure 42).

```
BMC Software ----------------- Rule Set Overview ---------------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
Rule Set ID: RULMAGIE    Ruleset Strategy ===> FIRST          DATE --- 01/02/23
Primary commands: Add, Save, Sort, Unsort, Reset, Filter      TIME --- 16:13:37
LC CMDS --- (S)elect, (E)nable, (D)isable, (T)est, (DE)lete, (I)nsert
            (C)opy/(CC)opy, (M)ove/(MM)ove, (B)efore or (A)fter, (R)epeat
Sort Criterion:                                                right/left

LC  Rule-id Stat Text-id          Type   Fired EXEC        Changed      ID
    ________ _   _________________ ____        _________ ________ _____ ________
___ NEWRULE1 ENA $HASP395         MSG        0          01/02/21 15:00 BAOMXY2
___ RUL98888 ENA $HASP395         MSG        2 ES       01/02/23 16:12 BAOMXY2
___ RUL98887 ENA IEF450I          MSG        0          01/02/21 14:07 MES1
___ RUL98886 ENA                  TIME       7          01/02/21 14:14 MES1
___ RUL98885 SUS                  TIME       0          01/02/21 14:15 MES1
___ *TEST    ENA *TEST            ALRT       0          01/02/21 14:43 MES1
        ************************* END OF DATA *************************
```

Figure 42. Creating a Suppression Rule (1): Rule Set Overview Panel (Example 2)

The Rule is enabled and will begin to fire when the $HASP395 message is issued from any job beginning with the characters BAO but at this point it is not saved to disk.

14. To save the Rule, enter the Save primary command on the command line.

    If you do not enter the Save primary command, the following warning is displayed when you press PF3:

```
BMC Software ----------- Confirm Rule Set Modifications --------- AutoOPERATOR
COMMAND ===>                                                TGT  --- AO62


+-------------------------------------------------------------------------+
+ WARNING! Changes made to Rule Set RULMAGIE have not been saved.  Those    +
+ changes were one or more of the following:                               +
+                                                                          +
+  o A Rule was changed.                                                   +
+  o The status of a Rule was modified.                                    +
+  o A Rule was added, deleted, inserted, or copied.                       +
+  o A Rule was moved.                                                     +
+  o The individual Rule Set strategy changed.                             +
+                                                                          +
+                                                                          +
+ Please do one of the following:                                          +
+                                                                          +
+ - Enter SAVE to save RULMAGIE to the BBIPARM dataset.                    +
+ - Enter NOSAVE  to exit WITHOUT saving RULMAGIE to the BBIPARM dataset.  +
+ - Press END to return to Rule Set Overview.                              +
+-------------------------------------------------------------------------+
```

Figure 43. Creating a Suppression Rule (1): Confirming Rule Set Modifications Panel

The options are:

– To save the Rule, enter SAVE.

– To cancel saving the newly created Rule, enter NOSAVE.

– To return to the Rule Set Overview panel, press END.

Example 2: Creating a Suppression Rule with Variables

This example shows how to suppress the IEF450I message when it states the job BCIS0001 abends with a code of S522. This example also shows that the Rule will notify a TSO user ID when the Rule fires. The Rule will be added to the RULMAGIE Rule Set (used in Example 1).

To create this Rule:

1. Enter the ADD primary command on the Rule Set Overview panel.

```
BMC Software ----------------- Rule Set Overview ---------------- AutoOPERATOR
COMMAND ===> add                                                 TGT  --- AO62
Rule Set ID: RULMAGIE    Ruleset Strategy ===> FIRST            DATE --- 01/02/23
Primary commands: Add, Save, Sort, Unsort, Reset, Filter        TIME --- 14:15:11
LC CMDS --- (S)elect, (E)nable, (D)isable, (T)est, (DE)lete, (I)nsert
            (C)opy/(CC)opy, (M)ove/(MM)ove, (B)efore or (A)fter, (R)epeat
Sort Criterion:                                                   right/left

LC  Rule-id Stat Text-id          Type   Fired EXEC        Changed      ID
    ________ _   _________________ ____        _________ ________ _____ ________
___ DUMMY001 DIS $HASP*           MSG        0          01/02/21 14:58 BAOMXY2
___ RUL00001 ENA *TEST            JRNL       0          01/02/21 15:00 BAOMXY2
___ RUL98888 ENA $HASP395         MSG        0          01/02/21 15:20 BAOMXY2
___ RUL98887 ENA IEF450I          MSG        0          01/02/21 14:07 MES1
___ RUL98886 ENA                  TIME       0          01/02/21 14:14 MES1
___ RUL98885 SUS                  TIME       0          01/02/21 14:15 MES1
___ *TEST    ENA *TEST            ALRT       0          01/02/21 14:43 MES1
___ TESTRULE ENA IEF403I          MSG        0          01/02/21 17:47 BAOMXY2
___ NEWRULE1 ENA $HASP00          MSG        0          01/02/26 10:32 BAOMXY2
         ************************* END OF DATA **************************
```

The Rule Processor Detail Control panel is displayed (Figure 44).

```
BMC Software ------------ Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                 TGT  --- AO62
                                                             TIME --- 14:23:18
  The following options are displayed in sequence, or may    DATE --- 01/02/23
  be selected by entering the two-character code

   S1 - Selection Criteria            A1 - Action Specification
   SV - Variable Dependencies         AA - Alert Actions(s) I
                                      AD - Alert Actions(s) II
Rule ID       ===>
Event Type    ===>          Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status   ===>          (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===>          Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===>
Last Modified by          on          at

  Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

Figure 44. Creating a Suppression Rule (2): Enter Rule Information on the Rule Processor Detail Control Panel

2. Enter the new Rule ID and the event type on the Rule Processor Detail Control panel. .

This example shows adding a new Rule named NEWRULE2 with an event type of MSG.

```
BMC Software ------------ Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
                                                              TIME --- 15:03:51
  The following options are displayed in sequence, or may     DATE --- 01/02/21
  be selected by entering the two-character code

   S1 - Selection Criteria            A1 - Action Specification
   SV - Variable Dependencies         AA - Alert Actions(s) I
                                      AD - Alert Actions(s) II
Rule ID       ===> NEWRULE2
Event Type    ===> MSG      Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status   ===>          (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===>          Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===>
Last Modified by BAOMXY2  on 01/02/21 at 14:07

  Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

3. Press ENTER; the Selection Criteria - MSG panel is displayed (Figure 45).

   Specify the criteria the Rule must match with the event to cause the Rule to fire.

```
BMC Software ------------- Selection Criteria - MSG  ------------ AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === NEWRULE2

Text Description:
Text ID     ===> IEF450I                           Ignore leading plus: Y
Text String (Enter Below):


Issuer Identification:
Job name    ===>                                  Name of issuer
Type        ===>                                  (JOB, STC, or TSO)
Jobclass    ===>                                  Job class of issuer
Acct Info   ===>                                  Job accounting information
RACF User   ===>                                  RACF Userid
RACF Group  ===>                                  RACF Group name
Route Codes ===>
Desc Codes  ===>
Console ID  ===>                                  Destination Console
Console Name ==>                                  Destination Console Name

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 45. Creating a Suppression Rule (2): Selection Criteria Panel

4. Enter IEF450I (the text-ID) of the message in the `Text ID` field.

Press ENTER; the Variable Dependencies - MSG panel is displayed (Figure 46).

```
BMC Software ------------ Variable Dependencies - MSG  ---------- AutoOPERATOR
COMMAND ===>                                                TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === NEWRULE2

                                                                          OR/
 Variable-name                    Op  Variable-Value                     AND
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 46. Creating a Suppression Rule (2): Variable Dependencies Panel

5. To specify that the Rule will fire only when the IEF450I message states that job BCIS001 has abended with an S522 code, use variables on the Variable Dependencies panel as selection criteria.

   Fill in the panel as shown.

```
BMC Software ------------ Variable Dependencies - MSG  ---------- AutoOPERATOR
COMMAND ===>                                                TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === NEWRULE2

                                                                          OR/
 Variable-name                    Op  Variable-Value                     AND

 WORD5___________________________ EQ  ABEND=S522_____________________  AND
 IMFOJOB_________________________ EQ  BCIS001_______________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

The following describes how the specifications on this panel will cause the Rule to work:

– The variable `WORD5` means the Rule Processor scans the IEF450I message for the fifth word to see if it resolves to `ABEND=S522`.

– The Rule Processor will check the variable IMFOJOB to see if the job name is BCIS001.

The Rule will fire only when both these conditions are true. If one of the variables does not match the variable-value specified above, the Rule will not fire.

6. Press ENTER; the Action Specifications - MSG panel is displayed (Figure 47).

Specify the actions the Rule will take when it is fired.

```
BMC Software ------------- Action Specification - MSG  ---------- AutoOPERATOR
COMMAND ===>                                                TGT  --- AO62
       Rule-set === RULMAGIE             Rule-id  === NEWRULE2
Display at dest. ===> NO        Journal ===> YES       SYSLOG Display ===>
EXEC Name/Parms  ===>
Send (TSO IDs)   ===> BAOMES1
Cmd (Type  MVS ) ===>

Set Variable     ===>                                    ===>
Reword Msg       ===>

Notify           ===>                                   Outboard Pager ID
Info             ===>

Console ID       ===>                                   Destination Console ID
Console Name     ===>                                   Dest. Console ID Name
Route codes      ===>
Descriptor codes ===>
Update Rout/Desc Codes ===>                             (ADD/REPL)
DOM Id           ===>                                   Delete Operator Message
Issue WTO Msg    ===>

DOM Message      ===>                                   (Yes/No)
Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 47. Creating a Suppression Rule (2): Action Specification Panel

7. To specify that the message is suppressed from the console, written to the BBI-SS PAS Journal, and a notification is sent to a TSO user ID:

– Enter NO for `Display at dest.` and the IEF450I message will be suppressed from the console.

– Enter YES for `Journal` to have the message written to the BBI-SS PAS Journal.

Note: For event type MSG, if you journal the message, the entire message (including the reply ID) is written to the journal.

– Enter the user ID in the `Send (TSO IDs)` field and a message will be sent to the user ID that states the Rule has fired.

8. Press ENTER; the Alert Action(s) I - MSG panel is displayed (Figure 48).

```
BMC Software ------------- Alert Action(s) I - MSG  ------------- AutoOPERATOR
COMMAND ===>                                                TGT  --- AO62
       Rule-set === RULMAGIE             Rule-id  === NEWRULE2
Function ===>                                  (ADD, DELETE, DELETEQ)
Key      ===>
Text     ===>


Queues   ===>                                  Alert Queue Name(s)
Priority ===>                                  (CRIT,MAJ,MIN,WARN,INFO,CLEAR)
Color    ===>                                  RED,PINK,YEL,DKBL,LTBL,GRE,WHI
PCMD     ===>

System   ===>                                  Return to target after PCMD
Exec     ===>                                  Follow-up EXEC
Help     ===>                                  Associated HELP Panel
Targets  ===>                                  Target System
Udata    ===>                                  User Data
Origin   ===>                                  Origin
User     ===>                                  Userid
Alarm    ===>                                  Sound Alarm (YES/NO)
Publish  ===>                                  Messages (ADD/REPLACE/NO)

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 48. Creating a Suppression Rule (2): Alert Action(s) I Panel

9. This Rule will not issue an ALERT; press Enter and the Alert Action(s) II - MSG panel is displayed (Figure 49).

```
BMC Software ------------ Alert Action(s) II - MSG  ------------- AutoOPERATOR
COMMAND ===>                                                TGT  --- AO62

       Rule-set === RULMAGIE    Alert    Rule-id  === RUL98888

Auto Delete           ===>                                    Yes/No
Retain                ===>                                    Yes/No

Escalate Direction    ===>                                    Up/Down
         Interval     ===>
                      ===>
                      ===>
                      ===>
                      ===>
                      ===>
         Disposition  ===>                                    Keep/Delete

         Exec    ===>

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 49. Creating a Suppression Rule (2): Alert Action(s) II Panel

10. Press ENTER; the Rule Processor Detail Control panel is re-displayed (Figure 50).

```
BMC Software ------------ Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
                                                              TIME --- 16:09:31
  The following options are displayed in sequence, or may    DATE --- 01/02/23
  be selected by entering the two-character code

   S1 - Selection Criteria           A1 - Action Specification
   SV - Variable Dependencies        AA - Alert Actions(s) I
                                     AD - Alert Actions(s) II
Rule ID       ===> NEWRULE2
Event Type    ===> MSG      Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status   ===>          (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===>          Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===>
Last Modified by          on          at

  Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

Figure 50. Creating a Suppression Rule (2): Rule Processor Detail Control Panel

11. Press END to apply changes.

    The Rule Set Overview panel is displayed.

12. Repeat Step 13 on page 113 and Step 14 on page 114 to complete saving the Rule NEWRULE2 to Rule Set RULMAGIE.

Creating TIME-Initiated Rules

The following sections provide examples of TIME-initiated Rules, which fire based on what time it is.

Example 1: Creating Simple TIME-Initiated Rules

This example shows how to create a TIME-initiated Rule that begins to fire at midnight and fires every 15 minutes thereafter. The action the Rule takes is to display initiators.

To create this Rule:

1. Enter the new Rule ID and the event type on the Rule Processor Detail Control panel.

```
BMC Software ------------ Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
                                                              TIME --- 15:05:34
  The following options are displayed in sequence, or may    DATE --- 01/02/21
  be selected by entering the two-character code

   S1 - Selection Criteria            A1 - Action Specification
   SV - Variable Dependencies         AA - Alert Actions(s) I
                                      AD - Alert Actions(s) II
Rule ID       ===> NEWRULE3
Event Type    ===> TIME     Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status   ===>          (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===>          Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===> DISPLAY INITS EVERY 15
Last Modified by BAOMXY2  on 01/02/21 at 14:14

  Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

2. Press ENTER; the Selection Criteria - TIME panel is displayed (Figure 51).

Specify the start time for the Rule to fire and the time interval.

```
BMC Software ------------- Selection Criteria - TIME ------------ AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
                                                              TIME --- 15:05:45
       Rule-set === RULMAGIE             Rule-id  === NEWRULE3

Start Specification:
Start Time           ===> 00:00:01          hh:mm:ss or ATSTART

Frequency:
Interval             ===> 00:15:00          hh:mm:ss

Stop Specification:
Stop time            ===> 24:00:00          hh:mm:ss
          or
Stop Count           ===>


Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 51. Creating a TIME-Initiated Rule (1): Selection Criteria Panel

3. Enter the time specifications as follows:

   – To specify that the Rule begins firing at midnight, specify `00:00:01` for the `Start Time` field.

     A Rule cannot begin firing at `00:00:00`. The earliest after midnight a Rule can begin firing is one second after midnight.

   – To specify that the Rule fires every 15 minutes, enter `00:15:00` for the `Interval` field.

   – To specify the Rule should fire all day and all night, specify `24:00:00` for the `Stop time` field.

   With these specifications, the Rule will fire for the first time at 15 minutes and 1 second after midnight. A message (with message ID AV9999I) appears in the BBI-SS PAS Journal indicating that the Rule fired. The Rule will fire once every 15 minutes and display initiators in the SYSLOG.

4. Press ENTER; the Variable Dependencies - TIME panel is displayed (Figure 52).

```
BMC Software ------------ Variable Dependencies - TIME ---------- AutoOPERATOR
COMMAND ===>                                                TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === NEWRULE3

                                                                          OR/
 Variable-name                    Op  Variable-Value                     AND

 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 52. Creating a TIME-Initiated Rule (1): Variable Dependencies Panel

5. This example does not make use of variables; press ENTER.

   The Action Specification - TIME panel is displayed (Figure 53). Specify the actions the Rule will take when it is fired.

```
BMC Software ------------- Action Specification - TIME ---------- AutoOPERATOR
COMMAND ===>                                                TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === NEWRULE3

Automation Actions:
Journal          ===>
EXEC Name/Parms  ===>
Send (TSO IDs)   ===> BAOMES1
Cmd (Type  MVS ) ===> $DI

Set Variable     ===>                                    ===>


Notify           ===>                                  Outboard Pager ID
Info             ===>

DOM Id           ===>                                  Delete Operator Message
Issue WTO Msg    ===>



Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 53. Creating a TIME-Initiated Rule (1): Action Specification Panel

6. To specify that a notification is sent to a TSO user ID, enter the user ID name in the `Send (TSO IDs)` field.

To specify that the command the Rule issues when it is fired, enter the command in the `CMD (Type MVS)` field. In this case the MVS command is:

```
$DI
```

which will cause the initiators to be displayed in the SYSLOG.

This command will be issued every 15 minutes and a notification will be sent to TSO user ID BAOMES1.

7. Press ENTER; the Alert Action(s) I - MSG panel is displayed (Figure 54).

```
BMC Software ------------- Alert Action(s) I - TIME ------------- AutoOPERATOR
COMMAND ===>                                                 TGT  --- AO62
       Rule-set === RULMAGIE             Rule-id  === NEWRULE3
Function ===>                                  (ADD, DELETE, DELETEQ)
Key      ===>
Text     ===>


Queues   ===>                                  Alert Queue Name(s)
Priority ===>                                  (CRIT,MAJ,MIN,WARN,INFO,CLEAR)
Color    ===>                                  RED,PINK,YEL,DKBL,LTBL,GRE,WHI
PCMD     ===>

System   ===>                                  Return to target after PCMD
Exec     ===>                                  Follow-up EXEC
Help     ===>                                  Associated HELP Panel
Targets  ===>                                  Target System
Udata    ===>                                  User Data
Origin   ===>                                  Origin
User     ===>                                  Userid
Alarm    ===>                                  Sound Alarm (YES/NO)
Publish  ===>                                  Messages (ADD/REPLACE/NO)

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 54. Creating a TIME-Initiated Rule (1): Alert Action(s) I Panel

8. This Rule will not issue an ALERT; press Enter and the Alert Action(s) II - MSG panel is displayed (Figure 55).

```
BMC Software ------------ Alert Action(s) II - TIME ------------- AutoOPERATOR
COMMAND ===>                                                 TGT  --- AO62

       Rule-set === RULMAGIE    Alert    Rule-id  === NEWRULE3

Retain                ===>                                      Yes/No

Escalate Direction    ===>                                      Up/Down
         Interval     ===>
                      ===>
                      ===>
                      ===>
                      ===>
                      ===>
         Disposition  ===>                                      Keep/Delete

         Exec    ===>

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 55. Creating a TIME-Initiated Rule (1): Alert Action(s) II Panel

9. Press ENTER; the Rule Processor Detail Control panel is re-displayed (Figure 56).

```
BMC Software ------------ Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
                                                              TIME --- 16:09:31
  The following options are displayed in sequence, or may    DATE --- 01/02/23
  be selected by entering the two-character code

   S1 - Selection Criteria            A1 - Action Specification
   SV - Variable Dependencies         AA - Alert Actions(s) I
                                      AD - Alert Actions(s) II
Rule ID       ===> NEWRULE3
Event Type    ===> TIME     Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status   ===>          (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===>          Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===>
Last Modified by          on          at

  Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

Figure 56. Creating a TIME-Initiated Rule (1): Rule Processor Detail Control Panel

10. Press END to apply changes.

    The Rule Set Overview panel is displayed.

11. Repeat Step 13 on page 113 and Step 14 on page 114 to complete saving the Rule NEWRULE3 to Rule Set RULMAGIE.

Important
If the subsystem on which a TIME-initiated Rule is fired comes up after the specified Start Time and before the specified Stop Time (or stop time implied by the Stop Count value), the firing of the Rules will behave as though the Rule fired on the Start Time and every interval thereafter.

Example 2: Creating a TIME-Initiated Rule with ATSTART

This example shows how to create a TIME-initiated Rule that begins to fire at BBI-SS PAS startup and fires every 15 minutes for a total of 10 times. After the Rule fires 10 times, the Rule will stop firing and it won't begin firing until the next BBI-SS PAS startup. The action the Rule takes is to display initiators.

To create this Rule:

1. Enter the new Rule ID and the event type on the Rule Processor Detail Control panel.

```
BMC Software ------------ Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
                                                              TIME --- 15:07:07
  The following options are displayed in sequence, or may    DATE --- 01/02/21
  be selected by entering the two-character code

   S1 - Selection Criteria            A1 - Action Specification
   SV - Variable Dependencies         AA - Alert Actions(s) I
                                      AD - Alert Actions(s) II
Rule ID       ===> NEWRULE4
Event Type    ===> TIME     Type  of event ( ? for list)
Initial Mode  ===> ENABLE   (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status   ===>          (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===>          Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===> DISPLAY INITS AT ATSTART
Last Modified by BAOMXY2  on 01/02/21 at 14:15

  Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

Figure 57. Creating a TIME-Initiated Rule (2): Rule Processor Detail Control Panel (Example 1)

2. Press ENTER; the Selection Criteria - TIME panel is displayed (Figure 58).

   Specify the start time for the Rule to fire and the time interval.

```
BMC Software ------------- Selection Criteria - TIME ------------ AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
                                                              TIME --- 15:07:20
       Rule-set === RULMAGIE             Rule-id  === NEWRULE4

Start Specification:
Start Time           ===> ATSTART           hh:mm:ss or ATSTART

Frequency:
Interval             ===> 00:15:00          hh:mm:ss

Stop Specification:
Stop time            ===>                   hh:mm:ss
          or
Stop Count           ===> 10



Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 58. Creating a TIME-Initiated Rule (2): Selection Criteria Panel

3. Enter the time specifications as follows:
   - To specify that the Rule begins firing at BBI-SS PAS startup, specify `ATSTART` for the `Start Time` field.
   - To specify that the Rule fires every 15 minutes, enter `00:15:00` for the `Interval` field.
   - To specify the Rule should fire a total of 10 times, specify `10` for the `Stop Count` field.

   The Rule will stop firing after 10 times and will not fire again until the next BBI-SS PAS startup.

4. Press ENTER; the Variable Dependencies - TIME panel is displayed (Figure 59).

```
BMC Software ------------ Variable Dependencies - TIME ---------- AutoOPERATOR
COMMAND ===>                                                TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === NEWRULE4

                                                                        OR/
 Variable-name                    Op  Variable-Value                   AND

 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 59. Creating a TIME-Initiated Rule (2): Variable Dependencies Panel

5. This example does not make use of variables; press ENTER.

The Action Specification - TIME panel is displayed (Figure 60). Specify the actions the Rule will take when it is fired.

```
BMC Software ------------- Action Specification - TIME ---------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === NEWRULE4

Automation Actions:
Journal          ===>
EXEC Name/Parms  ===>
Send (TSO IDs)   ===> BAOMES1
Cmd (Type  MVS ) ===> $DI

Set Variable     ===>                                    ===>


Notify           ===>                                   Outboard Pager ID
Info             ===>

DOM Id           ===>                                   Delete Operator Message
Issue WTO Msg    ===>



Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 60. Creating a TIME-Initiated Rule (2): Action Specification Panel

6. To specify that a notification is sent to a TSO user ID, enter the user ID name in the `Send (TSO IDs)` field.

   To specify the command the Rule issues when it is fired, enter the command in the `CMD (Type MVS)` field. In this case the MVS command to be issued is:

   `$DI`

   which will cause the initiators to be displayed in the SYSLOG.

   This command will be issued every 15 minutes and a notification will be sent to TSO user ID BAOMES1.

7. Press ENTER; the Alert Action(s) I - MSG panel is displayed (Figure 61).

```
BMC Software ------------- Alert Action(s) I - TIME ------------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
       Rule-set === RULMAGIE             Rule-id  === NEWRULE4
Function ===>                                  (ADD, DELETE, DELETEQ)
Key      ===>
Text     ===>


Queues   ===>                                  Alert Queue Name(s)
Priority ===>                                  (CRIT,MAJ,MIN,WARN,INFO,CLEAR)
Color    ===>                                  RED,PINK,YEL,DKBL,LTBL,GRE,WHI
PCMD     ===>

System   ===>                                  Return to target after PCMD
Exec     ===>                                  Follow-up EXEC
Help     ===>                                  Associated HELP Panel
Targets  ===>                                  Target System
Udata    ===>                                  User Data
Origin   ===>                                  Origin
User     ===>                                  Userid
Alarm    ===>                                  Sound Alarm (YES/NO)
Publish  ===>                                  Messages (ADD/REPLACE/NO)

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 61. Creating a TIME-Initiated Rule (2): Alert Action(s) I Panel

8. This Rule will not issue an ALERT; press Enter and the Alert Action(s) II - MSG panel is displayed (Figure 62).

```
BMC Software ------------ Alert Action(s) II - TIME ------------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62

       Rule-set === RULMAGIE    Alert    Rule-id  === NEWRULE4

Retain                ===>                                      Yes/No

Escalate Direction    ===>                                      Up/Down
         Interval     ===>
                      ===>
                      ===>
                      ===>
                      ===>
                      ===>
         Disposition  ===>                                      Keep/Delete

         Exec    ===>

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 62. Creating a TIME-Initiated Rule (2): Alert Action(s) II Panel

9. Press ENTER; the Rule Processor Detail Control panel is re-displayed (Figure 63).

```
BMC Software ------------ Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
                                                              TIME --- 16:09:31
  The following options are displayed in sequence, or may    DATE --- 01/02/23
  be selected by entering the two-character code

   S1 - Selection Criteria           A1 - Action Specification
   SV - Variable Dependencies        AA - Alert Actions(s) I
                                     AD - Alert Actions(s) II
Rule ID       ===> NEWRULE4
Event Type    ===> TIME     Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status   ===>          (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===>          Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===>
Last Modified by          on          at

  Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

Figure 63. Creating a TIME-Initiated Rule (2): Rule Processor Detail Control Panel (Example 2)

10. Press END to apply changes.

    The Rule Set Overview panel is displayed.

11. Repeat Step 13 on page 113 and Step 14 on page 114 to complete saving the Rule NEWRULE4 to Rule Set RULMAGIE.

Creating More Complex Rules

The following examples show how to create Rules using other features of the Rule Processor, such as the criteria match rate fields and resolving compound variables, as part of the primary selection criteria.

The two Rules in this section should be used together. The first Rule in Example 1 (NEWRULE5) shows how to set a variable as a result of an incoming event and then increases the value of the variable by 1.

The second Rule (NEWRULE6) shows how a Rule can fire when the variable value (set by NEWRULE5) has reached or increased past the set limit. First, Example 1 shows how to set a variable whose value increases with each occurrence of the incoming event.

Example 1: Setting a Variable with a Rule

This example shows how to set a variable whose value increases by 1 with each occurrence of the incoming event.

To create this Rule:

1. Enter the new Rule ID and the event type on the Rule Processor Detail Control panel.

```
BMC Software ------------ Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                 TGT  --- AO62
                                                             TIME --- 15:45:12
  The following options are displayed in sequence, or may    DATE --- 01/02/21
  be selected by entering the two-character code

   S1 - Selection Criteria            A1 - Action Specification
   SV - Variable Dependencies         AA - Alert Actions(s) I
                                      AD - Alert Actions(s) II
Rule ID       ===> NEWRULE5
Event Type    ===> MSG      Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status   ===>          (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===> MVS      Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===> SET VARIABLE, CREATE ALERT
Last Modified by          on          at

  Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

Figure 64. Setting a Variable with a Rule: Rule Processor Detail Control Panel (Example 1)

2. Press ENTER; the Selection Criteria - MSG panel is displayed (Figure 65).

   Specify the criteria the Rule must match with the event to cause the Rule to fire.

```
BMC Software ------------- Selection Criteria - MSG  ------------ AutoOPERATOR
COMMAND ===>                                                 TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === NEWRULE5

Text Description:
Text ID     ===> IEF403I                           Ignore leading plus: Y
Text String (Enter Below):


Issuer Identification:
Job name    ===>                                   Name of issuer
Type        ===>                                   (JOB, STC, or TSO)
Jobclass    ===>                                   Job class of issuer
Acct Info   ===>                                   Job accounting information
RACF User   ===>                                   RACF Userid
RACF Group  ===>                                   RACF Group name
Route Codes ===>
Desc Codes  ===>
Console ID  ===>                                   Destination Console
Console Name ==>                                   Destination Console Name

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 65. Setting a Variable with a Rule: Selection Criteria Panel

3. Enter IEF403I (the text-ID) of the message in the `Text ID` field.

   Press ENTER; the Variable Dependencies - MSG panel is displayed (Figure 66).

```
BMC Software ------------ Variable Dependencies - MSG  ---------- AutoOPERATOR
COMMAND ===>                                                 TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === NEWRULE5

                                                                        OR/
 Variable-name                    Op  Variable-Value                   AND

 ________________________________ __  ______________________________  ___
 ________________________________ __  ______________________________  ___
 ________________________________ __  ______________________________  ___
 ________________________________ __  ______________________________  ___
 ________________________________ __  ______________________________  ___
 ________________________________ __  ______________________________  ___
 ________________________________ __  ______________________________  ___
 ________________________________ __  ______________________________  ___
 ________________________________ __  ______________________________  ___
 ________________________________ __  ______________________________  ___
 ________________________________ __  ______________________________  ___
 ________________________________ __  ______________________________  ___

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 66. Setting a Variable with a Rule: Variable Dependencies Panel

4. This example does not make use of this panel; press ENTER.

The Action Specification - MSG panel is displayed (Figure 67).

Specify the actions the Rule will take when it is fired.

```
BMC Software ------------- Action Specification - MSG  ---------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
       Rule-set === RULMAGIE             Rule-id  === NEWRULE5
Display at dest. ===> YES       Journal ===>           SYSLOG Display ===>
EXEC Name/Parms  ===>
Send (TSO IDs)   ===>
Cmd (Type  MVS ) ===>

Set Variable     ===> a.&imfjtype..&imfjnum              ===> +1
Reword Msg       ===>

Notify           ===>                                    Outboard Pager ID
Info             ===>

Console ID       ===>                                    Destination Console ID
Console Name     ===>                                    Dest. Console ID Name
Route codes      ===>
Descriptor codes ===>
Update Rout/Desc Codes ===>                              (ADD/REPL)
DOM Id           ===>                                    Delete Operator Message
Issue WTO Msg    ===>

DOM Message      ===>                                    (Yes/No)
Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 67. Setting a Variable with a Rule: Action Specification Panel

5. To specify the variable and the actions for this Rule:

   – To set a variable, enter the variable name in the `Set Variable` field.

     The variable set in this example is `a.&imfjtype..&imfjnum`. For more information about how this variable actually resolves, refer to “Creating More Powerful Rules” on page 155.

   – To increase the value of this variable by 1 every time the Rule Processor sees this event, enter +1 for the second field in the `Set Variable` field. What this does is increase the variable value by one every time the message is seen by the Rule Processor.

6. Press ENTER; the Alert Action(s) I - MSG panel is displayed (Figure 68).

```
BMC Software ------------- Alert Action(s) I - MSG ------------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
       Rule-set === RULMAGIE             Rule-id  === NEWRULE5
Function ===>                                  (ADD, DELETE, DELETEQ)
Key      ===>
Text     ===>


Queues   ===>                                  Alert Queue Name(s)
Priority ===>                                  (CRIT,MAJ,MIN,WARN,INFO,CLEAR)
Color    ===>                                  RED,PINK,YEL,DKBL,LTBL,GRE,WHI
PCMD     ===>

System   ===>                                  Return to target after PCMD
Exec     ===>                                  Follow-up EXEC
Help     ===>                                  Associated HELP Panel
Targets  ===>                                  Target System
Udata    ===>                                  User Data
Origin   ===>                                  Origin
User     ===>                                  Userid
Alarm    ===>                                  Sound Alarm (YES/NO)
Publish  ===>                                  Messages (ADD/REPLACE/NO)

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 68. Setting a Variable with a Rule: Alert Action(s) I Panel

7. This Rule will not issue an ALERT; press Enter and the Alert Action(s) II - MSG panel is displayed (Figure 69).

```
BMC Software ----------------- Alert Action(s) II - MSG -------------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62

       Rule-set === RULMAGIE    Alert    Rule-id  === NEWRULE5

Retain                ===>                                      Yes/No

Escalate Direction    ===>                                      Up/Down
         Interval     ===>
                      ===>
                      ===>
                      ===>
                      ===>
                      ===>
         Disposition  ===>                                      Keep/Delete

         Exec    ===>

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 69. Setting a Variable with a Rule: Alert Action(s) II Panel

8. Press ENTER; the Rule Processor Detail Control panel is re-displayed (Figure 70).

```
BMC Software ------------- Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
                                                              TIME --- 16:09:31
  The following options are displayed in sequence, or may     DATE --- 01/02/23
  be selected by entering the two-character code

   S1 - Selection Criteria            A1 - Action Specification
   SV - Variable Dependencies         AA - Alert Actions(s) I
                                      AD - Alert Actions(s) II
Rule ID       ===> NEWRULE5
Event Type    ===> TIME     Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status   ===>          (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===>          Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===>
Last Modified by          on          at

  Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

Figure 70. Setting a Variable with a Rule: Rule Processor Detail Control Panel (Example 2)

9. Press END to apply changes.

   The Rule Set Overview panel is displayed.

10. Repeat Step 13 on page 113 and Step 14 on page 114 to complete saving the Rule NEWRULE5 to Rule Set RULMAGIE.

This Rule should be used in conjunction with the Rule NEWRULE6 described in Example 2. The Rule NEWRULE6 uses the variable value set in NEWRULE5 to determine whether it should fire.

Example 2: Creating a Rule That Fires When a Variable Value Is Met

This example shows how to create a Rule that will fire when the variable set in Example 1 (in NEWRULE5) reaches a specific value.

To create this Rule:

1. Enter the new Rule ID and the event type on the Rule Processor Detail Control panel.

```
BMC Software ------------- Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                 TGT  --- AO62
                                                             TIME --- 15:45:12
  The following options are displayed in sequence, or may    DATE --- 01/02/21
  be selected by entering the two-character code

   S1 - Selection Criteria           A1 - Action Specification
   SV - Variable Dependencies        AA - Alert Actions(s) I
                                     AD - Alert Actions(s) II
Rule ID       ===> NEWRULE6
Event Type    ===> MSG      Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===> 10       (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===> 60       (Interval length, 1-99999 seconds)
then status   ===> SUSPEND  (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===> MVS      Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===> CREATE ALERT
Last Modified by          on          at

  Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

Figure 71. Triggering a Rule with Variables: Rule Processor Detail Control Panel (Example 1)

You should use the Criteria match rate fields to prevent the Rule from over-firing. Occasionally, the Rule Processor will be flooded with the same event due to an abnormal system situation, and you do not want the Rule to fire for every event.

To prevent the Rule in this example from over-firing, fill in the Criteria match rate fields as follows.

2. To specify that the Rule will be suspended when it is matched 10 times in one minute:
   - Enter 10 for the number of times the Rule might match incoming events
   - Enter 60 (in seconds)
   - Enter SUSPEND and the Rule will be suspended if the Rule matches an event 10 times or more within one minute

3. Press ENTER; the Selection Criteria - MSG panel is displayed (Figure 72).

   Specify the criteria the Rule must match with the event to cause the Rule to fire.

```
BMC Software -------------- Selection Criteria - MSG  ------------ AutoOPERATOR
COMMAND ===>                                                 TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === NEWRULE6

Text Description:
Text ID     ===> IEF403I                          Ignore leading plus: Y
Text String (Enter Below):


Issuer Identification:
Job name    ===>                                   Name of issuer
Type        ===>                                   (JOB, STC, or TSO)
Jobclass    ===>                                   Job class of issuer
Acct Info   ===>                                   Job accounting information
RACF User   ===>                                   RACF Userid
RACF Group  ===>                                   RACF Group name
Route Codes ===>
Desc Codes  ===>
Console ID  ===>                                   Destination Console
Console Name ==>                                   Destination Console Name

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 72. Triggering a Rule with Variables: Selection Criteria Panel

4. Enter IEF030I (the text-ID) of the message in the `Text ID` field.

   Press ENTER; the Variable Dependencies - MSG panel is displayed (Figure 73).

```
BMC Software ------------- Variable Dependencies - MSG  ---------- AutoOPERATOR
COMMAND ===>                                                 TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === NEWRULE6

                                                                        OR/
 Variable-name                    Op  Variable-Value                   AND

 &!a.imfjtype.imfjnum____________ gt  12_________________________________  ___

 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 73. Triggering a Rule with Variables: Variable Dependencies Panel

5. Fill in the panel as shown above to specify that the Rule will fire only when the IEF403I message has been seen by the Rule Processor more than 12 times.

The following describes how the specifications on this panel will cause the Rule to work:

– The variable `&!a.imfjtype.imfjnum` **is a stem variable** that was set in Example 1 whose value increases by one every time the Rule Processor sees the IEF403I event.

– The Rule will fire when the value of the variable is greater than 12.

  There is no other selection criteria for this Rule; therefore, the Rule will fire only when this condition is true.

6. Press ENTER; the Action Specifications - MSG panel is displayed (Figure 74).

   Specify the actions the Rule will take when it is fired.

```
BMC Software -------------- Action Specification - MSG  ---------- AutoOPERATOR
COMMAND ===>                                                 TGT  --- AO62
       Rule-set === RULMAGIE             Rule-id  === NEWRULE6
Display at dest. ===> NO        Journal ===> YES       SYSLOG Display ===>
EXEC Name/Parms  ===>
Send (TSO IDs)   ===>
Cmd (Type  MVS ) ===>

Set Variable     ===>                                    ===>
Reword Msg       ===>

Notify           ===>                                   Outboard Pager ID
Info             ===>

Console ID       ===>                                   Destination Console ID
Console Name     ===>                                   Dest. Console ID Name
Route codes      ===>
Descriptor codes ===>
Update Rout/Desc Codes ===>                             (ADD/REPL)
DOM Id           ===>                                   Delete Operator Message
Issue WTO Msg    ===>

DOM Message      ===>                                   (Yes/No)
Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 74. Triggering a Rule with Variables: Action Specification Panel

7. To suppress the message from the console and write it to the BBI-SS PAS Journal, specify NO for `Display at dest.` and YES for `Journal`.

8. Press ENTER; the Alert Actions I - MSG panel is displayed (Figure 75).

```
BMC Software -------------- Alert Action(s) I - MSG  ------------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
       Rule-set === RULMAGIE            Rule-id  === NEWRULE6
Function ===> add                              (ADD, DELETE, DELETEQ)
Key      ===> &imfday&imfrlfrd
Text     ===> job &word2 started more than 12 times


Queues   ===>                                  Alert Queue Name(s)
Priority ===>                                  (CRIT,MAJ,MIN,WARN,INFO,CLEAR)
Color    ===>                                  RED,PINK,YEL,DKBL,LTBL,GRE,WHI
PCMD     ===>

System   ===>                                  Return to target after PCMD
Exec     ===>                                  Follow-up EXEC
Help     ===>                                  Associated HELP Panel
Targets  ===>                                  Target System
Udata    ===>                                  User Data
Origin   ===>                                  Origin
User     ===>                                  Userid
Alarm    ===>                                  Sound Alarm (YES/NO)
Publish  ===>                                  Messages (ADD/REPLACE/NO)

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 75. Triggering a Rule with Variables: Alert Action(s) I Panel

9. To specify an ALERT will be issued as one of the Rule's actions, you must:

   – Specify ADD for the `Function` field to issue an ALERT.

   – Specify a unique ALERT key in the queue associated with it. ALERTs with the same key override existing ones.

     To create a unique key, this example uses the IMFDAY and IMFRLFRD variables. These variables will resolve to the day of the week and the number of times the Rule was fired. This ensures the key will be unique.

   – Enter ALERT text that states that the job (which issued the message that triggers this Rule) has started more than 10 times.

     This text warns the operator that the job may be having difficulty starting and may require operator attention.

10. Press ENTER; the Alert Actions II - MSG panel is displayed (Figure 76).

```
BMC Software ------------- Alert Action(s) II - MSG  ------------- AutoOPERATOR
COMMAND ===>                                                TGT  --- AO62

       Rule-set === RULMAGIE    Alert    Rule-id  === NEWRULE6

Auto Delete           ===>                                      Yes/No
Retain                ===>                                      Yes/No

Escalate Direction    ===>                                      Up/Down
         Interval     ===>
                      ===>
                      ===>
                      ===>
                      ===>
                      ===>
         Disposition  ===>                                      Keep/Delete

         Exec    ===>

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 76. Triggering a Rule with Variables: Alert Action(s) II Panel

The ALERT will not use any of the features available on this panel.

11. Press ENTER; the Rule Processor Detail Control panel is re-displayed (Figure 77).

```
BMC Software ------------- Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                TGT  --- AO62
                                                            TIME --- 16:09:31
  The following options are displayed in sequence, or may    DATE --- 01/02/23
  be selected by entering the two-character code

   S1 - Selection Criteria            A1 - Action Specification
   SV - Variable Dependencies         AA - Alert Actions(s) I
                                      AD - Alert Actions(s) II
Rule ID       ===> NEWRULE6
Event Type    ===> TIME     Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status   ===>        (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===>          Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===>
Last Modified by          on          at

  Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

Figure 77. Triggering a Rule with Variables: Rule Processor Detail Control Panel (Example 2)

12. Press END to apply changes.

    The Rule Set Overview panel is displayed.

13. Repeat Step 13 on page 113 and Step 14 on page 114 to complete saving the Rule NEWRULE6 to Rule Set RULMAGIE.

Using the Rules

To properly use the two Rules created in these examples, NEWRULE5 should come before NEWRULE6 in the Rule Set. In other words, NEWRULE6 should follow immediately after NEWRULE5.

Set the automation strategy of the Rule Set to FIRST. This means that events are evaluated through the Rule Set and only the first Rule that matches with an event will be fired. Once the Rule fires, the event is handled and subsequent Rules are not compared against the event and, therefore, will not fire.

In this example, each time IEF403I message matches NEWRULE5, the value of variable `&!a.imfjtype.&imfjnum` will increase by 1. When the value reaches 13, the NEWRULE6 will fire because the value of the `!a.&imfjtype.&imfrlfrd` has become greater than 12 and an ALERT will be issued.

The ALERT will call attention to the fact that a job has started more than 12 times, which might indicate that the job needs operator attention.

Creating ALERTs from Rules

The following sections describe how to create an ALERT from a Rule. The first example shows how to modify an existing ALERT to be retained across BBI-SS PAS restarts and MVS IPLs.

The second example shows how to create an ALERT whose priority can increase over user-specified periods of time.

Example 1: Creating an ALERT Retained across BBI-SS PAS Restarts and MVS IPLs

To issue an ALERT from a Rule that will be written to disk and saved across BBI-SS PAS restarts and MVS IPLs:

1. Enter the new Rule ID and the event type on the Rule Processor Detail Control panel.

```
BMC Software ------------- Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                 TGT  --- AO62
                                                             TIME --- 08:08:08
  The following options are displayed in sequence, or may    DATE --- 01/02/22
  be selected by entering the two-character code

   S1 - Selection Criteria            A1 - Action Specification
   SV - Variable Dependencies         AA - Alert Actions(s) I
                                      AD - Alert Actions(s) II
Rule ID       ===> NEWRULE7
Event Type    ===> ALRT     Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status   ===>         (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===> payroll  Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===> Modify Payroll ALERT
Last Modified by          on          at

  Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

Figure 78. Modifying an ALERT to Survive MVS IPLs: Rule Processor Detail Control Panel (Example 1)

2. Press ENTER; the Selection Criteria - ALRT panel is displayed (Figure 79).

   Specify the criteria the Rule must match with the event to cause the Rule to fire.

```
BMC Software -------------- Selection Criteria - ALRT ------------ AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === NEWRULE7

Text Description:
Text ID     ===> *                              First Word of ALert
Text String (Enter Below):
*payroll*failed*

Key      ===>
Queue    ===>                                   Alert Queue









Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 79. Modifying an ALERT to Survive MVS IPLs: Selection Criteria Panel

By specifying the text in the `Text String` field as shown above, the Rule will match any event that contains the text

`*payroll*failed*`

The asterisks represent any amount of text surrounding the words `payroll` and `failed`.

3. Press ENTER; the Variable Dependencies - ALRT panel is displayed (Figure 80).

```
BMC Software ------------- Variable Dependencies - ALRT ---------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === NEWRULE7

                                                                        OR/
 Variable-name                    Op  Variable-Value                   AND

 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________  ___
 ________________________________ __  ____________________________________

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 80. Modifying an ALERT to Survive MVS IPLs: Variable Dependencies Panel

4. This example does not make use of variables; press ENTER.

   The Action Specification - ALRT panel is displayed (Figure 81).

   Specify the actions the Rule will take when it is fired.

```
BMC Software -------------- Action Specification - ALRT ---------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === NEWRULE7

Automation Actions:
Display at dest. ===> YES       Journal ===>
EXEC Name/Parms  ===>
Send (TSO IDs)   ===>
Cmd (Type  MVS ) ===>

Set Variable     ===>                                      ===>
Reword Alert     ===>

Notify           ===>                                    Outboard Pager ID
Info             ===>

DOM Id           ===>                                    Delete Operator Message
Issue WTO Msg    ===>



Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 81. Modifying an ALERT to Survive MVS IPLs: Action Specification Panel

5. Specify YES for the `Display at dest.` field to cause the message to be written to the console.

6. Press ENTER; the Alert Action(s) I - ALRT panel is displayed (Figure 82).

```
BMC Software -------------- Alert Action(s) I - ALRT ------------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
       Rule-set === RULMAGIE             Rule-id  === NEWRULE7
Function ===> modify                            (ADD, DELETE, DELETEQ, MODIFY)
Key      ===>
Text     ===>


Queues   ===>                                   Alert Queue Name(s)
Priority ===>                                   (CRIT,MAJ,MIN,WARN,INFO,CLEAR)
Color    ===>                                   RED,PINK,YEL,DKBL,LTBL,GRE,WHI
PCMD     ===>

System   ===>                                   Return to target after PCMD
Exec     ===>                                   Follow-up EXEC
Help     ===>                                   Associated HELP Panel
Targets  ===>                                   Target System
Udata    ===>                                   User Data
Origin   ===>                                   Origin
User     ===>                                   Userid
Alarm    ===>                                   Sound Alarm (YES/NO)
Publish  ===>                                   Messages (ADD/REPLACE/NO)

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 82. Modifying an ALERT to Survive MVS IPLs: Alert Action(s) I Panel

7. Specify MODIFY for the `Function` field.

The MODIFY function allows you to modify an ALRT event. In this example, the Rule will modify the original ALRT event to be retained and saved to disk. An ALERT which is retained will survive a BBI-SS PAS restart and MVS IPLs. When the system is restarted, the ALERT will still be recovered.

8. Press ENTER; the Alert Action(s) II - ALRT panel is displayed (Figure 83).

```
BMC Software ------------- Alert Action(s) II - ALRT ------------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62

       Rule-set === RULMAGIE    Alert    Rule-id  === NEWRULE7

Retain                ===> Yes                                    Yes/No

Escalate Direction    ===>                                        Up/Down
         Interval     ===>
                      ===>
                      ===>
                      ===>
                      ===>
                      ===>
         Disposition  ===>                                        Keep/Delete

         Exec    ===>

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 83. Modifying an ALERT to Survive MVS IPLs: Alert Action(s) II Panel

9. To specify that the ALERT will be saved to survive BBI-SS PAS restarts and MVS IPLs, enter YES in the `Retain` field.

10. Press ENTER; the Rule Processor Detail Control panel is displayed (Figure 84).

```
BMC Software ------------- Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
                                                              TIME --- 16:09:31
  The following options are displayed in sequence, or may    DATE --- 01/02/23
  be selected by entering the two-character code

   S1 - Selection Criteria            A1 - Action Specification
   SV - Variable Dependencies         AA - Alert Actions(s) I
                                      AD - Alert Actions(s) II
Rule ID       ===> NEWRULE6
Event Type    ===> TIME     Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status   ===>         (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===>          Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===>
Last Modified by          on          at

  Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

Figure 84. Modifying an ALERT to Survive MVS IPLs: Rule Processor Detail Control Panel (Example 2)

11. Press END to apply changes.

    The Rule Set Overview panel is displayed.

12. Repeat Step 13 on page 113 and Step 14 on page 114 to complete saving the Rule NEWRULE7 to Rule Set RULMAGIE.

Example 2: Creating an ALERT Whose Priority Increases Over Time

You can create an ALERT from a Rule whose priority will increase or decrease over user-specified periods of time. When the ALERT reaches its final priority level, you can specify that the ALERT is kept or discarded. You also can specify that an EXEC is issued.

To issue an ALERT whose priority will increase over time from a Rule:

1. Enter the new Rule ID and the event type on the Rule Processor Detail Control panel.

```
BMC Software ------------- Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                TGT  --- AO62
                                                            TIME --- 08:14:14
  The following options are displayed in sequence, or may   DATE --- 01/02/22
  be selected by entering the two-character code

   S1 - Selection Criteria            A1 - Action Specification
   SV - Variable Dependencies         AA - Alert Actions(s) I
                                      AD - Alert Actions(s) II
Rule ID       ===> NEWRULE8
Event Type    ===> MSG      Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status   ===>        (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===> MVS      Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===> Create ALERT for Enqueue
Last Modified by          on          at

  Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

Figure 85. Escalating an ALERT: Rule Processor Detail Control Panel (Example 1)

2. Press ENTER; the Selection Criteria - MSG panel is displayed (Figure 86).

   Specify the criteria the Rule must match with the event to cause the Rule to fire.

```
BMC Software -------------- Selection Criteria - MSG  ------------ AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === NEWRULE8

Text Description:
Text ID     ===> IEF863I                          Ignore leading plus: Y
Text String (Enter Below):


Issuer Identification:
Job name    ===>                                  Name of issuer
Type        ===>                                  (JOB, STC, or TSO)
Jobclass    ===>                                  Job class of issuer
Acct Info   ===>                                  Job accounting information
RACF User   ===>                                  RACF Userid
RACF Group  ===>                                  RACF Group name
Route Codes ===>
Desc Codes  ===>
Console ID  ===>                                  Destination Console
Console Name ==>                                  Destination Console Name

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 86. Escalating an ALERT: Selection Criteria Panel

3. Enter IEF863I in the `Text-ID` field.

4. Press ENTER; the Variable Dependencies - MSG panel is displayed (Figure 87).

```
BMC Software ------------- Variable Dependencies - MSG  ---------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === NEWRULE8

                                                                          OR/
 Variable-name                    Op  Variable-Value                     AND

 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________  ___
 ________________________________ __  ________________________________

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 87. Escalating an ALERT: Variable Dependencies Panel

5. This example does not make use of variables; press ENTER.

The Action Specification - MSG panel is displayed (Figure 88). Specify the actions the Rule will take when it is fired.

```
BMC Software -------------- Action Specification - MSG  ---------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
       Rule-set === RULMAGIE             Rule-id  === NEWRULE8
Display at dest. ===> NO        Journal ===>           SYSLOG Display ===>
EXEC Name/Parms  ===>
Send (TSO IDs)   ===>
Cmd (Type  MVS ) ===>

Set Variable     ===>                                    ===>
Reword Msg       ===>

Notify           ===>                                   Outboard Pager ID
Info             ===>

Console ID       ===>                                   Destination Console ID
Console Name     ===>                                   Dest. Console ID Name
Route codes      ===>
Descriptor codes ===>
Update Rout/Desc Codes ===>                             (ADD/REPL)
DOM Id           ===>                                   Delete Operator Message
Issue WTO Msg    ===>

DOM Message      ===>                                   (Yes/No)
Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 88. Escalating an ALERT: Action Specification Panel

6. Specify NO for `Display at dest.`, and the IEF863I message will be suppressed from the console

7. Press ENTER; the Alert Action(s) I - MSG panel is displayed (Figure 89).

```
BMC Software -------------- Alert Action(s) I - MSG  ------------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
       Rule-set === RULMAGIE             Rule-id  === NEWRULE8
Function ===> ADD                                (ADD, DELETE, DELETEQ)
Key      ===> &IMFDAY&WORD5&IMFRLFRD
Text     ===> &IMFTEXT


Queues   ===> MAIN                               Alert Queue Name(s)
Priority ===> INFORMATIONAL                      (CRIT,MAJ,MIN,WARN,INFO,CLEAR)
Color    ===>                                    RED,PINK,YEL,DKBL,LTBL,GRE,WHI
PCMD     ===>

System   ===>                                    Return to target after PCMD
Exec     ===>                                    Follow-up EXEC
Help     ===>                                    Associated HELP Panel
Targets  ===>                                    Target System
Udata    ===>                                    User Data
Origin   ===>                                    Origin
User     ===>                                    Userid
Alarm    ===>                                    Sound Alarm (YES/NO)
Publish  ===>                                    Messages (ADD/REPLACE/NO)

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 89. Escalating an ALERT: Alert Action(s) I Panel

8. To issue an ALERT from a Rule:
   - Use the ADD function for the `Function` field.
   - Every ALERT must have a unique ALERT key associated with it. ALERTs with the same key override each other.

     To create a unique key, this example uses the IMFDAY, WORD5, and IMFRLFRD variables. These variables will resolve to the day of the week, the jobname, and the number of times the Rule was fired. This ensures the key will be unique.
   - The ALERT text will resolve the actual text of the IEF863I message.
   - The ALERT will belong to a queue named MAIN.
   - The starting priority will be INFORMATIONAL.

     The color of the ALERT will default to green, which is the default color associated with INFORMATIONAL-level ALERTs.
9. Press ENTER; the Alert Actions II - MSG panel is displayed (Figure 90).

```
BMC Software ------------- Alert Action(s) II - MSG  ------------- AutoOPERATOR
COMMAND ===>                                                 TGT  --- AO62

       Rule-set === RULMAGIE    Alert    Rule-id  === NEWRULE8

Auto Delete           ===>                                      Yes/No
Retain                ===>                                      Yes/No

Escalate Direction    ===> up                                   Up/Down
         Interval     ===> 5
                      ===> 5
                      ===> 5
                      ===> 5
                      ===>
                      ===>
         Disposition  ===> keep                                 Keep/Delete

         Exec    ===>

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 90. Escalating an ALERT: Alert Action(s) II Panel

10. To specify that the ALERT will increase in priority, specify UP for the `Escalate Direction` field.

    You must also specify the interval, in minutes, over which the ALERT will increase.

    For this example, the ALERT will begin at INFORMATIONAL (and remain there for 5 minutes); at the end of five minutes the ALERT will become a WARNING level ALERT and remain there for 5 minutes. The ALERT will then progress through MINOR, MAJOR, and finally, CRITICAL priorities.

    When the last time interval expires, you can specify that the ALERT is either kept in the queue or deleted in the `Disposition` field. This example shows the ALERT is kept.

11. Press ENTER; the Rule Processor Detail Control panel is displayed (Figure 91).

```
BMC Software ------------- Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
                                                              TIME --- 16:09:31
  The following options are displayed in sequence, or may    DATE --- 01/02/23
  be selected by entering the two-character code

   S1 - Selection Criteria           A1 - Action Specification
   SV - Variable Dependencies        AA - Alert Actions(s) I
                                     AD - Alert Actions(s) II
Rule ID       ===> NEWRULE8
Event Type    ===> TIME     Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status  ===>           (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===>          Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===>
Last Modified by          on          at

  Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

Figure 91. Escalating an ALERT: Rule Processor Detail Control Panel (Example 2)

12. Press END to apply changes.

    The Rule Set Overview panel is displayed.

13. Repeat Step 13 on page 113 and Step 14 on page 114 to complete saving the Rule NEWRULE7 to Rule Set RULMAGIE.

Creating a VAR-Initiated Rule

The following section describes how to create a Rule that is fired when a variable in the SHARED pool is updated. VAR-initiated Rules are fired based on a variable value you specify being updated, deleted, or created.

To create a Rule that is fired when a variable is updated:

1. Enter the new Rule ID and the event type on the Rule Processor Detail Control panel.

```
BMC Software ------------- Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62
                                                              TIME --- 08:34:47
  The following options are displayed in sequence, or may    DATE --- 01/02/22
  be selected by entering the two-character code

   S1 - Selection Criteria            A1 - Action Specification
   SV - Variable Dependencies         AA - Alert Actions(s) I
                                      AD - Alert Actions(s) II
Rule ID       ===> NEWRULE9
Event Type    ===> VAR      Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status   ===>          (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===> MVS      Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===> variable updated
Last Modified by          on          at

  Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

Figure 92. Creating a VAR-Initiated Rule: Rule Processor Detail Control Panel (Example 1)

2. Press ENTER; the Selection Criteria - VAR panel is displayed (Figure 93).

   Specify the variable name and when the Rule should fire (for example, when the variable is updated, deleted, or created).

```
BMC Software -------------- Selection Criteria - VAR  ------------ AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === NEWRULE9

Variable Description:  (Shared variable only)
Name        ===> myvar

Fire this Rule when:

    Variable is  ===> upd                                (Created,Deleted,Upd)



Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 93. Creating a VAR-Initiated Rule: Selection Criteria Panel

3. Enter the variable name in the `Name` field (in this example, the variable name is MYVAR).

4. Specify the action that will fire the Rule; valid values are:
   – Created - The Rule will fire when MYVAR is created in the SHARED pool.
   – Deleted - The Rule will fire when MYVAR is deleted from the SHARED pool.
   – Upd - The Rule will fire when the value of MYVAR is updated in the SHARED pool.

   In this example, the Rule will fire when the variable is updated.
5. Press ENTER; the Variable Dependencies - VAR panel is displayed (Figure 94).

```
BMC Software ------------- Variable Dependencies - VAR  ---------- AutoOPERATOR
COMMAND ===>                                                 TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === NEWRULE9

                                                                          OR/
 Variable-name                    Op  Variable-Value                     AND

 myvar_____________________________ eq  error_______________________________  ___
 ________________________________ __  ________________________________________  ___
 ________________________________ __  ________________________________________  ___
 ________________________________ __  ________________________________________  ___
 ________________________________ __  ________________________________________  ___
 ________________________________ __  ________________________________________  ___
 ________________________________ __  ________________________________________  ___
 ________________________________ __  ________________________________________  ___
 ________________________________ __  ________________________________________  ___
 ________________________________ __  ________________________________________  ___
 ________________________________ __  ________________________________________  ___
 ________________________________ __  ________________________________________  ___
 ________________________________ __  ________________________________________  ___

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 94. Creating a VAR-Initiated Rule: Variable Dependencies Panel

6. Specify the value of the variable that, when matched, will trigger the Rule. In this example, when the MYVAR variable contains the value `error`, the Rule will fire.
7. Press ENTER.

The Action Specification - MSG panel is displayed (Figure 95). Specify the actions the Rule will take when it is fired.

```
BMC Software -------------- Action Specification - VAR  ---------- AutoOPERATOR
COMMAND ===>                                                 TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === NEWRULE9

Automation Actions:
Journal          ===>
EXEC Name/Parms  ===>
Send (TSO IDs)   ===> baomxy2
Cmd (Type  MVS ) ===>

Set Variable     ===>                                    ===>


Notify           ===>                                  Outboard Pager ID
Info             ===>

DOM Id           ===>                                  Delete Operator Message
Issue WTO Msg    ===>



Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 95. Creating a VAR-Initiated Rule: Action Specification Panel

8. Specify the action the Rule will take when it fires. Here, the only action is that TSO user BAOMXY1 will be notified when the Rules fires.

9. Press ENTER; the Alert Action(s) I - VAR panel is displayed (Figure 96).

```
BMC Software -------------- Alert Action(s) I - VAR  ------------- AutoOPERATOR
COMMAND ===>                                                 TGT  --- AO62
       Rule-set === RULMAGIE             Rule-id  === NEWRULE9
Function ===>                                 (ADD, DELETE, DELETEQ)
Key      ===>
Text     ===>


Queues   ===>                                 Alert Queue Name(s)
Priority ===>                                 (CRIT,MAJ,MIN,WARN,INFO,CLEAR)
Color    ===>                                 RED,PINK,YEL,DKBL,LTBL,GRE,WHI
PCMD     ===>

System   ===>                                 Return to target after PCMD
Exec     ===>                                 Follow-up EXEC
Help     ===>                                 Associated HELP Panel
Targets  ===>                                 Target System
Udata    ===>                                 User Data
Origin   ===>                                 Origin
User     ===>                                 Userid
Alarm    ===>                                 Sound Alarm (YES/NO)
Publish  ===>                                 Messages (ADD/REPLACE/NO)

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 96. Creating a VAR-Initiated Rule: Alert Action(s) I Panel

10. This Rule will not issue an ALERT; press Enter and the Alert Action(s) II - VAR panel is displayed (Figure 97).

```
BMC Software ------------- Alert Action(s) II - VAR -------------- AutoOPERATOR
COMMAND ===>                                                TGT  --- AO62

       Rule-set === RULMAGIE    Alert    Rule-id  === NEWRULE9

Retain                ===>                                       Yes/No

Escalate Direction    ===>                                       Up/Down
         Interval     ===>
                      ===>
                      ===>
                      ===>
                      ===>
                      ===>
         Disposition  ===>                                       Keep/Delete

         Exec    ===>

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
```

Figure 97. Creating a VAR-Initiated Rule: Alert Action(s) II Panel

11. Press ENTER; the Rule Processor Detail Control panel is re-displayed (Figure 98).

```
BMC Software ------------- Rule Processor Detail Control --------- AutoOPERATOR
COMMAND ===>                                                TGT  --- AO62
                                                            TIME --- 16:09:31
  The following options are displayed in sequence, or may    DATE --- 01/02/23
  be selected by entering the two-character code

   S1 - Selection Criteria            A1 - Action Specification
   SV - Variable Dependencies         AA - Alert Actions(s) I
                                      AD - Alert Actions(s) II
Rule ID       ===> NEWRULE9
Event Type    ===> VAR      Type  of event ( ? for list)
Initial Mode  ===> ENABLED  (ENABLED/DISABLED/TEST)

Criteria match rate threshold:
If matched    ===>          (Maximum # times matched within INTERVAL, 0-100)
in seconds    ===>          (Interval length, 1-99999 seconds)
then status   ===>          (SUSPEND, DISABLE, NOACTION)

Application information:
Group         ===>          Function      ===>          Code   ===>
Author        ===> BAOMXY2  Description   ===>
Last Modified by          on          at

  Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

Figure 98. Creating a VAR-Initiated Rule: Rule Processor Detail Control Panel (Example 2)

12. Press END to apply changes.

    The Rule Set Overview panel is displayed.

13. Repeat Step 13 on page 113 and Step 14 on page 114 to complete saving the Rule NEWRULE5 to Rule Set RULMAGIE.

Chapter 8. Creating More Powerful Rules

This chapter describes how to use:

- Pattern matching
- Variables

to create more powerful Rules in the Rule Processor.

Introduction

When you specify data on any of the Selection Criteria or Action Specification panels (except where noted), you can specify:

- Constant values (or literals)
- Patterns of strings of data
- Variables
- Combinations of all of the above

Specifying only literals when you create a Rule might limit the Rule's applicability to only very specific situations. However, by making use of variables, combinations of literals and variables, or pattern matching, you can create more powerful and flexible Rules that apply to more general situations. Using these techniques should allow you to accomplish an automation task using less EXECs and more effective Rules.

The following two sections describe how to use variables and pattern matching in Rules:

- “Using Pattern Matching on Selection Criteria Panels” on page 156
- “Using Variables in the Rule Processor” on page 161

Using Pattern Matching on Selection Criteria Panels

On the selection criteria panels for all events, you can use pattern matching. The following sections describe how to do this.

You can use wildcard characters on most of the fields in the various Selection Criteria panels. The wildcard characters are:

- The + sign

  Stands for 1 position that can be a blank or any character.

- The * sign

  Stands for any number (including zero) of non-blank characters.

- The Not sign (¬)

  Stands for a negative logic test.

  For example, ¬ABCD matches with everything **except** ABCD.

- The single quote mark (‘)

  Acts as a toggle switch and thus cannot be coded in the Text String or Variable depencencies panel. The first single quote mark turns pattern matching off, the next turns it on again. Pattern matching must be used in place of single quote marks.

  For example, *IS A* or *TEST MESSAGE* matches THIS 'IS A' TEST MESSAGE.

If a + or * is embedded within the pattern string and the pattern string is specified within quotes, the + or * values are recognized as part of the string. For example, 'ABC+D' matches only with ABC+D.

Use these wildcard characters when you want to create a pattern for values in your selection criteria. For example, assuming all error messages begin with ERR, you might want to write a Rule that fires for all error messages.

Instead of creating a Rule for every error message, you can specify a pattern of characters that the Text-ID must match: for example `Text-ID===> ERR*`. This pattern specifies that the Rule should be fired every time a message that begins with ERR is issued.

Examples of Using Pattern Matching

This section describes how you might use pattern matching in the Text ID and Text String fields.

Example 1

Select a message; for example:

```
DFH3461I ABCD SNFL 8:38:41 NODE RS8014 SESSION STARTED
```

Suppose you want a Rule to fire only for the message DFH346I that contains ABCD anywhere in the text. Enter the selection criteria on the Rule Selection Criteria panel like this:

```
Text Description:
Text-ID      ==> DFH3461I                   First word of event
Text String  ==> (Enter Below):
*ABCD*
```

Example 2

In this example:

```
Text ID      ==> INIT+++                    First word of event
Text String  ==> (Enter Below):             Text string (incl. Text-ID)
*INACTIVE*MISSION*
```

any message whose first word:

- Is 7 characters long
- Starts with `INIT`
- Has three positions containing characters (not blanks)
- The word INACTIVE followed anywhere in the character string by the word MISSION

is selected.

Example 3

In this example:

```
Text ID      ==> INIT+++                    First word of event
Text String  ==> (Enter Below):             Text string (incl. Text-ID)
*INACTIVE*MISSION
```

any message whose first word:

- Starts with `INIT`
- Has three positions containing characters (not blanks)
- The word INACTIVE followed **at the end** of the character string by the word MISSION

is selected.

Note that this text string definition will match only if the message ends with `MISSION` and **no other characters or blanks follow**. If the message contains any characters or blanks after the `N`, then you must code the text string with an asterisk after the word MISSION in Example 2.

Example 4

For better performance, use wildcard characters discriminately. For example:

```
Text-ID      ==> *
Text String  ==> (Enter Below)
*STARTED*
```

searches *every* message for the word STARTED. It would be more efficient if you did the following instead:

```
Text-ID      ==> IM9161I
Text String  ==> (Enter Below)
*STARTED*
```

which searches only IM9161I messages for the word STARTED.

Example 5

The Rule Processor treats each word of the Accounting Information (Acct Info) field separately. Words of accounting information are denoted by commas.

For example, if the job card has the following accounting information:

```
(3912,PROD,AAO)
```

the Rule Processor would process 3912, PROD, and AAO separately.

For the above example, pattern matching could be specified as:

Table 16. Pattern Matching for Acct Info Field

Character String	Pattern	Results
(3912,PROD,AAO)	*	Match
(3912,PROD,AAO)	*, PROD	Match
(3912,PROD,AAO)	*, PROD, *	Match
(3912,PROD,AAO)	3912,*,AAO	Match
(3912,PROD,AAO)	3*,P*,A*	Match
(3912,PROD,AAO)	PROD	No Match

Table 17 shows some examples of pattern matches.

Table 17. Example of Pattern Matches

Character String	Pattern	Results
ABCDEF	ABC++F	Match
ABCDEF	ABC+F	No match
ABCDEF	A*	Match
ABCDEF	*D+F	Match
ABCDEF	A+C*	Match
ABCDEF	A*F	Match
ABCDEF	A*+F	Match
ABCDEF	A+FF	No match
ABCDEF	A*BCDEF	Match
ABCDEF	A+B*	No match
ABCDEF	'A*BCDEF'	No match
ABCDEF	¬XYZ	Match
ABCDEF	¬*BCD*	No match

Additional Notes

The single quote is used a toggle switch for pattern matching and can not be coded in the Text String field of the Selection Criteria panel or Variable Depencencies panel. Pattern matching must be used in place of single quotes.

For example: On the Variable Dependencies panel, coding

```
WORD3 EQ +IS MESSAGE
```

will result in a match with:

```
IJK5868 THIS 'IS A' TEST MESSAGE
```

In the Text String field, coding:

```
*IS A*
```

Or

```
*TEST MESSAGE*
```

will also result in a match with:

```
IJK5868 THIS 'IS A' TEST MESSAGE
```

Debugging

For debugging purposes, to determine if the match is successful, route the message to a specific location, such as the Journal log or your TSO user ID, for verification of the pattern selection criteria.

Using Variables in the Rule Processor

The following sections describe how to use variables in the Rule Processor application.

Variables may be used in most fields of the Selection Criteria panels, the Action Specification panels, and the Variable Dependencies panel. The definitions for the individual fields include information about variable usage.

Defining Variables

The Rule Processor application treats any character string that begins with an ampersand sign (&) as a variable. Variable names can be up to 32 characters long, excluding the ampersand sign (&). The variable values can be up to 255 characters long.

There are two types of variables that can be used in Rules:

- SHARED variables from the BBI-SS PAS SHARED variable pool.
- EVENT variables created by the Rule Processor application; they exist for duration of the event.

  EVENT variables consist of:

  – WORDxxx variables, where xxx represents a number (for example, &WORD4 represents the fourth word of the message text)

  – AutoOPERATOR-supplied variables

  See Table 18 on page 169 for a list of EVENT variables.

Resolving Variables

When resolving the contents of variables:

- The variable names and the variable values are always converted to uppercase.
- Conversion operations are performed.

  If the variables contain numeric values, conversion operations are performed automatically. For example, if you have two variables where `&ABC=5` and `&DEF=0005`, these both equal `5`.

  Variable values can also contain combinations of numerics and characters. When numerics and characters are combined, the conversion operation is a little different. For example, if you have two variables where `&ABC=XYZ0005` and `&DEF=XYZ5`, these values are not equal.
- The following search order is used:

  a. BBI-SS PAS SHARED variables

  b. EVENT variables (WORDxxx variables and AutoOPERATOR-supplied variables)

  Variables that are not found result in a null value (zero length).

  For example, if the SHARED variable pool contains a variable named &WORD1, the value from the SHARED pool is used instead of the actual first word of the event text.

When the Rule Processor resolves the contents of variables, the Rule Processor accepts two styles of variable specifications: simple variables (formerly referred to as CLIST-style) and compound variables.

Important
The compound variables are resolved similar to REXX variables but **they are not REXX stem variables** and do not behave like REXX stem variables (do not refer to any REXX manuals for information about the behavior of AutoOPERATOR compound variables).

The following sections describe these two types of variables.

Simple Variables

When an ampersand sign (&) prefixes text in a field in the Rule Processor fields, it denotes a simple variable name. The end of a variable name is designated by one of the following:

Blank	
Period	(.)
Comma	(,)
Ampersand	(&)
Apostrophe	(')
Hyphen	(-)
Asterisk	(*)
Plus sign	(+)
Forward slash	(/)
Backslash	(\)
Colon	(:)

Example 1

When:

Variable `&MSGID` contains the characters `IEF1234I`

Variable `&TEST` contains the characters `ABCD`

The combination:

```
&MSGID.&TEST
```

is resolved as two separate variables:

```
IEF1234IABCD
```

Compound Variables

When an ampersand sign (&) prefixes text in a field in the Rule Processor fields, it denotes a simple variable name. If the ampersand sign is directly followed by an exclamation mark (!) and if there are periods in the variable name, the Rule Processor resolves the contents of the compound variable differently.

The first node of a compound variable precedes the first period and is called the stem. For example, in the variable `&!ABC.MYNAME`, the stem is `ABC`.

The stem of a compound variable is not resolved to a value and **two passes** are used to resolve the variable.

In another example, suppose you have the following three variables and values:

```
ABC = FIRST
MYNAME = SECOND
ABC.SECOND = THIRD
```

The Rule Processor resolves the variable:

```
&!ABC.MYNAME
```

as `THIRD`.

Variable resolution for this compound variable occurs in this sequence:

1. `ABC` is the stem and is not resolved.
2. In the first pass, the variable following the stem is resolved and `MYNAME` becomes `SECOND`.
3. After the first pass, the variable value is now `ABC.SECOND`.
4. In the second pass, the variable `ABC.SECOND` is resolved to `THIRD`.

   Therefore: `&!ABC.MYNAME` is resolved by the Rule Processor to `THIRD`.

Note: The exclamation mark (!) might have a different representation when used on international keyboards. As a substitute, use the equivalent symbol for the hexadecimal value X'5A'.

The end of a variable is designated by one of the following:

Blank	
Comma	(,)
Ampersand	(&)
Apostrophe	(')
Hyphen	(-)
Asterisk	(*)
Plus sign	(+)
Forward slash	(/)
Backslash	(\)
Colon	(:)

If the Rule Processor cannot find a value for the variable, it is resolved as a null value. For example, if the Rule Processor cannot find a value for variable `&!ABC`, the Rule Processor assumes it is a null value.

For compound variables, if the variable `&!CICS.SYSA` does not have a value, the Rule Processor also assumes it is a null value.

In another example, if:

```
&!SHIFT1.CICSTAT = SCHEDULED
CICSTAT = not defined
```

The following combination:

```
&!SHIFT1.CICSTAT
```

is resolved as:

```
SCHEDULED
```

Using Variables in Rule Processor Application Fields

To use simple variables in the Rule Processor application fields, enter the variable name in the field prefixed with an ampersand (&). On the Variable Dependencies panel, you do not have to prefix the variable name with an ampersand for simple variables.

To use compound variables, enter the variable name in the field prefixed with an ampersand and an exclamation mark (&!). On the Variable Dependencies panel, you must still prefix the variable name with an ampersand and exclamation mark (&!) for compound variables.

The only exception to these two rules is for the `Set Variable` field. In this field, you do not need to use the special characters for either simple or compound variables since any value entered in this field is treated as a variable.

Setting Variables

On the Action Specification panel for all event types, you can use the field:

```
Set Variable     ===>                                ===>
```

to set a variable to a value.

For simple variables, enter the variable name in the first field and the value in the second field. For example:

```
Set Variable     ===>DAYOFWEEK                       ===>MONDAY
```

As a result, the SHARED variable `DAYOFWEEK` has a value of `MONDAY`.

For a variable with more than one node, separate the node names with an extra period:

```
Set Variable     ===>ABC&JOBNAME..&DAY.              ===>PRINTIT
```

As a result, if `JOBNAME` is a variable with a value of `MVSJOB` and `DAY` is a variable with a value of `SUN`, the resulting SHARED variable named `ABCMVSJOB.SUN` has a value of `PRINTIT`.

Variable Modifiers

By appending a modifier to the end of a variable, it is possible to perform certain operations on its contents before the variable is evaluated. Currently, two operations may be performed: **substringing** and **trimming**.

Trimming and substringing techniques further increase the power of a Rule and allow even more tasks to be accomplished. These operations are valid for both simple variables and compound variables.

The following sections describe:

- “Variable Substringing” on page 166
- “Trimming” on page 168

Variable Substringing

To access only a part (a substring) of a variable, specify a starting position and an optional length of the substring. To do this, append a colon (:) to the end of the variable name followed by a numeric designating the base position. Positions start at 1 (not zero) and may not be longer than the length of the variable.

Optionally, another colon may follow a substring specification which can designate the total length of the character string to evaluate.

Defaults: The defaults are 1 for a substring starting position and 255 for the variable length. If a substring specification extends past the end of a variable, only the portions that could properly be evaluated are returned.

Invalid Specifications: If a substring specification cannot be properly evaluated because of an invalid specification, it will be evaluated to the extent that specifications are valid.

Important Restriction
Starting position and length values must be specified explicitly. You cannot use variables to specify the starting position and the length of the substring. You cannot nest substring statements.

Examples: For the following examples, assume the variable ABC contains the character string `THIS IS A TEST`.

Example 1

By specifying:

```
&ABC:11
```

the string is resolved as:

```
TEST
```

Example 2

The string:

```
&ABC:1:4
```

is resolved as:

```
THIS
```

Example 3

The string:

```
&ABC:11:38
```

still returns the string:

```
TEST
```

Example 4

The string:

```
&ABC:6:NOT
```

is resolved as:

```
IS A TEST:NOT
```

This example shows that you can add more characters that are appended to the value returned from the substringing operation.

Trimming

Variable trimming means removing leading and trailing blanks from a variable's contents. Use trimming by appending a slash (/) to the end of a specified variable name. The slash must follow any substring specifications if both are used together.

For example, suppose you want to minimize the length of an ALERT text so it fits on a single line. To do this, you will want to eliminate the blanks that are padded to variables &IMFOJOB, &IMFJCLAS, and &WORD4. The result is:

```
JOB &IMFOJOB/ IN CLASS &IMFJCLAS HAS ABENDED WITH CODE &WORD4/
```

Trimming can be a useful tool for instring matches that occur when a particular string is contained in another string.

Assume that you want to find out whether the current job is one of 20 jobs whose names are contained in a character string, one following the other and delimited by a blank. This variable is called &TARGETJOB and bears the contents:

```
JOB1 JOB2 PRODJOB TESTJOB ....
```

The current jobname is contained in the variable &IMFOJOB (which is always 8 characters long and padded with blanks). For this example, assume the contents of `IMFOJOB` is `JOB2` followed by 4 blanks. By specifying:

```
&TARGETJOB EQ *&IMFOJOB/*
```

a match can be determined.

If the trim indicator is omitted, the expression would have evaluated as false because the Rule Processor will try to find the characters JOB2 followed by 4 blanks in the target string.

Note: The asterisks in this example are examples of using pattern matching. Refer to “Using Pattern Matching on Selection Criteria Panels” on page 156 for information about pattern matching.

List of AutoOPERATOR-Supplied EVENT Variables

EVENT variables are variables that are used by the Rule Processor application and exist for the duration of the event.

EVENT variables consist of:

- AutoOPERATOR-supplied variables
- WORDxxx variables, where xxx represents a number (for example, &WORD4)

These variables may also be available for use by AutoOPERATOR EXECs. For more information about variables and AutoOPERATOR EXECs, refer to the variables discussion in the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* and the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.

When creating Rules for the ALRT event type, additional variables are available. Refer to Table on page 174 for a list.

The following table shows all the EVENT variables supplied by AutoOPERATOR.

Table 18. AutoOPERATOR-Supplied EVENT Variables

Variable Name	Description
&IMFACCTG	Contains all accounting fields for a particular event. The accounting field values are separated by blanks. Maximum length is 142.
&IMFALID	The alarm ID associated with an alarm created by MAINVIEW Alarm Manager.
&IMFALPRI	The user-assigned priority of the alarm. Possible values are: **1** Critical **2** Major **3** Minor **4** Warning **5** Informational **6** Clearing
&IMFALQID	The name of the queue to which the alarm was assigned.
&IMFALRM	Contains either Y (sound an alarm) or N (do not an sound alarm).
&IMFCLOCK	Is a number derived from the first word of the S/390 Time-of-Day clock. The value is incremented approximately once every 1.05 seconds. Each Rule that fires has its own copy of the &IMFCLOCK variable, but the variable is not passed to Rule-initiated EXECs. You can use this variable to help construct names and values that are unique for the life of this IPL. For example, you can combine the IMFCLOCK and IMFRLID variables to create a unique variable to create a unique ALERT key. On the Alert Action(s) I panel in the Rules Processor, specify: Key ===> &IMFRLID::&IMFCLOCK

Table 18. AutoOPERATOR-Supplied EVENT Variables (Continued)

Variable Name	Description
&IMFCNTXT	The name of the context of the alarm.
&IMFCONID	Console ID of the message, if message was issued for a specific console.
&IMFCONNM	Console name to which the WTO was issued. Valid only for MVS SP4 and above. &IMFCONNM may be used to identify the origin of an MVS command. The contents of the variable (by origin) are: **Origin** **IMFCONNM Value** Rule Internal SDSF TSO user ID that issued the command Console Console Name where the command was issued
&IMFDAY	Three-character day of the week: MON, TUE, WED, THU, FRI, SAT, SUN.
&IMFDDNAM	Contains the DDNAME specified by the user to generate an external events (EXT event type). EXT events are generated by using the SUBSYS= parameter on a DD statement in JCL. Refer to “EXT Events” on page 53 for more information about EXT events.
&IMFDOMID	The DOM ID associated with a WTO that caused a Rule to fire.
&IMFEVFRD	The number of Rules that have fired for a specific event.
&IMFETYPE	The event type that caused the Rule to fire. If a Rule invokes an EXEC, then IMFETYPE contains the value from the Rule that invoked the EXEC. Possible values for IMFETYPE are: • MSG • CICS • CMD • JRNL • IMS • ALRT • DB2 • TIME • ALRM • EXT • VAR • MQS • JES3 For more information about these event types, refer to “Describing Events” on page 43.
&IMFGROUP	The RACF group ID for the address space that issued the event. The group ID is taken from the GROUP= parameter on the job card.
&IMFJCLAS	Job class from the job card of the batch job that has generated the message.

Table 18. AutoOPERATOR-Supplied EVENT Variables (Continued)

Variable Name	Description
&IMFJNUM	The JES job number of the job, STC, or TSU that issued the message. It is a fixed length five-digit or a variable length value depending on the setting of the IMFJNUM option in member AAOPRMxx. IMFJNUM can also contain blanks (one or five characters as appropriate) for WTOs that are issued by non-JES tasks, such as a STC started under MSTR. When IMFJNUM=5 (the default setting) and the job number is greater than 99,999 (for example, T0100000, S0999999, etc.) are encountered, IMFJNUM will be null (zero length).
&IMFJTYPE	Type of job issuing message: **J** Batch Job **T** TSO User **S** Started Task
&IMFLPROD	The name of the product associated with the alarm.
&IMFLTYPE	A literal value associated with the alarm; possible values can be START or STOP.
&IMFLUSER	The user-specified user ID associated with the alarm.
&IMFMPFAU	Contains YES, NO, or an 8-character token set by the user in SYS1.PARMLIB.
&IMFMPFSP	Contains either YES or NO.
&IMFMSTYP	Contains a 2-character variable for the message type. This variable is only for the CMD and MSG event types. Valid values for the first character can be: **N** A regular WTO **W** A regular WTOR **M** A major line of a multi-line WTO (MLWTO) Valid values for the second character can be: **C** Command **R** Command response
&IMFOASID	Originating Address Space ID (ASID) of the message. For IMFEOM, it is set to the ASID that is being terminated.
&IMFODATE	Date when the message was issued.

Table 18. AutoOPERATOR-Supplied EVENT Variables (Continued)

Variable Name	Description
&IMFOJOB	For WTOs, &IMFOJOB contains the job or started task that issued the WTO. For CICS messages, &IMFOJOB contains the CICS region name that the subsystem issued the message for, which is useful when monitoring multiple CICS regions with one BBI-SS PAS. For DB2 messages, &IMFOJOB contains the DB2 region name that the subsystem issued the message for, which is useful when monitoring multiple DB2 regions with one BBI-SS PAS. For IMS messages, &IMFOJOB contains: • The IMS job name for IMS MTO messages • The IMS job name for commands (and their responses) entered from AutoOPERATOR • The originating LTERM for commands (and their responses) entered from an IMS LTERM For Journal messages issued by an EXEC, &IMFOJOB contains the user ID of the person who invoked the EXEC.
&IMFOQID	CICS transient data queue name if source of message is CICSTD.
&IMFORGN	**Type** — **Description** **MSG** — Jobname of issuer **CMD** — Jobname of issuer **ALRT** — AutoOPERATOR jobname issuing the ALERT **DB2** — DB2 address space jobname **CICS** — CICS address space jobname **IMS MSG** — IMS region name **IMS CMD** — IMS LTERM name **JES3** — JES3 SYSID name **JRNL** — Jobname on whose behalf the Journal message was issued **TIME** — AutoOPERATOR jobname **EXT** — Jobname of issuer **ALRM** — Name of issuer
&IMFORGSS	BBI Subsystem ID of the BBI-SS PAS.

Table 18. AutoOPERATOR-Supplied EVENT Variables (Continued)

Variable Name	Description

&IMFOTIME	Time when the message was issued. Valid only for messages captured through the Rule Processor.
	
	The format is:
	
	`HH.MM.SS`
	
	for MSG event types. For all other event types, the format is
	
	`HH:MM:SS`

&IMFPCMD	The PCMD associated with the alarm.
&IMFREPLY	Reply ID of the WTOR message.
&IMFRLFRD	Number of times a Rule was fired.
&IMFRLID	Rule identifier.
&IMFRLMAT	Number of times the Rules search criteria was matched.
&IMFRLSET	Name of the Rule Set the Rule belongs to.
&IMFRLSTA	Rule status: **TEST** Indicates the Rule is in a TEST state. **ACTIVE** Indicates the Rule is in ACTIVE state.
&IMFRUSER	The user ID from the USER= parameter on the job card.
&IMFSCOPE	The name of the scope associated with the alarm.
&IMFSTOKN	Address Space STOKEN. This name is unique for the life of the IPL.
&IMFSYSID	**Type** **Description** **MSG** Jobname of issuer **CMD** Jobname of issuer **ALRT** AutoOPERATOR jobname issuing the ALERT **DB2** DB2 address space jobname **CICS** CICS address space jobname **IMS MSG** IMS region name **IMS CMD** IMS LTERM name **JES3** JES3 SYSID name **JRNL** Jobname on whose behalf the Journal message was issued **TIME** AutoOPERATOR jobname **EXT** Jobname of issuer **ALRM** Name of issuer
&IMFTEXT	Message text that caused the EXEC to be scheduled.

Table 18. AutoOPERATOR-Supplied EVENT Variables (Continued)

Variable Name	Description
&IMFTOKEN	Token ID of the message. Same as hardcopy ID. Used to attach multi-line WTO Minor/Major Lines.
&IMFVIEW	The name of the view associated with the alarm.
WORDxxx	Where xxx represents a number (for example, &WORD4). These are variables created by the Rule Processor to represent the different words in the text of the event that fire a Rule. The variable &WORD3 should resolve to the third word of the text of the event. Word variables are delimited by blanks and commas in the text.

List of Variables for ALRT-initiated Rules

The following table shows the variables available when you create Rules for the ALRT event type.

Table 19. Variables Available for Creating Rules for ALRT Events

Name	Contents	Length/Format
AMFKEY	Key of the ALERT	1-64 / Character
AMFTEXT	Text of the ALERT	0-255 / Character
AMFALARM	Alarm value of the alert	1 / Y (YES) or N (NO)
AMFCOLOR	Color of ALERT	6 / As specified by COLOR parameter
AMFEDIR	Increase or decrease the priority of the ALERT when it is escalated	1 / Character (U or D)
AMFEDISP	Keep or delete the ALERT at the final escalation level	1 / Character (K or D)
AMFEEXEC	Name of EXEC and EXEC parameters scheduled at final escalation priority	0 to 255 / Character
AMFEINT1 AMFEINT2 AMFEINT3 AMFEINT4 AMFEINT5 AMFEINT6	Number (in minutes) from 0 to 9999	4 / Numeric (or null)
AMFEXEC	EXEC and EXEC parameters associated with the ALERT	0-256 / Character
AMFHELP	Extended Alert member name	8 / Character
AMFIDATE	Date ALERT was issued	9 / DD-MMM-YY
AMFITIME	Time ALERT was issued	8 / hh:mm:ss

Table 19. Variables Available for Creating Rules for ALRT Events

AMFORGN	Origin of ALERT	1-8 / Character
AMFPCMD	Primary command specified in ALERT	0-256 / Character
AMFPRIOR	Priority of ALERT	13 / As specified in PRIORITY parameter
AMFPSYS	Value for SYSTEM keyword (could be either YES or NO)	1 / Character (Y or null)
AMFPUB	Value of the Publish keyword when an ALERT is created	2-7/ADD, REPLACE, or NO
AMFQUEUE	Name of queue for ALERT	8 / Character
AMFRTAIN	Specifies whether or not to retain an ALERT across BBI-SS PAS warm and cold starts	1 / Character (Y or N)
AMFSSID	System from which ALERT was issued	8 / Character
AMFTGT	Target to which ALERT was issued	1-8 / Character
AMFUDATA	User data string	0-256 / Character
AMFUSER	Name of the user ID that the ALERT is addressed to	8 / Character

Chapter 9. Managing Rules and Automation Using the Automation Control Panel

After you have created Rules and Rule sets, you can use option 2, Display / Modify Rules and Rule Sets, from the Automation Menu to display the Automation Control Menu.

The Automation Control panel, shown in Figure 99, has three areas that provide:

- An overview of automation status and strategy (top area)
- An overview of the automation statistics (middle area)
- A scrollable list of Rule Sets in the Automation library (bottom portion), indicating:
  - Whether a Rule Set is enabled or disabled
  - Number of Rules in the Rule set
  - Number of times AutoOPERATOR has fired Rules in a Rule Set
  - How many events were filtered through a Rule Set
  - Whether Rule Set filtering is enabled or disabled
  - Date and time of the firings
  - What automation strategy a Rule set has been assigned

```
BMC Software --------------------- Automation Control ---------------- AutoOPERATOR
COMMAND ===>                                                TGT ===> MVR3
Primary commands: Add, Statshow, Cmdshow                    DATE --- 01/02/14
                                                            TIME --- 13:23:05
Automation Status     ===> ACTIVE         (Active, Inactive)
Automation Strategy   ===> INDIVIDUAL     (Individual, All, First, Qualified)
Honor MPF Suppression ===> NO             (NO/YES)

                        Automation Statistics
  Total Events                 624   Display suppressed              0
  Events Handled                49   Hardcopy suppressed             0
  Current arrival rate    3  / sec   Rule generated Alerts           6
  Peak arrival rate      10  / sec   Rule invoked Execs              0
                         Automation Library
LC CMDS --- (S)elect, (E)nable, (D)isable, (T)est, (SA)ve
            (M)ove, (B)efore or (A)fter, (F)ilter Criteria
LC   Rule-Set Status   Rules     Fired  Filtered    Date      Time   Strategy
___  RULQUAL1 ENABLED      6        46       624d 01-FEB-01 13:18:50 QUAL
___  RULJRNL1 ENABLED    108        93       183  01-FEB-01 13:18:49 FIRST
___  RULJRNL2 ENABLED     53         6         8  01-FEB-01 13:18:49 FIRST
___  RULJRNL3 ENABLED     59        46        32  01-FEB-01 13:18:49 ALL
___  RULJRNL4 ENABLED     56         9        17  01-FEB-01 13:18:49 FIRST
___  RULJRNL5 ENABLED     37       346        46  01-FEB-01 13:18:49 ALL
___  RULJ911  ENABLED      6        46        46  01-FEB-01 13:18:49 FIRST
___  AAORULCC DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULJB DISABLED   N/A   N/A       N/A          N/A    N/A
```

Figure 99. Automation Control Panel

This panel is scrollable; use the PF7/UP and PF8/DOWN keys to scroll through the list of Rule Sets.

Use the Automation Control panel to:

- Set your Automation Strategy
- See the Automation Statistics
- Manage groups of Rules, called Rule Sets, to provide simpler management of your basic automation

  This includes using Rule Set filtering which allows you to limit the number of events and the types of events that are passed through a Rule Set. For more information about using Rule Set filtering, refer to “Filtering Events for Rule Sets” on page 188.

The following sections describe setting automation strategy for your Rule Sets.

Setting Automation Status/Strategy (Upper Portion)

```
BMC Software ----------------- Automation Control ---------------- AutoOPERATOR
COMMAND ===>                                                  TGT ===> MVR3
Primary commands: Add, Statshow, Cmdshow                      DATE --- 01/02/14
                                                              TIME --- 13:23:05
Automation Status     ===> ACTIVE         (Active, Inactive)
Automation Strategy   ===> INDIVIDUAL     (Individual, All, First, Qualified)
Honor MPF Suppression ===> NO             (NO/YES)
```

Figure 100. Automation Control Panel: Upper Portion

Automation Status refers to whether automation is inactive or active. Inactive means that there is no automation taking place even if you have Rule Sets that are enabled. Use this field when you want to shut down automation quickly by typing `INACTIVE` in the field. To restart automation, type `ACTIVE` in this field.

Automation Strategy determines which Rules (and how many Rules) get fired to handle events. For example, an event might match the selection criteria for **more than one** Rule. If you decide that you want all the Rules to fire, you must specify the strategy ALL.

If you decide that you do **not** want all the Rules fired, you must choose a different strategy. The different strategy names are as follows:

- FIRST
- QUALIFIED
- INDIVIDUAL

Choosing an Automation Strategy is a very important decision and it can affect how you create Rules. Refer to “Choosing an Automation Strategy” on page 186 for a complete discussion about how to set Automation Strategy.

Using Primary Commands

You can enter the following primary command in the COMMAND field of the Automation Control panel.

Command	Parameters	Description
ADD	xx|RULxxxxxx	If xx is specified, causes a new Rule Set to be created with the name AAORULxx, where xx is the suffix ID of the Rule Set. The default status of the new Rule Set is ENABLED. If RULxxxxx is specified, a new Rule Set is created with the name RULxxxxx, where xxxxx is a user-specified suffix. The suffix can be five characters or fewer. The default status of the new Rule Set is ENABLED. **Note:** The new Rule Set is **not saved** until a SAVE command is performed on the Rules Set Overview panel. If you change the status of the new Rule Set to DISABLED before you save the Rule Set, the Rule Set is permanently deleted from storage. Rule Sets cannot be named AAORULBx, where x is any alphanumeric value. This naming pattern is reserved for AutoOPERATOR use only. Valid characters that can be used in the Rule Set name are letters A through Z, numbers 0 through 9, the pound sign (#), the at sign (@), and the dollar sign ($).

Command	Parameters	Description
STATSHOW	ON|OFF	Shows or hides the Automation Statistics from the Automation Control display. Hiding the Automation Statistics allows for more scrollable lines in the Automation Library section of the display. Enter on the COMMAND line: `STATSHOW OFF`: To hide Automation Statistics `STATSHOW ON`: To show display Automation Statistics `STATSHOW`: Toggles between showing and hiding Automation Statistics. If Automation Statistics are displayed, hides them. If Automation Statistics are not displayed, displays them. The STATSHOW setting remains in effect until you change it.
CMDSHOW	ON|OFF	Shows or hides the Line Command Help from the Automation Control display. Hiding the Line Command Help allows for more scrollable lines in the Automation Library section of the display. Enter on the COMMAND line: `CMDSHOW OFF`: To hide Line Command Help `CMDSHOW ON`: To show Line Command Help `CMDSHOW`: Toggles between showing and hiding Line Command help. If Line Command Help is displayed, hides it. If Line Command Help is not displayed, displays it. The CMDSHOW setting remains in effect until you change it.

Describing Fields

Following is a description of the fields.

Field	Description
Automation Status	Indicates whether the Rule Processor itself is ACTIVE or INACTIVE. This field can be modified by typing over the field.
Automation Strategy	Controls which Rules will be fired when an event meets the selection criteria for more than one Rule. Valid values are FIRST, QUALIFIED, ALL, and INDIVIDUAL. This field can be modified by typing over the field. Rules are organized by Rules Sets and the search order for a Rule depends on the order in which the Rule Sets appear in the "Automation Control Panel" on page 177. Refer to "Choosing an Automation Strategy" on page 186 for information about selecting an Automation Strategy.
Honor MPF Suppression	Provides support for the IBM Message Processing Facility (MPF). **Note:** This field is supported only if you have the MAINVIEW AutoOPERATOR for OS/390 option. If NO is specified, all messages are sent to the Rule Processor regardless of any action that MPF might have taken with the message. If YES is specified, any message that is suppressed with MPF is **not** sent to the Rule Processor. The message does not appear in the Event Activity Statistics application.

Viewing Automation Statistics (Middle Portion)

```
                        Automation Statistics
  Total Events                 624   Display suppressed             0
  Events Handled                49   Hardcopy suppressed            0
  Current arrival rate    3  / sec   Rule generated Alerts          6
  Peak arrival rate      10  / sec   Rule invoked Execs             0
```

Figure 101. Automation Control Panel: Middle Portion

All of the automation statistics shown on this panel are statistics accumulated since the last AutoOPERATOR cold start or since the last time statistics was reset using the `.RESET STATS` command. Following is a list of the fields in the Automation Statistics section of the Automation Control panel.

Note: You can hide this portion of the panel to allow for more scrollable lines in the bottom portion of the panel by using the STATSHOW primary command. Refer to “Using Primary Commands” on page 179 for more information.

Field	Description
Total Events	Is the total number of events that have been presented to the Rule Processor as potential candidates for automation.
Display Suppressed	Is the total number of times the Rule Processor suppressed the display of an event.
Events Handled	Is the total number of times a Rule was fired on a matched event.
Hardcopy Suppressed	Is the total number of times the Rule Processor has suppressed messages destined for SYSLOG or the Journal.
Current Arrival Rate	Is the rate at which events are presented to the Rule Processor during the last 10 seconds.
Rule generated Alerts	Is the total number of ALERTs generated directly from the Rule Processor. This field does not include any ALERTs that are generated by the IMFEXEC ALERT command.
Peak arrival rate	Is the highest arrival rate of events passed to the Rule Processor since activation.
Rule invoked EXECs	Is the total number of times the Rules processor has scheduled an EXEC.

Grouping Rules into Rule Sets (Bottom Portion)

Use this portion of the panel to see all the Rule Sets, their status, and how many Rules have been fired.

```
                          Automation Library
LC CMDS --- (S)elect, (E)nable, (D)isable, (T)est, (SA)ve
            (M)ove, (B)efore or (A)fter, (F)ilter Criteria
LC   Rule-Set Status   Rules     Fired  Filtered    Date      Time   Strategy
___  RULQUAL1 ENABLED      6        46       624d 01-FEB-01 13:18:50 QUAL
___  RULJRNL1 ENABLED    108        93       183  01-FEB-01 13:18:49 FIRST
___  RULJRNL2 ENABLED     53         6         8  01-FEB-01 13:18:49 FIRST
___  RULJRNL3 ENABLED     59        46        32  01-FEB-01 13:18:49 ALL
___  RULJRNL4 ENABLED     56         9        17  01-FEB-01 13:18:49 FIRST
___  RULJRNL5 ENABLED     37       346        46  01-FEB-01 13:18:49 ALL
___  RULJ911  ENABLED      6        46        46  01-FEB-01 13:18:49 FIRST
___  AAORULCC DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULJB DISABLED   N/A   N/A       N/A          N/A    N/A
```

Figure 102. Automation Control Panel: Bottom Portion

This panel also shows if Rule Set filtering has been turned on for a Rule Set. You turn on Rule Set filtering for a Rule Set with the line command `(F)ilter Criteria`.

Rule Sets that have been set to filter events but the filter is turned off are noted by a `d` in the `Filtered` column; for example:

```
LC   Rule-Set Status   Rules     Fired  Filtered    Date      Time   Strategy
___  RULQUAL1 ENABLED      6        46       624d 01-FEB-01 13:18:50 QUAL
```

Rule Sets that have been set to filter events and the filter is turned on will display an `e` in the `Filtered` column. A blank indicates that no filter has been created for the Rule Set. For more information about Rule Set filtering, refer to “Filtering Events for Rule Sets” on page 188.

To see more information about any one Rule Set, select the Rule Set with the `(S)elect` line command in the `LC` column and view the Rule Set Overview panel. Refer to Chapter 10, “Using the Rule Set Overview Panel to Manage Rules in Rule Sets” on page 197 for more information about the Rule Set Overview panel.

The fields in the Automation Library section of the Automation Control panel include:

Field	Description
Rule-Set	Is the PDS member name of a Rule Set within the Automation Library where all Rules are contained. Rule Sets names cannot use AAORULBx, where x is any alphanumeric value. This naming pattern is reserved for AutoOPERATOR use only.
Status	Is the state of the Rule Set. Valid values are ENABLED, DISABLED, and TEST. – ENABLED means that the Rules within a Rule Set are available to process events. However, an ENABLED Rule Set can contain DISABLED Rules and TEST Rules. Rule Set names specified with the parameter `RULESET=` in BBPARM member BBISSP00 are automatically ENABLED after a cold start. – DISABLED means that the Rules within a Rule Set are not available to process events. Rule Sets **not** specified with the parameter `RULESET=` in BBPARM member BBISSP00 are DISABLED after a cold start. – TEST means that the Rules within a Rule Set are fired in test mode. This means that the only action to be performed is to invoke the appropriate EXEC with the variable IMFRLSTA set to TEST.
Rules	Is the number of Rules contained in the Rule Set.
Fired	Is the number of times Rules within the Rule Set were fired.
Filtered	Is the number of events filtered through a Rule Set. For more information about filtering events through Rule Sets, refer to “Filtering Events for Rule Sets” on page 188. In the space immediately following the `Filtered` column, there can be one of three values: d Indicates Rule Set filtering is disabled for a Rule Set e Indicates Rule Set filtering is enabled for a Rule Set blank Indicates no Rule Set filtering has been set for a Rule Set
Date	Is the date on which the Rule Set was last activated in the format DD-MMM-YY.
Time	Is the time when the Rule Set was last activated.
Strategy	Shows a value of: FIRST, QUALIFIED, ALL, or INDIVIDUAL. Strategy will be displayed only if an automation strategy of INDIVIDUAL was specified in the upper portion of the panel. Refer to “Choosing an Automation Strategy” on page 186.

Using Line Commands

You can hide this portion of the panel to allow for more scrollable lines in the bottom portion of the panel by using the CMDSHOW primary command. Refer to “Using Primary Commands” on page 179 for more information.

Use the following line commands in the LC column:

Command	Description
Select|S	Presents a summary of all the Rules within a Rule Set (displays the Rule Set Overview panel). This command does not work on a Rule Set that is DISABLED.
Enable|E	Makes all the enabled Rules in the Rule Set eligible to participate in event selection.
Disable|D	Stops the selection criteria filtering process for all Rules within the Rule Set. If the Rule Set is not saved, then all changes made to the Rule Set will be lost.
Test|T	Makes the Rules within a Rule Set available to message traffic, but in test mode. This means that the only action to be performed will be invoking the appropriate EXEC with variable IMFRLSTA set to TEST.
SAve|SA	Saves a Rule Set in the Automation Library.
Move|M	Moves an ENABLED Rule Set to a different location within the Automation Library list of active Rule Sets. The sequence of the Rule Sets in the Library is important if you specify an automation strategy of FIRST because this means the first Rule within a Rule Set is fired when it matches an event. Used in conjunction with the After|A and Before|B line commands.
Before|B	Indicates the line within the Automation Library List of ENABLED Rule Sets before which an ENABLED Rule Set is moved with the Move|M line command.
After|A	Indicates the line within the Automation Library List of ENABLED Rule Sets after which an ENABLED Rule Set is moved with the Move|M line command.

Choosing an Automation Strategy

Automation Strategy determines which Rules (and how many Rules) get fired to handle events.

For example, an event might match the selection criteria for **more than one** Rule. If you decide that you want all the Rules to fire, you must specify the strategy ALL.

If you decide that you do **not** want all the Rules fired, you must choose a different strategy. The different strategy names are as follows:

- FIRST
- QUALIFIED
- INDIVIDUAL

If you choose an automation strategy of ALL, FIRST, or QUALIFIED, all Rule Sets are searched for Rules to match a certain event. You can specify ALL, FIRST, or QUALIFIED for each of the Rule Sets on this panel.

Choose a strategy of INDIVIDUAL if you want to set an Automation Strategy for **each Rule Set**.

If you do not select a strategy, the default is FIRST.

Selecting FIRST

Selecting FIRST means that **only the first Rule** that matches an event will be fired.

If you select FIRST, the order of your Rules within a Rule Set and the order of Rule Sets in the Automation Library is very important because this order determines which Rule fires.

Selecting ALL

Selecting an Automation Strategy of ALL means that all the Rules within all Rule Sets that match an event will be fired.

When you select this strategy, potentially more than one can fire to handle a single event.

Selecting QUALIFIED

Selecting QUALIFIED means that when an event matches more than one Rule, the Rule Processor fires the Rule that **specifies the greatest number of fields** on the Selection Criteria panel.

For example, RUL0001 specifies selection criteria in the Text-ID, Text-String, Jobname, and Type fields (total number of fields is 4). RUL0002 specifies selection criteria in the Text-ID, Text-String, and Jobname fields (total number of fields is 3). When the Automation Strategy is QUALIFIED, the selection criteria for both these Rules might match the event but only RUL0001 is fired because it matches on a greater number of fields.

For example, a specific event might match the selection criteria for 2 Rules. If you were using a strategy of ALL, they would both fire. If you were using a strategy of FIRST, the first Rule in the first Rule Set would fire.

By using QUALIFIED, the Rule Processor fires only the Rule that specified the larger number of criteria.

You might even have a situation where you specify QUALIFIED, and an event matches 2 Rules and both Rules have the same number of specified criteria. In this case, the **first Rule** encountered is fired.

Example 1 of using QUALIFIED: Two Rules, each specifying more than 1 field

Rule 1 specifies one field with the same value:

- Text-id = $HASP100

Rule 2 specifies 2 fields:

- Text-id = $HASP100
- Text String = * ON TSOINRDR

The message that is received is:

```
$HASP100 SUP3X    ON TSOINRDR
```

so Rule 1 matches on 1 criteria field and Rule 2 matches on 2 criteria fields. Therefore Rule 2 is more qualified and is the only Rule fired.

Example 2 of using QUALIFIED: Two Rules, each specifying only 1 field

Rule 1 Specifies the only 1 field, the same as Rule 1:

- Text-id = $HASP1*

Rule 2 specifies only 1 field:

- Text-id = $HASP10*

The same $HASP100 message as in the previous example is received. Both Rules have exactly 1 selection field that matches the message, so neither is more qualified than the other. The first Rule that AO encounters in its Rule search will fire.

Selecting INDIVIDUAL

If you select INDIVIDUAL automation strategy, you can specify that each Rule Set has its own strategy of FIRST, ALL, or QUALIFIED. The default value is FIRST. In this case, you can have several Rule Sets that will use FIRST and one Rule from each Rule Set can fire.

In other words, the strategy of Rules are applied for each Rule Set instead of for all active Rule Sets.

Use the Rule Set Overview Panel to select a strategy for a Rule Set.

Rule Set Filter

The Rule Set filter resolves two distinct requirements for a Rule Set:

- Filters events for the Rule Set to reduce the number of events with which each Rule in the Rule Set must be compared.
- Sets defaults for all Rules in the Rule Set. Currently, the default is limited to Criteria Match Rate Threshold.

Filtering Events for Rule Sets

A very important task for making your automation with Rules more efficient is to use Rule Set filtering. Using Rule Set filtering, you can limit the number of events and the types of events that are passed through a specific Rule Set.

Without Rule Set filtering enabled, every Rule Set is searched for matching Rules for every event the Rule Processor sees. With Rule Set filtering enabled, you can specify that events **must meet a Rule Set selection criteria** (called filter criteria) before a Rule Set is searched for matching Rules. This feature enables you to **selectively lessen the number of events that are passed through each Rule Set.**

For example, you can specify that a Rule Set named RULHASP will be searched for all events with a Text-ID of $HASP*. You can limit the search for matching Rules to a single Rule Set or group of Rule Sets.

In Figure 103, note the column `Filtered`. This value represents the number of events that the Rule Set was searched for matching Rules.

```
                        Automation Statistics
  Total Events                 624   Display suppressed              0
  Events Handled                49   Hardcopy suppressed             0
  Current arrival rate    3  / sec   Rule generated Alerts           6
  Peak arrival rate      10  / sec   Rule invoked Execs              0
                         Automation Library
LC CMDS --- (S)elect, (E)nable, (D)isable, (T)est, (SA)ve
            (M)ove, (B)efore or (A)fter, (F)ilter Criteria

LC   Rule-Set Status   Rules     Fired  Filtered    Date      Time   Strategy
___  RULQUAL1 ENABLED      6        46       624d 01-FEB-01 13:18:50 QUAL
___  RULJRNL1 ENABLED    108        93       183  01-FEB-01 13:18:49 FIRST
___  RULJRNL2 ENABLED     53         6         8  01-FEB-01 13:18:49 FIRST
___  RULJRNL3 ENABLED     59        46        32  01-FEB-01 13:18:49 ALL
___  RULJRNL4 ENABLED     56         9        17  01-FEB-01 13:18:49 FIRST
___  RULJRNL5 ENABLED     37       346        46  01-FEB-01 13:18:49 ALL
___  RULJ911  ENABLED      6        46        46  01-FEB-01 13:18:49 FIRST
___  AAORULCC DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULJB DISABLED   N/A   N/A       N/A          N/A    N/A
```

Figure 103. Automation Control Panel: Rule Set Filtering

So, the information for Rule Set RULJRNL1 looks like:

```
                        Automation Statistics

  Total Events                 624   Display suppressed              0
  Events Handled                49   Hardcopy suppressed             0
  Current arrival rate    3  / sec   Rule generated Alerts           6
  Peak arrival rate      10  / sec   Rule invoked Execs              0
                         Automation Library
LC CMDS --- (S)elect, (E)nable, (D)isable, (T)est, (SA)ve
            (M)ove, (B)efore or (A)fter, (F)ilter Criteria
LC   Rule-Set Status   Rules     Fired  Filtered    Date      Time   Strategy
___  RULQUAL1 ENABLED      6        46       624d 01-FEB-01 13:18:50 QUAL

___  RULJRNL1 ENABLED    108        93       183  01-FEB-01 13:18:49 FIRST
```

Rule Set RULJRNL1 contains a total of 108 Rules. Of them, 93 were fired for 183 events that were seen by (or filtered through) the Rule Set. However, a total of 624 events were seen by the Rule Processor. Therefore with filtering enabled, the number of events seen by Rule Set RULJRNL1 was lessened, enabling the Rules Processor to perform more efficiently.

In contrast, for Rule Set RULQUAL1, note the lowercase **d** in the `Filtered` column. The **d** means that Rule Set filtering has been disabled for RULQUAL1. Also, note that the number of events filtered through the Rule Set is 624, because every event seen by the Rule Processor was also seen by Rule Set RULQUAL1.

"Enabling Rule Set Filtering and Rule Set Match Rate" on page 190 describes how to enable Rule Set filtering.

Enabling Rule Set Filtering and Rule Set Match Rate

For existing Rule Sets, Rule Set filtering can be modified in two ways:

- From the Automation Control panel, type `F` in the line command (LC) column for the targeted Rule Set. Using this method, the filter changes are saved to the Rule Set BBPARM member immediately upon exiting the filter.
- From the Rule Set Overview panel, type **`Filter`** on the COMMAND line. Using this method, the filter is updated when you SAVE and exit the Rule Set.

The following example shows how a Rule Set with filtering enabled lessens the number of events seen by the Rule Set and makes automation with Rules more efficient.

To view the filter criteria for Rule Set RULJ911:

1. From the Automation Control panel, type `F` in the `LC` column next to the Rule Set (Figure 104).

```
BMC Software ----------------- Automation Control ---------------- AutoOPERATOR
COMMAND ===>                                                  TGT ===> MVR3
Primary commands: ADD, STATSHOW, CMDSHOW                      DATE --- 01/02/14
                                                              TIME --- 13:23:05
Automation Status     ===> ACTIVE         (Active, Inactive)
Automation Strategy   ===> INDIVIDUAL     (Individual, All, First, Qualified)
Honor MPF Suppression ===> NO             (NO/YES)

                        Automation Statistics
  Total Events                 624   Display suppressed              0
  Events Handled                49   Hardcopy suppressed             0
  Current arrival rate    3  / sec   Rule generated Alerts           6
  Peak arrival rate      10  / sec   Rule invoked Execs              0
                         Automation Library
LC CMDS --- (S)elect, (E)nable, (D)isable, (T)est, (SA)ve
            (M)ove, (B)efore or (A)fter, (F)ilter Criteria
LC   Rule-Set Status   Rules     Fired  Filtered    Date      Time   Strategy
___  RULQUAL1 ENABLED      6        46       624d 01-FEB-01 13:18:50 QUAL
___  RULJRNL1 ENABLED    108        93       183  01-FEB-01 13:18:49 FIRST
___  RULJRNL2 ENABLED     53         6         8  01-FEB-01 13:18:49 FIRST
___  RULJRNL3 ENABLED     59        46        32  01-FEB-01 13:18:49 ALL
___  RULJRNL4 ENABLED     56         9        17  01-FEB-01 13:18:49 FIRST
___  RULJRNL5 ENABLED     37       346        46  01-FEB-01 13:18:49 ALL
F__  RULJ911  ENABLED      6        46        46  01-FEB-01 13:18:49 FIRST
___  AAORULCC DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULJB DISABLED   N/A   N/A       N/A          N/A    N/A
```

Figure 104. Automation Control Panel: Viewing a Rule Set Filter

Figure 105 is displayed:

```
BMC Software  ------------------- Filter Criteria ---------------- AutoOPERATOR
COMMAND ===>                                                   TGT  --- JPP1

       Rule-set === RULJUN00

Text Description:
Text ID     ===> IM911*                            First word of message
Text String (Enter Below):


Event Type  ===> JRNL                              Type  of event ( ? for list)

Mode        ===> ENABLED                           (Status - ENABLED/DISABLED)

Ruleset criteria match rate threshold:
If matched    ===>               (Maximum # times matched within INTERVAL, 1-100)
in seconds    ===>               (Interval length, 1-99999 seconds)
then status ===>     (SUSPEND, DISABLE, NOACTION)



   Press ENTER for Variable Dependencies Panel,
   END to apply changes, CANCEL to cancel changes
```

Figure 105. Filter Criteria Panel: Viewing the Filter Crtieria

On the Filter Criteria panel, note how you can specify a Text-ID, Text String, and Event Type on the panel. The filter criteria for this Rule Set shows that all events with a Text-ID that begins with IM911 and are JRNL events will be seen by this Rule Set.

The following describes the fields on this panel:

Field Name	Description
Text ID	Fill in the Text-ID of the event you want to be seen by this Rule Set. This maximum length of the Text-ID is 16 characters. To have the Rule Set select events with more information, use the Text String field.
Text String	Specify message/event text to be matched in addition to the Text-ID. The maximum length of the Text-ID is 75 characters.
Event Type	Specify the event type for the events that will be seen by the Rule Set. You can specify: ALERT JES3 DB2 CMD MSG IMS EXT CICS JRNL

Note: For ecvent type MSG:

- If you journal the message, the entire message (including the reply ID) is written to the journal.
- If you use IMFTEXT the reply ID is stripped from the message.
- WORD1 is the first word after the reply ID.
- TEXT ID is the first word after the reply ID.

- TEXT STRING does not include the reply ID.

Mode Allows you to enable or disable Rule Set filtering for the Rule Set.

You can specify filter criteria and then disable the filtering for the Rule Set. This option means the Rule Set will be searched for matching Rules for every event the Rule Processor sees.

When Rule Set filtering is disabled, a lowercase `d` appears in the `Filtered` column of the Automation Control panel. On the Rule Set Overview panel, a field denotes whether the Rule Set filtering has been enabled or disabled.

Rule Set Criteria Match Rate Threshold

The Rule Set Criteria Match Rate Threshold provides a default Criteria Match Rate to be used for each Rule in this Rule Set that does not have a value specified. Refer to Chapter 5, “Creating Rules for Events: Using the Rule Processor Detail Control Panel” on page 55 for details on Criteria Match Rates within an individual Rule.

If matched Specifies the number of times a Rule can be matched to an event within a certain interval of time.

Possible values range is 1 to 100 times.

This value, also called match count target, is used in conjunction with the `in seconds` and `then status` fields to define what event rate is used to prevent the Rule from matching too often.

The Rule Set match rate is reset only when the Rule Set filter is updated or if the BBI-SS PAS is cold started.

in seconds Is the interval of time, set in seconds, over which the match count will be collected for an event.

Possible interval range is 1 to 99999 seconds.

When a Rule is matched the number of times set in the `If matched` field within the time interval set in the `in seconds` field, the action specified in the `then status` field is taken.

then status Specifies the mode to which a Rule is set when the number of times it is matched reaches the match count target within a specific time interval. Possible values are:

SUSPEND: Suspends the Rule.

When a Rule is in the SUSPEND state, events will no longer fire this Rule. In a Rule Set with a strategy of FIRST, the event will be eligible to fire a subsequent Rule in this Rule Set. When the match rate falls below the threshold, the Rule will be eligible to fire again.

DISABLE: Disables the Rule.

Event matches to the Rule are ignored when the actual match count exceeds the specified match count target, and the Rule is disabled. You must manually re-enable the Rule.

NOACTION: Takes no action.

When the match count for a Rule exceeds its match rate, the Rule is set for NOACTION. The Rule will match and the fired count will increase, but no actions specified for that Rule will take place. The action is resumed when the match rate falls below the threshold.

2. Press ENTER.

   Figure 106 is displayed.

```
BMC Software ------------- Variable Dependencies - Filter -------- AutoOPERATOR
COMMAND ===>                                                TGT  --- MVR3

       Rule-set === RULJ911
                                                                          OR/
 Variable-name                    Op  Variable-Value                     AND

 ________________________________ __  ______________________________________  ___
 ________________________________ __  ______________________________________  ___
 ________________________________ __  ______________________________________  ___
 ________________________________ __  ______________________________________  ___
 ________________________________ __  ______________________________________  ___
 ________________________________ __  ______________________________________  ___
 ________________________________ __  ______________________________________  ___
 ________________________________ __  ______________________________________  ___
 ________________________________ __  ______________________________________  ___
 ________________________________ __  ______________________________________  ___
 ________________________________ __  ______________________________________  ___
 ________________________________ __  ______________________________________  ___
 ________________________________ __  ______________________________________

   Press ENTER for Filter Criteria Panel,
   END to apply changes, CANCEL to cancel changes
```

Figure 106. Variable Dependencies - Filter Panel

You can use this panel to specify variables as additional filter criteria for the Rule Set. You can use any SHARED or LOCAL variables on this panel. Refer to “Using Variables in the Rule Processor” on page 161 for more information about using variable dependencies for selection criteria.

3. Press PF3 to return to the Automation Control Panel.

After you have specified the filter criteria for the Rule Set, you might want to examine the Rules within the Rule Set to ensure the Rules are adequate for automating the events.

To do this:

1. From the Automation Control panel, type `S` in the `LC` column next to the Rule Set (Figure 107).

```
BMC Software ----------------- Automation Control ---------------- AutoOPERATOR
COMMAND ===>                                                  TGT ===> MVR3
Primary commands: Add, Statshow, Cmdshow                      DATE --- 01/02/14
                                                              TIME --- 13:23:05
Automation Status     ===> ACTIVE         (Active, Inactive)
Automation Strategy   ===> INDIVIDUAL     (Individual, All, First, Qualified)
Honor MPF Suppression ===> NO             (NO/YES)

                        Automation Statistics
  Total Events                 624   Display suppressed              0
  Events Handled                49   Hardcopy suppressed             0
  Current arrival rate    3  / sec   Rule generated Alerts            6
  Peak arrival rate      10  / sec   Rule invoked Execs              0
                         Automation Library
LC CMDS --- (S)elect, (E)nable, (D)isable, (T)est, (SA)ve
            (M)ove, (B)efore or (A)fter, (F)ilter Criteria
LC   Rule-Set Status   Rules     Fired  Filtered    Date      Time   Strategy
___  RULQUAL1 ENABLED      6        46       624d 01-FEB-01 13:18:50 QUAL
___  RULJRNL1 ENABLED    108        93       183  01-FEB-01 13:18:49 FIRST
___  RULJRNL2 ENABLED     53         6         8  01-FEB-01 13:18:49 FIRST
___  RULJRNL3 ENABLED     59        46        32  01-FEB-01 13:18:49 ALL
___  RULJRNL4 ENABLED     56         9        17  01-FEB-01 13:18:49 FIRST
___  RULJRNL5 ENABLED     37       346        46  01-FEB-01 13:18:49 ALL
S__  RULJ911  ENABLED      6        46        46  01-FEB-01 13:18:49 FIRST
___  AAORULCC DISABLED   N/A   N/A       N/A          N/A    N/A
___  AAORULJB DISABLED   N/A   N/A       N/A          N/A    N/A
```

Figure 107. Automation Control Panel: Viewing a Rule Set's Rules

Figure 108 is displayed:

```
BMC Software ------------------ Rule Set Overview ---------------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- MVR3
Rule Set ID: RULJ911     Ruleset Strategy ===> FIRST          DATE --- 01/02/14
Primary commands: Add, Save, Sort, Unsort, Reset, Filter      TIME --- 13:23:24
LC CMDS --- (S)elect, (E)nable, (D)isable, (T)est, (DE)lete, (I)nsert
            (C)opy/(CC)opy, (M)ove/(MM)ove, (B)efore or (A)fter, (R)epeat
Sort Criterion:            Filter ENABLED                       right/left

LC  Rule-id Stat Text-id          Type   Fired EXEC        Changed      ID
    ________ _   _______________ ____        _________ ________ ______ ________
___ IM9113I  ENA IM9113I          JRNL      12          01/02/27 16:36 CMF16
___ IM9114I  ENA IM9114I          JRNL      13          01/02/21 15:28 CMF16
___ IM9117I  ENA IM9117I          JRNL       0          01/02/27 16:35 CMF16
___ IM9118I  ENA IM9118I          JRNL      17          01/02/05 08:12 CMF16
___ IM9119I  ENA IM9119I          JRNL       0          01/02/22 07:52 CMF16
___ ALLJRNL  ENA IM911*           JRNL       4          01/02/05 08:10 CMF16
      ************************* END OF DATA **************************
```

Figure 108. Rule Set Overview Panel: Viewing a Rule Set's Rules

The center of this panel shows that the filter is enabled. You also can see in the `Text-id` fields, the Rules all specify to match with events that begin with `IM911`. The `Type` column also shows that all the event types for these Rules are JRNL. There is even a “catch-all” Rule, Rule-ID ALLJRNL, that will be searched for all events that begin with IM911.

Press PF3 to exit this Rule Set. If you did not change any Rules, you will not be prompted to save the Rule Set. If you made changes, enter SAVE in the command line from the Rule Set Overview or enter SAVE when prompted while exiting the Rule Set.

Chapter 10. Using the Rule Set Overview Panel to Manage Rules in Rule Sets

Use the Rule Set Overview panel to display and manage Rules within a specific Rule Set. Rules are grouped into Rule Sets based on criteria such as:

- All are used for a single automation procedure (for example, CSM)
- All were created by the same department
- All address the same MVS component (for example, allocation)

Use the Rule Set Overview panel to add, delete, enable, disable, or modify individual Rules within a specific Rule Set. Rule Set Filtering also can be enabled or disabled from this panel. For more information about Rule Set filtering, refer to “Filtering Events for Rule Sets” on page 188.

Access the Rule Set Overview panel by selecting a Rule Set from the Automation Control panel. You can scroll this list of Rules either up/down or right/left. Figure 109 on page 197 shows the Rule Set Overview panel when you access it from the Automation Control panel.

```
BMC Software ------------------ Rule Set Overview ---------------- AutoOPERATOR
COMMAND ===>
Rule Set ID: AAORULST    Ruleset Strategy ===> FIRST         DATE --- 01/02/01
Primary commands: Add, Save, Sort, Unsort, Reset, Filter     TIME --- 14:21:30
LC CMDS --- (S)elect, (E)nable, (D)isable, (T)est, (DE)lete, (I)nsert
            (C)opy/(CC)opy, (M)ove/(MM)ove, (B)efore or (A)fter, (R)epeat
Sort Criterion:               Filter ENABLED                      right/left

LC  Rule-id Stat Text-id          Type   Fired EXEC        Changed      ID
    ________ _   _________________ ____         ________ ________ _____ ________
___ IS0834I  DIS IS0834I          CMD        0 AB2START 01/01/27 08:59 MXY1
___ IS0836I1 DIS IS0836I          JRNL       0 AB2TERM  01/02/02/13 16:53 DDH1
___ DFS9940  DIS DFS994I          IMS        0 IMSTART1 01/02/13 17:32 DDH1
___ IM0001I  DIS IM0001I          IMS        0 IMSTART1 01/02/13 17:32 DDH1
___ IMSCHCK  ENA /CHE             IMS        0 IMSTERM1 01/03/27 09:02 MXY1
___ RUL00001 ENA /CHE             JRNL       0 IMSTERM1 01/02/16 17:22 DDH1
___ IS0861I  ENA IS0861I          JRNL       1 AB2START 01/02/13 16:52 DDH1
___ IS0862I  ENA IS0862I          JRNL       0 AB2TERM  01/02/13 16:53 DDH1
___ IM9218I  ENA IM9218I          JRNL       0 IMSTART1 01/03/13 17:32 DDH1
         ************************* END OF DATA *************************
```

Figure 109. Rule Set Overview Panel—View 1

Scroll **left** to see the `Group`, `Function`, `Code`, `Author`, and `Description` columns as shown in Figure 110 on page 199.

The following describes the fields on this view of the panel.

Field	Description
Rule-id	Is the 1- to 8-character unique ID assigned to this Rule at its creation.
Stat	Is the Rule status, which can be one of the following: ENA, DIS, TST, or SUS. SUS status applies when the Rule is suspended through the use of the `If matched`, `in seconds`, and `then status` fields on the Rule Processor Detail Control panel. Refer to "Using the Criteria Match Rate Fields" on page 70.
Text-id	Is the 1- to 16-character first word of the message or command defined for this Rule.
Type	Is the origin of the event, which can be one of the following: MSG, CMD, JRNL, IMS, ALE, CICS, JES3, EXT, DB2, or TIME.
Fired	Is the number of times the Rule was triggered.
EXEC	Is the name of the EXEC that will be scheduled if the Rule fires. If the EXEC name specified in the Rule is a variable, its length may be more than 8 characters. Only the first 8 characters of the EXEC field in the Rule are presented in this field.
Changed	Is the date (YY/MM/DD) and time (HH:MM:SS) the Rule was last modified. This data is stored in the Rule.
ID	Is the user ID that made the most recent change to this Rule. This data is stored in the Rule.

Scrolling **left** shows the `Group`, `Function`, `Code`, `Author`, and `Description` fields (Figure 110).

```
BMC Software ------------------ Rule Set Overview ---------------- AutoOPERATOR
COMMAND ===>
Rule Set ID: AAORULST    Ruleset Strategy ===> FIRST         DATE --- 01/02/01
Primary commands: Add, Save, Sort, Unsort, Reset, Filter     TIME --- 14:21:30
LC CMDS --- (S)elect, (E)nable, (D)isable, (T)est, (DE)lete, (I)nsert
            (C)opy/(CC)opy, (M)ove/(MM)ove, (B)efore or (A)fter, (R)epeat
Sort Criterion:               Filter ENABLED                      right/left

LC  Rule-id Status    Group    Function Code Author   Description
    ________ _        ________ ________  __  ________ ___________________________
___ IS0834I  DIS      STRTTERM DB2START  DB  BOB      SCHEDULE DB2START EXEC
___ IS0836I1 DIS      STRTTERM DB2TERM   DB  BOB      SCHEDULE DB2TERM EXEC
___ DFS9940  DIS
___ IM0001I  DIS
___ IMSCHCK  ENA
___ RUL00001 ENA
___ IS0861I  ENA
___ IS0862I  ENA
___ IM9218I  ENA
         ************************* END OF DATA *************************
```

Figure 110. Rule Set Overview Panel—View 2

Scroll **right twice** from this panel to display the `Text-ID` and `Text-String` fields as shown in Figure 111.

The following describes the fields on this view of the panel.

Field	Description
Rule-id	Is the 1- to 8-character unique ID assigned to this Rule at its creation.
Stat	Is the Rule status, which can be one of the following: ENA, DIS, TST, or SUS. SUS status applies when the Rule is suspended through the use of the `If matched`, `in seconds`, and `then status` fields on the Rule Processor Detail Control panel. Refer to “Using the Criteria Match Rate Fields” on page 70.
Group	Is the 1- to 8-character group name defined for this Rule.
Function	Is the 1- to 8-character function name defined for this Rule.
Code	Is the 2-character code ID defined for this Rule.
Author	Is the 1- to 8-character name of author defined for this Rule.
Description	Is the description defined for this Rule. This can be up to 25 characters.

Scrolling **right twice** displays `Text-ID` and `Text-String` fields (Figure 111).

```
BMC Software ------------------ Rule Set Overview ---------------- AutoOPERATOR
COMMAND ===>
Rule Set ID: AAORULST    Ruleset Strategy ===> FIRST         DATE --- 01/02/01
Primary commands: Add, Save, Sort, Unsort, Reset, Filter     TIME --- 14:21:30
LC CMDS --- (S)elect, (E)nable, (D)isable, (T)est, (DE)lete, (I)nsert
            (C)opy/(CC)opy, (M)ove/(MM)ove, (B)efore or (A)fter, (R)epeat
Sort Criterion:               Filter ENABLED                      right/left

LC  Rule-id  Text-ID          Text-String
    _______ _______________ ______________________________________________
___ IS0834I
___ IS0836I1
___ DFS9940
___ IM0001I
___ IMSCHCK
___ RUL00001
___ IS0861I
___ IS0862I
___ IM9218I
         ************************* END OF DATA *************************
```

Figure 111. Rule Set Overview Panel—View 3

Scroll **right once** from this panel to see `Stat`, `Start Time`, `Interval`, `Stop Time`, `Stop Count`, and `Fired` fields as shown in Figure 112.

The following describes the fields on this view of the panel.

Field	Description
Rule-ID	Is the 1- to 8-character unique ID assigned to this Rule at its creation.
Text-ID	Is the 1- to 16-character first word of the message or command defined for this Rule.
Text-String	Is the text string defined for this Rule; can be up to 26 characters and includes the Text ID.

For information about Text-ID and Text-String, see page 191.

Scrolling **right once** displays `Stat`, `Start Time`, `Interval`, `Stop Time`, `Stop Count`, and `Fired` fields (Figure 112).

```
BMC Software ------------------ Rule Set Overview ---------------- AutoOPERATOR
COMMAND ===>                                                 TGT  --- DH22
Rule Set ID: AAORULST    Ruleset Strategy ===> FIRST         DATE --- 01/02/01
Primary commands: Add, Save, Sort, Unsort, Reset, Filter     TIME --- 14:21:30
LC CMDS --- (S)elect, (E)nable, (D)isable, (T)est, (DE)lete, (I)nsert
            (C)opy/(CC)opy, (M)ove/(MM)ove, (B)efore or (A)fter, (R)epeat
Sort Criterion:               Filter ENABLED                     right/left
                     Start               Stop      Stop
LC  Rule-id  Stat    Time      Interval  Time      Count      Fired
    ________ _       ________  ________  ________  _________
___ IS0834I  DIS                                                 0
___ IS0836I1 DIS                                                 0
___ DFS9940  DIS                                                 0
___ IM0001I  DIS                                                 0
___ IMSCHCK  ENA                                                 0
___ RUL00001 ENA                                                 0
___ IS0861I  ENA                                                 1
___ IS0862I  ENA                                                 0
___ IM9218I  ENA                                                 0
         ************************* END OF DATA *************************
```

Figure 112. Rule Set Overview Panel—View 4

Scroll **right once** from this panel to see `JOB NAME`, `JOB TYPE`, `JOB CLASS`, `RACF USERID`, and `RACF GROUP` fields as shown in Figure 113.

The following describes the fields on this view of the panel.

Field	Description
Rule-id	Is the 1- to 8-character unique ID assigned to this Rule at its creation.
Stat	Is the Rule status, which can be one of the following: ENA, DIS, TST, or SUS. SUS status applies when the Rule is suspended through the use of the `If matched`, `in seconds`, and `then status` fields on the Rule Processor Detail Control panel. Refer to “Using the Criteria Match Rate Fields” on page 70.
Start Time	Displays (in hours, minutes, and seconds) at what clock time a time-initiated Rule begins firing. This field contains data only for Rules whose Event type is TIME.
Interval	Displays (in hours, minutes, and seconds) how often the Rule fires. For example, an interval of 00:05:00 means the Rule will fire at the Start Time and every 5 minutes thereafter until the Stop Time or Stop Count value is reached. The minimum interval is 5 seconds.
Stop Time	Displays (in hours, minutes, and seconds) at what clock time the Rule should stop firing.

Stop Count Specifies the number of times for the Rule to be fired from the Start Time.

For example, a Stop Count value of 12 means that the Rule will begin firing at the specified Start Time and continue firing (at a rate determined by the Interval field) 12 times.

Fired Shows how many times the Rule was fired.

Scrolling **right once** displays `JOB NAME`, `JOB TYPE`, `JOB CLASS`, `RACF USERID`, and `RACF GROUP` fields (Figure 113).

```
BMC Software ------------------ Rule Set Overview ---------------- AutoOPERATOR
COMMAND ===>
Rule Set ID: AAORULST    Ruleset Strategy ===> FIRST         DATE --- 01/02/01
Primary commands: Add, Save, Sort, Unsort, Reset, Filter     TIME --- 14:21:30
LC CMDS --- (S)elect, (E)nable, (D)isable, (T)est, (DE)lete, (I)nsert
            (C)opy/(CC)opy, (M)ove/(MM)ove, (B)efore or (A)fter, (R)epeat
Sort Criterion:               Filter ENABLED                      right/left
             Job      Job  Job   RACF     RACF
LC  Rule-id  Name     Type Class User-ID  Group
    ________ ________ _    _     ________ ________
___ IS0834I
___ IS0836I1
___ DFS9940
___ IM0001I
___ IMSCHCK
___ RUL00001
___ IS0861I
___ IS0862I
___ IM9218I
         ************************* END OF DATA *************************
```

Figure 113. Rule Set Overview Panel—View 5

The following describes the fields on this view of the panel.

Field	Description
Rule-id	Is the 1- to 8-character unique ID assigned to this Rule at its creation.
Job Name	Is the name of the originating batch job, TSO user ID, or started task defined for this Rule. Blank means any.
Job Type	Is the type of originator defined for this Rule: JOB, STC, or TSO.
Job Class	Is the job class of the job defined for this Rule.
RACF User-ID	Is the user ID defined for this Rule, if RACF or CA-TOPSECRET or ACF2 is installed.
RACF GROUP	Is the name of the RACF Group defined for this Rule.

Using Primary Commands

You can enter the following primary commands in the COMMAND field of the Rule Set Overview panel.

Command	Parameters	Description
ADD		Displays the Rules Processor Detail Control panel.
SORT		Sorts the display in either ascending or descending order using the field name as the sort field; for example: `SORT FIRED D` sorts the display by Fired count in descending order, thus the Rules that fired most frequently will appear at the top of the display. You can also sort and filter the display based on: • Rule-id • Stat • Text-ID • Type • EXEC • ID The default is Ascending|A order for all these fields, except Fired and Changed which default to Descending|D.
UNSORT		Undoes the Sort command and returns the display to its original order; by physical location of the Rule within the Rule Set. This command must be issued before a Move|M, Copy|C, Repeat|R, or Insert| can be issued.
RESET		Allows you to cancel any pending move or copy line commands that you may have.
Filter		Allows you to invoke the Rule Set filter panels to specify filtering selection criteria for the Rule Set. For more information about Rule Set filtering, refer to“Filtering Events for Rule Sets” on page 188.

Using Line Commands

You can use the following line commands in the LC column.

Command	Description
Select|S	Transfers you directly into the Rule Processor Detail Control panel.
Enable|E	Activates a Rule and effectively starts the selection criteria filtering process.
Disable|D	Stops the selection criteria filtering process for this Rule.

Test|T Sets the Rule to test mode. Use the Enable or Disable command to get the Rule out of test mode.

Using test mode allows you to see if the Rule fires as expected but the actions specified for the Rule are not taken.

DElete|DE Deletes a Rule.

Insert|I Transfers directly into the Rule Processor Detail Control panel to create a new Rule before or after the line against which this command is issued.

Copy|C|CC Copies Rules to another location in the Rule Set or to another Rule Set.

The Rules can be enabled or disabled. For more information, refer to "Copying or Moving Rules" on page 205.

Move|M|MM Moves Rules to another location in the Rule Set or to another Rule Set.

The Rules can be enabled or disabled. For more information, refer to "Copying or Moving Rules" on page 205.

Before|B Indicates the line within the Rule List before which a Rule is copied or moved with the Move|M|MM or Copy|C|CC line commands.

For more information, refer to "Copying or Moving Rules" on page 205.

After|A Indicates the line within the Rule List after which a Rule is copied or moved with the Move|M|MM or Copy|C|CC line commands.

For more information, refer to "Copying or Moving Rules" on page 205.

Repeat|R Creates a Rule using the active pattern of that line. Use this command to create Rules that need only minor modifications rather than re-entering the information for each Rule.

Copying or Moving Rules

Use the Copy and Move commands in the `LC` column for moving/copying single Rules or blocks of Rules:

- To move or copy single Rules, place the M or C line command next to the Rule in the `LC` column.
- To move or copy blocks of Rules, place MM or CC line command next to the first and last Rules of a group of Rules.

The move and copy commands must be used in conjunction with the After|A and Before|B line commands.

You can copy or move Rules either within the same Rule Set or to another Rule Set that is **on the same target**. The Rules can be disabled when you copy or move them.

Copying or Moving Rules within Rule Set

To use the copy or move commands within the same Rule Set:

1. From the Automation Control Panel, select an ENABLED Rule Set.
2. From the Rule Set Overview panel, select a Rule or a block of Rules to copy or move.
3. Place the Copy|C|CC or Move|M|MM commands next to the Rule(s) in the `LC` column.
4. Enter either the After|A or Before|B line commands depending on where you want to copy or move the Rule(s) to in the Rule Set.
5. Press ENTER.

   The Rules are copied/moved to the new location.
6. Press PF3/END.

   The Confirm Rule Set Modifications panel is displayed.
7. To save the copied or moved Rules, enter SAVE.

   The Rules are copied or moved.

Copying or Moving Rules between Rule Sets

To use the copy or move commands between Rule Sets:

1. From the Automation Control Panel, select an ENABLED Rule Set.
2. From the Rule Set Overview panel, select a Rule or a block of Rules to copy or move.
3. Place the Copy|C|CC or Move|M|MM commands next to the Rule(s) in the `LC` column.
4. Press PF3/End to return to the Automation Control panel.
5. Select the Rule Set (on the same target) to copy or move the Rules to.

6. On the Rule Set Overview panel, enter the After|A or Before|B line commands depending on where you want to copy/move the Rule(s) from the first Rule Set to this Rule Set.

   This Rule Set must have at least one Rule already in it where you can enter the After|A or Before|B line commands.

7. Press ENTER.

   The Rules are copied/moved to the new location.

8. Press PF3/END.

   The Confirm Rule Set Modifications panel is displayed:

   – If you are copying Rules, entering SAVE saves the Rules to the copied to Rule Set and leaves a copy in the copied from Rule Set.

   – If you are moving Rules, entering SAVE saves the Rules to the moved to Rule Set and removes them from the moved from Rule Set.

9. To save the copied or moved Rules, enter SAVE.

   The Rules are copied or moved.

If you choose to cancel any pending moves or copies, use the Reset primary command.

Part 3. Managing Critical Applications Using CSM

This part contains documentation for using CSM to manage critical processes such as automating IPLs and shutdowns.

The discussion is organized into the following areas:

- CSM Terms and Concepts
- Creating CSM Objects and Creating CSM Relationships
- Displaying CSM Object States
- Describing CSM Panels

Chapter 11. Managing Critical Applications with Continuous State Manager 211
What Continuous State Manager Is 212
What the Benefits of CSM Are. 213
What CSM Requires 213
Defining CSM Terms and Concepts 214
CSM Terms and Concepts. 214
What an Object Is 214
What Object Relationships Are. 215
What an Object State Is. 218
What a Schedule Is 220
What Else CSM Must Know and Why. 222
How CSM Starts, Stops, and Cancels Objects 222
How CSM Recognizes when an Object Starts or Stops 222
How CSM Connects and Disconnects Objects 223
How CSM Works 224
When CSM Starts 224
Routine Scheduling 225
Recovering Failed Objects 225
Interfering with CSM Processing 226
Managing Objects Across BBI-SS PASs 227
Examples of CSM Processing 229
How CSM Automates the IPL Process 230
How CSM Automates the Shutdown Process 232
How CSM Manages Routine Object Scheduling 235
How CSM Handles Global Calendar Overrides 237
How to Effectively Plan for CSM Implementation 239
Determining Object Relationships 240
Planning Schedules 240
Listing Start, Stop, Cancel and (Dis)Connect Commands 241
Choosing Start and Stop Text-IDs. 241
Choosing Text-IDs for Abnormal Termination Events 241
Before You Begin 243
Setting Up the Shared Repository 243
Cross-System Object Management: Using Group 243
Repository Serialization 244
Securing Access to CSM. 244
Backing Up the CSM Repository 244
Using CSM Panels. 245
Accessing CSM for the First Time 247

How to Qualify (Mask) Panels in CSM 248
What the CSM Global Overview Panel Is 249
Viewing an Object's Documentation 249
Viewing Objects Whose States Do Not Match 250
Filtering Groups Displayed on Global Overview 251
Defining CSM Objects 255
Creating a Group 256
Entering the Object's Information 259
Entering Command Specifications for an Object 262
Additional Fields to Complete 264
Specifying Conditional Command Specifications for an Object 265
Specifying Start and Stop Events: Event Specifications 1 267
Specifying Abnormal Termination Events: Event Specifications 2 268
Specifying User Notification 270
Associating an Object with a Group 272
Moving an Object Between Groups 273
Defining the Object's Parents 274
Defining the Object's Servers 275
Defining a Transient Object 278
Defining a Grouping Object 278
Comparing CSM Object Types 278
Using the Repeat Line Command to Create Objects 279
Modifying Object Definitions 282
Adding Parents 282
Deleting Parents 286
Adding Servers 288
Deleting Servers 291
Modifying an Object's Routine Schedule 294
Determining the Schedule of the Object's Parent 294
Modifying the Object's Schedule 296
Modifying an Object's Schedule with CSM Schedule Tags 299
Dealing with Scheduling Conflicts at Modification 304
Overriding the Calendar for Special Situations 306
Enabling Cross-System Object Management 308
Routine Object Tasks 310
Starting an Object 311
Stopping an Object 314
Bouncing an Object 316
Resetting an Object 318
Canceling an Object 321
Moving an Object: Using the Moveable Attribute 323
Scheduling an EXEC Associated with a CSM ALERT 326
Displaying and Modifying Object States 327
What the Possible Desired States Are 327
What the Possible Actual States Are 330
Monitoring CSM Object States Manually 330
Using the Command Line Interface 335
Describing CSM Panels 338
CSM Global Overview Panel 338
Defining and Modifying Object Panels 349
Object Detail Control Panel 349
Command Specifications Panel 353
Conditional Command Specifications Panel 359
Event Specifications 1 Panel 361

Event Specifications 2 Panel 365
User Notification Panel 367
Object Groups Panel 369
Object Requisites Panel. 371
Client/Server Relationships Panel 373
Object Scheduling Panel 376
Global Calendar Override Panel 379
Object Groups Panel 383
Group Detail Control. 385

Chapter 11. Managing Critical Applications with Continuous State Manager

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

To Read About...	See page...
“What Continuous State Manager Is”	212
“Defining CSM Terms and Concepts”	214
“How CSM Works”	224
“How to Effectively Plan for CSM Implementation”	239
“Before You Begin”	243
“What the CSM Global Overview Panel Is”	249
“Defining CSM Objects”	255
“Routine Object Tasks”	310
“Displaying and Modifying Object States”	327
“Using the Command Line Interface”	335
“Describing CSM Panels”	338

What Continuous State Manager Is

The Continuous State Manager application is a Rule-driven process that minimizes operator intervention by automatically managing MVS IPL and shutdown, and object startup, shutdown, recovery, and notification. Initialization and shutdown of a system can be a time-consuming and error-prone process if performed manually. Having the application control these processes saves time, eliminates errors, and minimizes manual intervention.

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 214
- Relationships between objects, page 215
- States of objects, page 218
- Schedules, page 220

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 allowing the application to automatically manage your system in an ordered way.

States

To refer to an object as UP 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 am to 3:00 am 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 allowing important processes such as IPL and orderly shutdowns 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 allows you to do this. The section "Using the CSMACT EXEC to Manually Modify Object States" on page 331 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 will operate more smoothly, with fewer errors.

- Reducing IPL time up to 50%

  One of the key processes that CSM performs is MVS IPL and this 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 will save time.

- Restarting failed objects (if needed)

  Once they are properly defined to CSM, all the started tasks can be controlled and managed by CSM. This includes 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 automatically notified about objects that have changes of states outside of CSM's control or about objects that fail to start or stop correctly.

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

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

Defining CSM Terms and Concepts

Important terms and concepts used in CSM are described in:

- "What an Object Is" on page 214.
- "What Object Relationships Are" on page 215
- "What an Object State Is" on page 218
- "What a Schedule Is" on page 220
- "What Else CSM Must Know and Why" on page 222
- "Using CSM Panels" on page 245

CSM Terms and Concepts

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

Term	Definition
CSM-PLEX	A group of BBI-SS PAS, 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 used in previous releases of 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 MVS 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 allows you to define CSM objects and use them and their schedules across all the CSM-PLEX.

What an Object Is

You can define three different types of objects in CSM:

- Normal
- Transient
- Grouping

Normal Objects

A normal object is an MVS 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 259.

Transient Objects

A transient object is an MVS 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 may 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 will never attempt to change the object's state. For information about how to define transient objects, refer to "Defining a Transient Object" on page 278.

Grouping Objects

A grouping object is a user-defined imaginary object that allows you to group other objects together. This is a very important function 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 216). By stopping object CICSPROD, the children CICS1, CICS2, and CICS3 automatically stop 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 this 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 278.

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 two 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 two types of relationships.

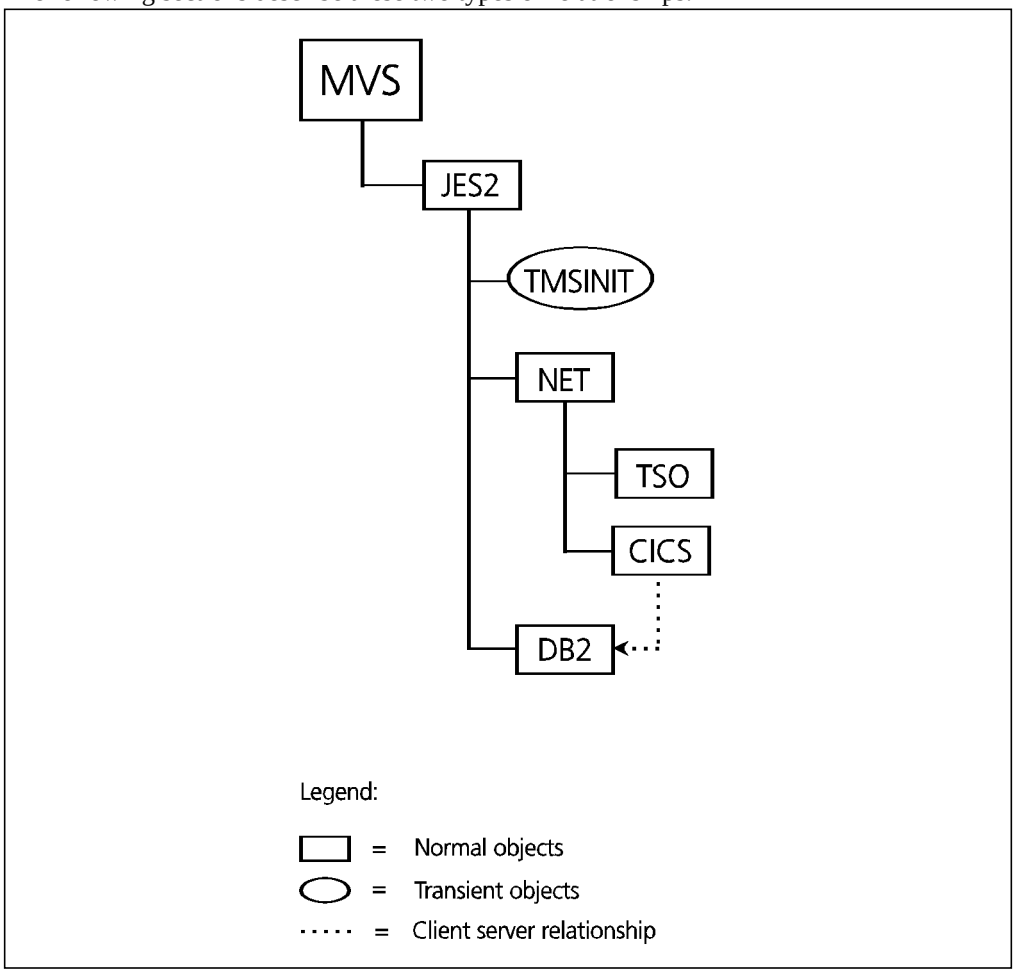

Figure 114. CSM Relationships

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 114 on page 216.

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 MVS, MVS 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 MVS 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 274.

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 114, 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 275.

The following table describes all the CSM object relationships in Figure 114.

Object	Object Type	Parent	Child	Client	Server
MVS	NORMAL	T Parent of JES2			
JES2	NORMAL	T Parent of NET, TMSINIT, and DB2	T Direct child of MVS		
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		

Note: Because MVS must be active for any other object to be active, MVS is by definition, the parent of all objects in the system. However, you do not need to explicitly define MVS as every object’s parent. This table shows only the direct relationship of every object.

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

What an Object State Is

There are two categories of states 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, then 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 218.

- Object actual states

  An object's actual state is the current status of the object and it may 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 actual states between UP and DOWN (see "Actual (Intermediate) States" on page 219).

For information about determining an object's state, refer to "Displaying and Modifying Object States" on page 327.

Object Desired States

Every time there is an event affecting 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 and/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
- Allows 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 327.

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, there is probably some problem with your object definitions. The problem is usually in the defined commands or message IDs 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 this object is in a 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 249 for more information.

What a Schedule Is

From “What an Object Is” on page 214, 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 is accomplished by creating a schedule as part of the object’s definition.

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

There are two types of object schedules: 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 an object is to be brought DOWN. Following is an example of a schedule for when an object should be DOWN:

```
Wednesdays from 3:00 pm to 7:00 pm
Fridays from 2:00 am to 3:00 am
Fridays from 6:00 am to 9:00 am
Sundays from 3:00 pm to 6:00 pm
```

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

When you define objects to CSM, you should plan and compare the schedule the object needs with the schedule of any children objects. This 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 parent(s). If it does not, a message prompts you during the definition of the object to correct the situation.

You also can 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 Mondays through Fridays 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 294.

Exceptional Scheduling for Objects

CSM allows 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 may discover that for a particular day an object that is scheduled to be DOWN at 4:00 pm may need to stay UP for an additional hour. You can schedule a Global Calendar Override for the object to stay UP until 5:00 pm 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. This 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 306.

What Else CSM Must Know and Why

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

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

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

- What are the messages that indicate to CSM that an object is UP or DOWN

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

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

  CSM uses the connect or disconnect commands 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 MVS 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, shutdown, or event affects an object.

You can specify either the MVS command that you might use to start, stop, or cancel the object from an MVS console or you can specify 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 MVS 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 signaling an object either starting or stopping. These Rules are called Start or Stop Rules; in turn, they schedule CSM EXECs which change the actual states of the object. These Rules reside in a CSM Rule Set specified in the object group's definition which 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 may find the CSM-generated Start and Stop Rules for an object do not meet all of your needs. If this 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 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 518 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 224. For information about defining objects, start, stop, cancel commands, and text-IDs, refer to "Defining CSM Objects" on page 255.

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 MVS console. For more information about defining client-server relationships, refer to "Defining the Object's Servers" on page 275.

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 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 am Monday, CSM reads the database and finds all the objects that are, according to their schedules, supposed to be UP at 9:00 am 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:

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

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

5. 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 am Monday, a CICS object is scheduled to be brought UP. When CSM initializes at 9:00 am, a time-initiated EXEC is scheduled to issue the start command defined for the CICS object at 10:00 am.

Recovering Failed Objects

There are two situations where an object may fail in CSM:

- 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. This means 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 does this 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" on page 225.

Specifying Start Command Limits

You can define the number of times CSM will attempt 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 re-start 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
- By using the RESET line command (and answering Y (for yes) on the last prompt) on the Global Overview panel
- By issuing the command `%CSMACT object RESETC` on the COMMAND line and starting the object from the panel
- By scheduling 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 an 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 an 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. This 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 may decide to (or have to) start, stop, or cancel an object at the last minute. You can do this 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 335.

If you use any of these methods, CSM will still be 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 line commands:

- `(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 manage objects from the panel.

Refer to “Routine Object Tasks” on page 310 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 allows 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 allows 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 wish 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 308.

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 MVS console
- Changing the state of an object from an EXEC
- Changing the state of an object from any AutoOPERATOR panel

For more information about using CSMACT, refer to “Using the CSMACT EXEC to Manually Modify Object States” on page 331.

Examples of CSM Processing

The following sections apply to the terms and concepts discussed up to this point in example of CSM processing; for example:

- “How CSM Automates the IPL Process” on page 230

  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 232

  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 235

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

- “How CSM Handles Global Calendar Overrides” on page 237

  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 224, 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 automatically attempts to start all the direct children of an object once the parent object is UP. This is because 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 MVS) so CSM tries to start all the objects that are defined in its database.

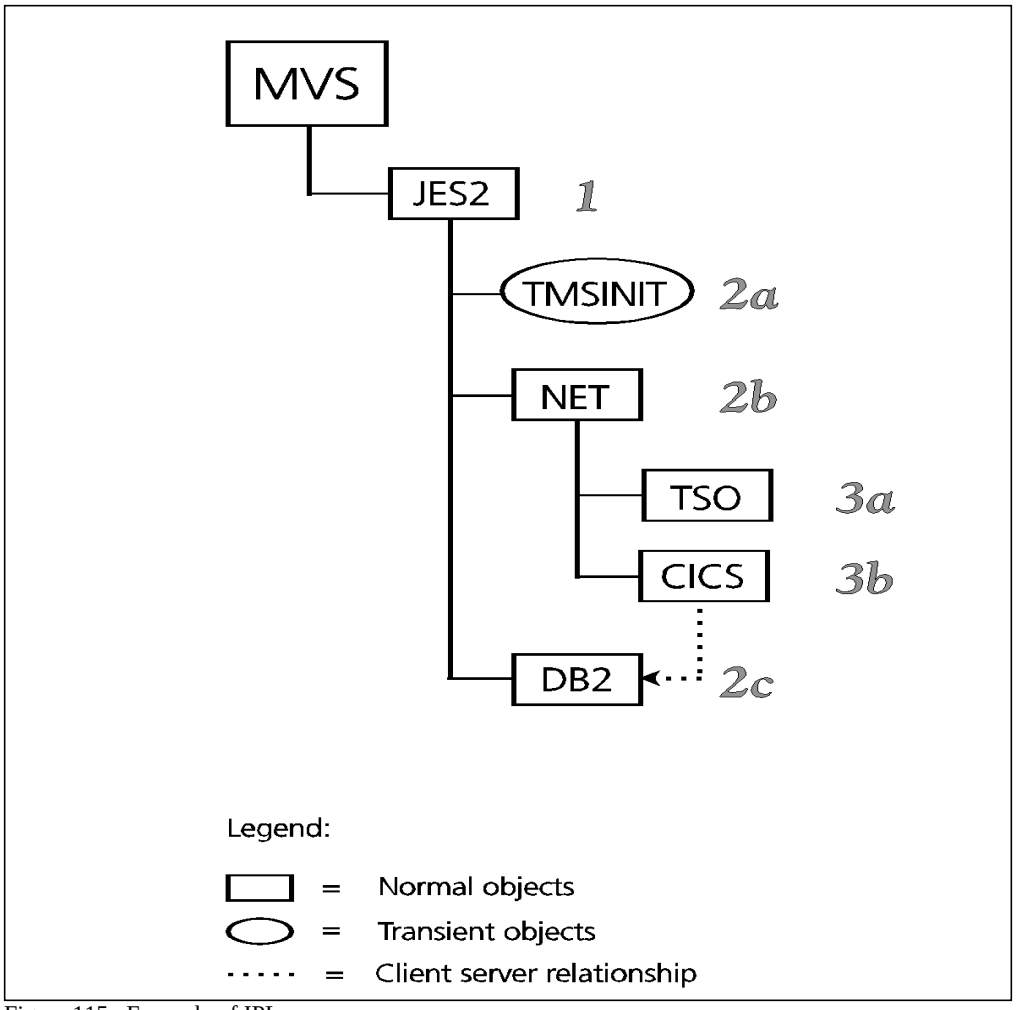

Figure 115. Example of IPL
The numbers (next to the objects) denote the order in which CSM attempts to start the objects.

During IPL: Suppose that the simplified system CSM is performing an IPL on is shown in Figure 115 on page 230 and all these objects are scheduled to be UP when CSM initializes. In this figure, MVS 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 MVS) 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 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 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 can always try 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 may have a scheduled down time every other Wednesday from 12:00 noon to 1:00 pm. CSM automatically performs scheduled processes at the specified times (see "How CSM Manages Routine Object Scheduling" on page 235).

- 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 225).

- An operator requests an action

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

- An Administration task is performed

  For example, someone with access to Administration functions in CSM might create or change a schedule for an object. Or someone 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 since CSM will automatically proceed to start objects immediately after CSM recognizes that their parent objects are UP. Another benefit is that CSM automatically notifies you with an ALERT if an object fails to start.

How CSM Automates the Shutdown Process

Using CSM for system shutdowns allows 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 you want to stop.

SHUTSYS: From the command line on the CSM dialog 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 MVS parent.

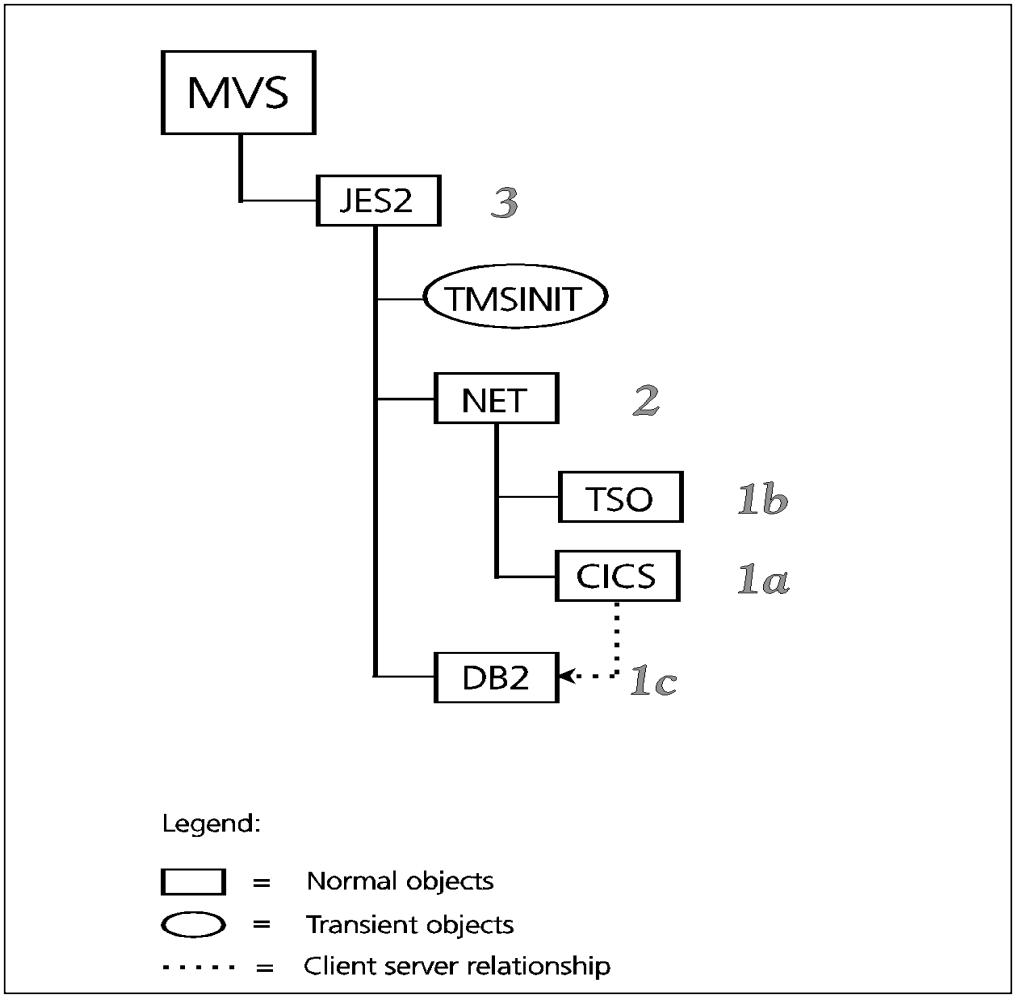

Figure 116. Example of Shutdown
The numbers (next to the objects) denote the order in which CSM attempts to stop the objects.

During shutdown: In Figure 116 on page 232, when the **SHUTSYS** command is issued against a group of objects, CSM finds the child that is both furthest away from MVS 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. This is because TMSINIT is a transient object and CSM only starts transient objects: CSM does not issue stop commands for transient objects.

TMSINIT may have already stopped when the shutdown began. If not, when CSM completes the shutdown, TMSINIT may 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 there are any objects in the system that were not defined to CSM, they may 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 117 on page 234 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.

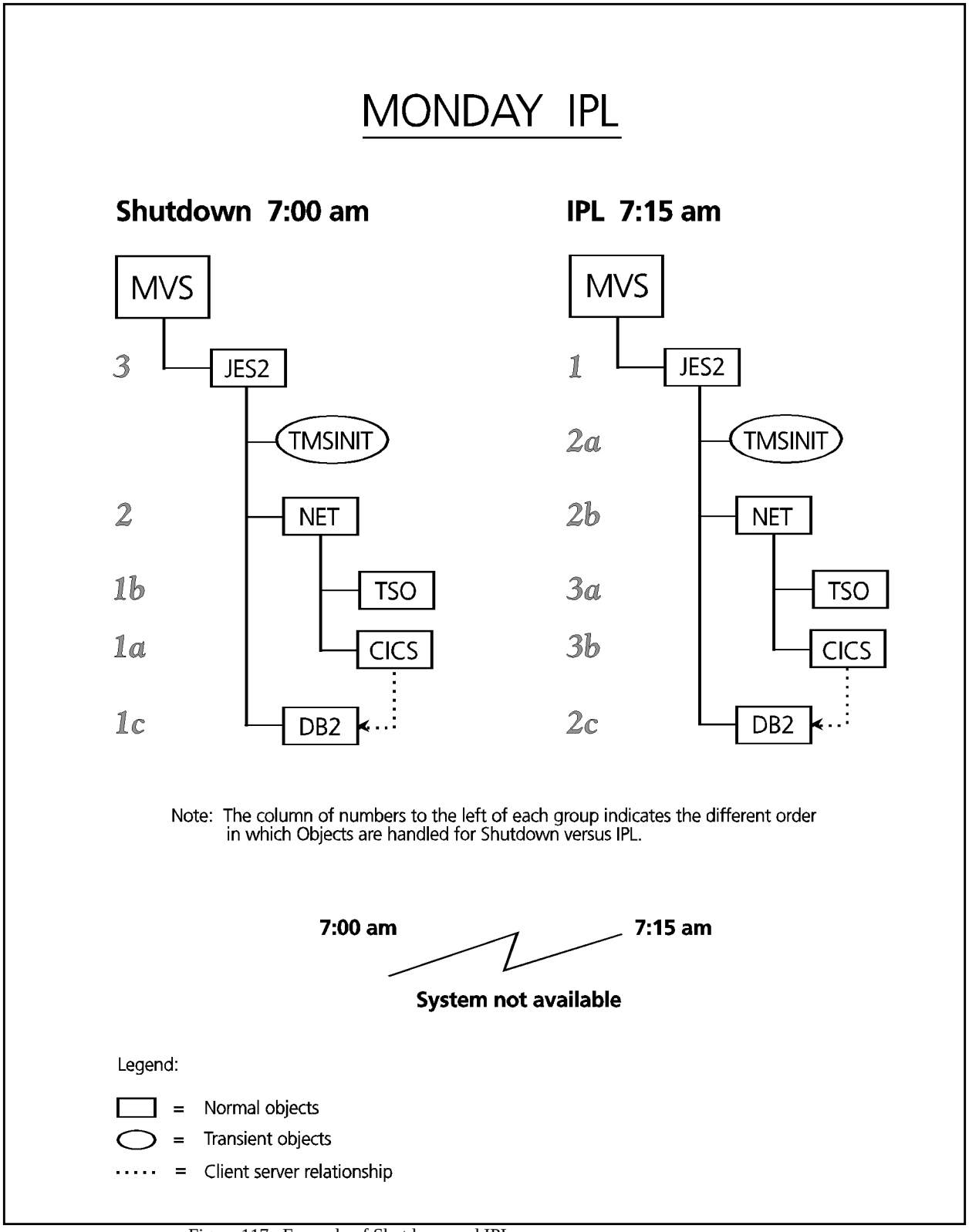

Figure 117. Example of Shutdown and IPL

How CSM Manages Routine Object Scheduling

As described in "Routine Scheduling" on page 225, during IPL, CSM schedules time-initiated EXECs to start and stop objects.

Scheduling to start objects: For example, suppose IPL occurs at 9:00 am and an object is scheduled to be started at 10:00 am. Prior to 10:00 am, the desired and actual states are DOWN. At 10:00 am, 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 pm, 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).

For example: Figure 118 on page 236 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 pm to 1:30 pm. 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 may not actually be DOWN until 1:10 pm.

At 1:30 pm, 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.

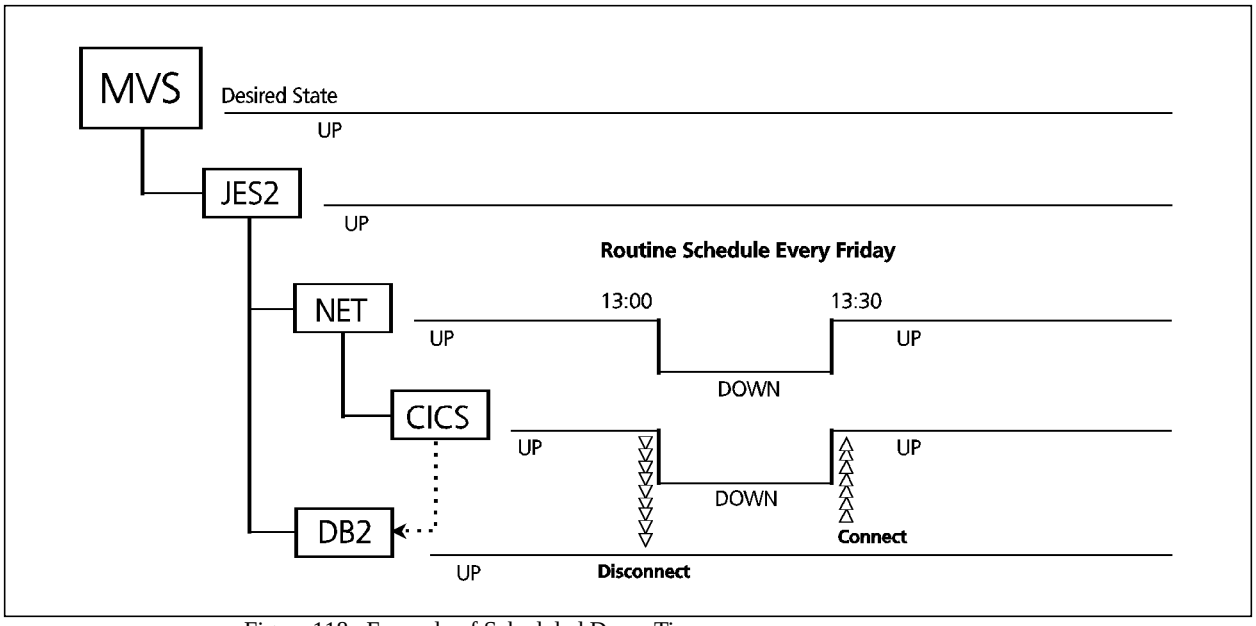

Figure 118. Example of Scheduled Down Time

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: 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 pm to 1:30 pm. This is when its parent, NET, is DOWN (because the child object cannot operate without the parent object).

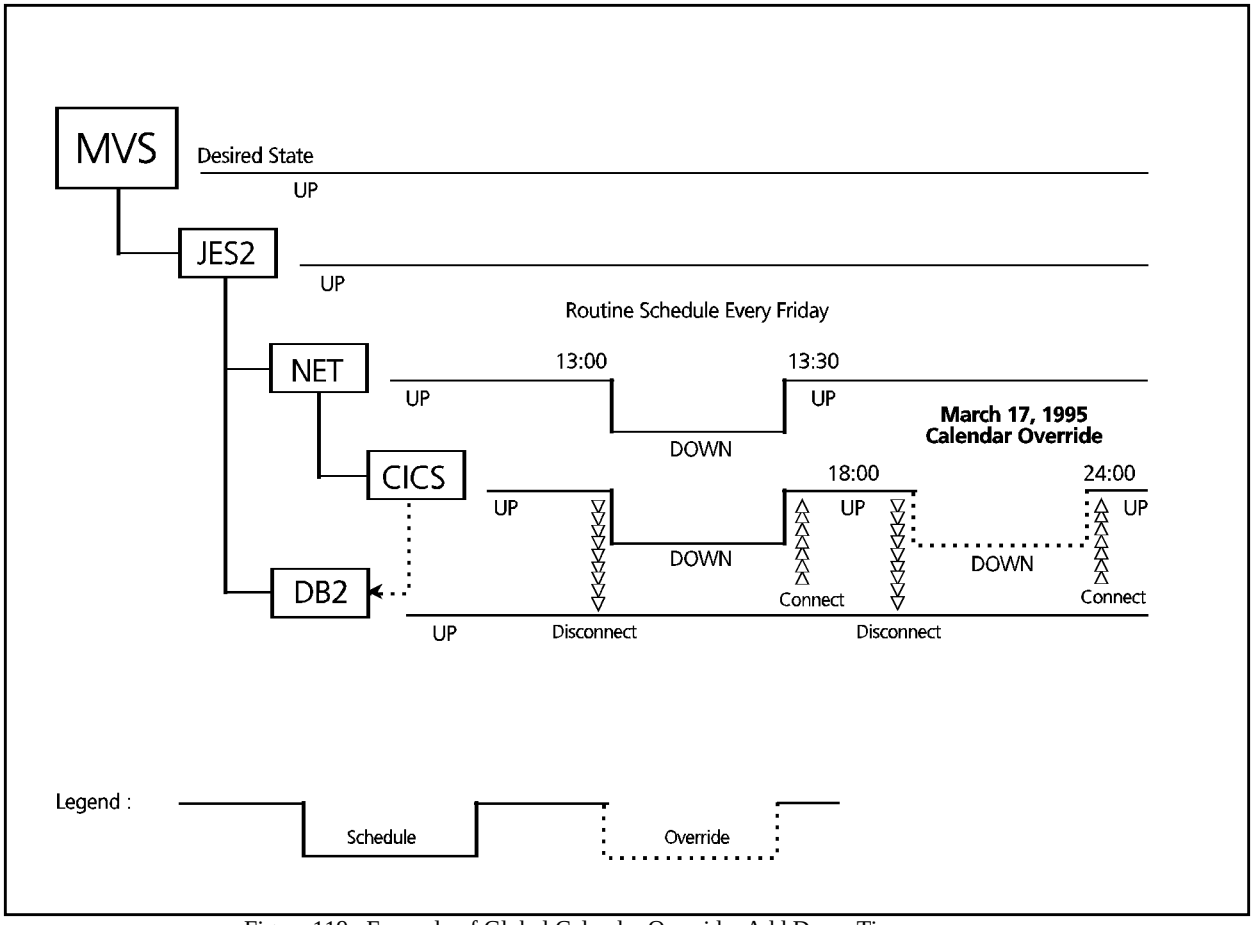

Figure 119. Example of Global Calendar Override: Add Down Time

This example also shows CICS has a Global Calendar Override scheduled that brings CICS DOWN from 6:00 pm to midnight in addition to its routine down time. At 6:00 pm, 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 120 shows an instance where a Global Calendar Override is scheduled to bring NET DOWN an hour earlier than the routine schedule specifies.

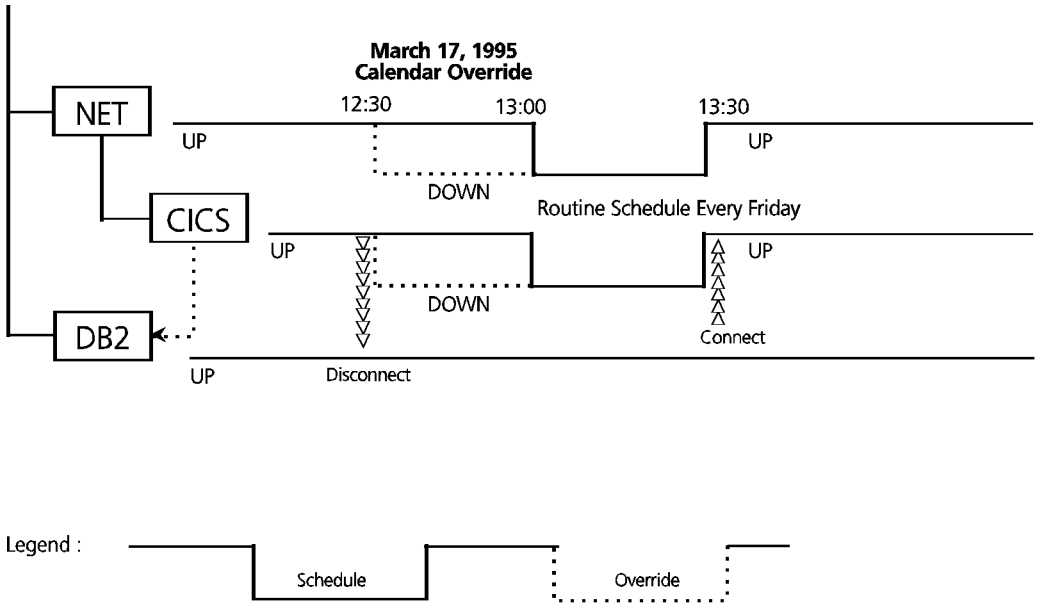

Figure 120. Example of Global Calendar Override: Extend Down Time

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 also will be 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 an AutoOPERATOR ALERT is issued if the object terminates unexpectedly (abends). In addition, CSM can automatically issue an EXEC or command. To do this 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 what 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 also may 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 114 on page 216.

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

Follow MVS with the objects as they are started manually by Operations during an IPL. Pay special attention to points in the process where the start-up of one object is the prerequisite for the start-up of one or more other objects. This 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 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. You will find that most objects fall into this category.

Some objects, usually on-line 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 MVS, 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 296 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 304 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 MVS commands 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 will find the commands.

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

Choosing Start and Stop Text-IDs

As described in "How CSM Recognizes when an Object Starts or Stops" on page 222, the text-IDs issued when an object has started or stopped is an integral part of an object's definition because 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 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.

Finding 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.

Hint
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 an abnormal termination event occurs and an AutoOPERATOR ALERT is issued if the object terminates unexpectedly (abends). In addition, CSM can automatically issue an EXEC or command. To do this 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 may 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 allow greater control over which abnormal termination event is handled and which is not. You must, however, specify the number of the WORD variable 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 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 local CSM’s active group. Cross-system management allows 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 CSMALTDB parameter in the MVS 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.

Repository Serialization

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 more than one person can make updates at a time but you cannot update another record at the same time as another.

Object entries are serialized on a 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 will 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, a user issues the EDIT line command. If CSM can get the exclusive enqueue for the group record, the user can edit that group. When that user is finished editing the group definition, and presses END, the group definition is immediately written to the repository and the enqueue is released. If another user tries to edit the group while the first user is doing so, they receive an 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 automatically create backups of the shared repository. You can run a simple jobstream that allocates a backup repository data set and copy the current one into the backup one. The JCL might look something like this:

```
//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)
/*
```

Figure 121. Sample JCL to Back Up CSM Shared Repository

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 AutoOPERATOR to schedule the backup job at a particular time each day.

Using CSM Panels

The CSM panels that allow 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:

1. Object Detail Control
2. Command Specification
3. Conditional Commands
4. Event Specification 1
5. Event Specification 2
6. User Notification
7. Object Groups
8. Object Prerequisites
9. Object Scheduling
10. Client/Server Relationships

Figure 122 shows an example of the Object Detail Control panel.

```
BMC Software ------------ Object Detail Control - Add ------------ 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
```

Figure 122. Object Detail Control Panel

As noted on the panel, CSM allows you to display any of the panels in the ring using shortcut commands entered on the COMMAND line. The commands are listed in the following table.

Table 20. Shortcut Commands for Navigating CSM Panels

Command	Displays the 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 will process any changes made to the object definition and will return you to Global Overview; pressing END from any other panel in the ring will return you to the Detail Control Panel.

Accessing CSM for the First Time

Subject:

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 BBV-based terminal session. CSM initialization begins after 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 AutoOPERATOR migrated to this version of CSM, the first thing you will begin doing is creating a group and then defining objects.

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

To access the CSM application, select Option 3, Continuous State Manager, from the Automation Menu (Figure 123).

```
BMC Software ---------------- AUTOMATION MENU -------------------- 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 1999-2000, BMC Software, Inc. All rights reserved.
```

Figure 123. AutoOPERATOR Automation Menu

The Global Overview panel is displayed (Figure 124).

```
BMC Software ------------------ Global Overview ------------------ 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                                                         NORM
___ 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
```

Figure 124. Global Overview Panel

For more information about the Global Overview panel, refer to “What the CSM Global Overview Panel Is” on page 249.

The first time you access the Global Overview panel, the only object that will appear will be the MVS object. The example in Figure 124 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 may begin defining objects. For information about creating a group, refer to “Creating a Group” on page 256.

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 will qualify (or mask) the data displayed to show specific information. For example, if you enter a specific SSID in the `SYSTEM` column and press ENTER, the panel will show 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

Subject: The first time you access the Global Overview panel in CSM, the only object that will appear will be the MVS object. The example in Figure 124 on page 248 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 the object is associated with
- The MVS 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 126 on page 250).

Viewing an Object's Documentation

To view additional information about each object, scroll left by pressing PF10/PF22 (see Figure 125).

```
BMC Software ------------ Global Overview - Desc/Stats ----------- 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                     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
___ 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
```

Figure 125. Global Overview Panel: Scrolled LEFT

This following information for each object is available for viewing:

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

Note: If an object is inactive (because the group that is belongs to is not enabled), the `IND` column will show no information.

To return to the original panel, press PF11/PF23. Figure 124 will redisplay.

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 automatically display the CSM Exceptions panel (Figure 126).

```
BMC Software ------------------- CSM Exceptions ------------------ 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 ******************************
```

Figure 126. CSM Exceptions Panel

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

The CSM Exceptions panel shows a list of those objects where:

- Any object's Actual and Desired states do not match

  In Figure 126, the Actual and Desired states for object AAOCSM65 do not match.

- Objects have at least one outstanding ALERT

  This is shown in the `IND` column. In this figure, `XE` denotes that there is an ALERT 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 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 may be difficult to view on the Global Overview panel. Refreshing the information might also prove to be time consuming.

The following figure shows an example when a large number of systems are monitored:

```
BMC Software ------------------- Global Overview ----------------- 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
___ 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
```

Figure 127. Pre-Version 6.2 Display of the Global Overview Panel

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 displaying 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 display.

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.

```
BMC Software ------------- Group Display Selection -------------- 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
```

Figure 128. Using the PROFile Command to Display the Group Display Selection - Initial Visit

The first time you visit the Group Display Selection panel, you will find that all of the groups are marked as INCLUDED. By default CSM will show 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 made on the Group Display Selection panel are good only for the TSO user ID making the changes. No other users will be affected by changes made by another user.

Groups targeted to be displayed on the Global Overview panel have a status of INCLUDED. Ones that are not have a status of EXCLUDED. The *Repository* column indicates to the user whether the group's definition resides in the local repository (LOCAL) or in a remote repository, or 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 in the panel shown above except for groups KMZ1 and KMZ2, complete the panel as shown in Figure 129.

```
BMC Software ------------- Group Display Selection -------------- 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
```

Figure 129. Group Display Selection Panel Showing Groups to Exclude

Press the ENTER key to mark the groups as INCLUDED or EXCLUDED. After the panel is refreshed, you will see the status change as shown in Figure 130.

```
BMC Software ------------- Group Display Selection -------------- 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
```

Figure 130. Group Display Selection Panel Showing Change in Group Status

The changes to the list of groups become active when you enter the END command or press the 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 this:

```
BMC Software ----------- Global Overview - * Filtered * ---------- 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
```

Figure 131. Filtered Global Overview Panel

You can tell at a single glance if the Global Overview panel you are looking at has been filtered using the PROFILE command. You can see the indicator **`* Filtered *`** on the title line.

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	256 through 258
Creating a CSM object with CSM panels	259 through 281
Modifying object definitions	282
Modifying an object's schedule	294 through 307
Enabling cross-system management	308

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 1- to 8-characters long but it is recommended that you name the first group the SSID of the subsystem that you bring CSM up on.

Note: The default action of CSM is to bring up the object group which 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:

1. From the Global Overview panel (Figure 132), enter the Group primary command on the COMMAND line.

```
BMC Software ------------------ Global Overview ------------------ 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     SJSD     UP           UP                 NORM

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

Figure 132. Global Overview Panel: Adding a Group

The Object Groups panel is displayed (Figure 133).

```
BMC Software ------------------- Object Groups ------------------- 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
     ________    ________    ____    ________   ______________________________
```

Figure 133. Object Groups Panel

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 will be filled in when the Group Detail Control - Add panel is displayed.

The Group Detail Control - Add panel is displayed (Figure 134).

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

Figure 134. Group Detail Control - Add Panel

3. Fill in the information for these required fields:
   - Group Name
   - Command Prefix
   - Ruleset/Dataset
   - Ruleset Id
   - Ruleset 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” will be 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 135 shows an example of a completed panel.

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

Group Name        ===> SYSB____
Command Prefix    ===> CSMKM___________

Rule information:
Ruleset Dataset   ===> BAOKMZ.AAO62.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
```

Figure 135. Group Detail Control - Add Panel (Completed)

4. Press PF3/END to process the changes on this panel and return to the Object Groups panel (Figure 136).

```
BMC Software ------------------- Object Groups ------------------- 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        SJSD        KMZ1          20   GROUP LIST FOR SYSB
```

Figure 136. Object Groups Panel: One Group Added

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

Entering the Object's Information

Following are 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 MVS parent.

To define a new object named AAOCSM01:

1. From the Global Overview panel (Figure 124 on page 248), enter the ADD primary command on the COMMAND line or;

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

   When using the ADD primary command, you can include the name of the object which will be filled in the `Object Name` field when the Object Detail Control panel is displayed.

The Object Detail Control panel is displayed.

```
BMC Software ------------ Object Detail Control - Add ------------ 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 15:43

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

Figure 137. Object Detail Control Panel: Adding a New Object

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

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.

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 138).

```
BMC Software ----------------- CSM Object Types ------------------ 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
```

Figure 138. CSM Object Types

4. Select a type by placing an S next to it. Only one object type can be selected at a time.
5. Press ENTER or END to return to the Object Detail Control panel.

   The `Object Type` field will contain 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's 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.

6. Press ENTER to continue to the next object definition panel, Command Specifications.

Entering Command Specifications for an Object

The Command Specification panel allows you to specify an object's 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:

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

See Figure 139 for an example of the 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
```

Figure 139. Command Specifications Panel

Using Variables in the Object Definition Fields

On this 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 AutoOPERATOR-supplied IMF-prefixed variables.

Example

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

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

The variable is resolved when the CSM subsystem application processes the command and 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:

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

You can set the variable with a Rule, an EXEC or an entry in the BBIVARxx member. The variable already must 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:

- A period (.)
- A pound sign (#)
- A forward slash (/)
- A question mark (?)
- A 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 2 ampersand signs (&&); for example:

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

The resolved command that is passed to MVS is:

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

When MVS process 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. AutoOPERATOR tries to process the command as an EXEC using EXEC parameters.

Additional Fields to Complete

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 225 for more information)
- Start Command Limit value (refer to “Specifying Start Command Limits” on page 225 for more information)
- Whether to reset the Start Command count where a Y or N specification determines whether the object's command count shared variable is reset each midnight to zero.
- Specifying the 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 265).

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 139 on page 262), the Conditional Commands Specification panel is displayed. Figure 140 shows the panel where you specify conditional Start commands.

```
BMC Software --------- Conditional Command Specification --------- 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 140. Conditional Command Specifications Panel: Start

Figure 141 shows an example of the Stop commands panel.

```
BMC Software --------- Conditional Command Specification --------- 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
```

Figure 141. Conditional Command Specifications Panel: Stop

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 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's Start or Stop event has not occurred, an ALERT (ACM720A) will be issued informing you that the command has timed out.

1. Enter the number of minutes after which a conditional Start or Stop command will be issued for an object.
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 126 bytes long and can contain variables. Any variables specified will be resolved from the local subsystem’s SHARED pool.

   When a command is issued, and 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.
```

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 142 shows an example of the 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
```

Figure 142. Event Specifications 1 Panel

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

Figure 143 shows an example of the 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
```

Figure 143. Event Specifications 2 Panel

1. (Optional Step) Specify the target string in the `Abend #` fields to search for within in 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 whenthe fifth word of the IEF450I message contains the string ABEND=S0C7. In this case, the Exceptions Specification panel would be filled in as in Figure 143.

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.

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

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.
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 using the Elan Workstation product option (if it is installed at your site)
- Manage the ALERT queue
- Up to 10 TSO users can be notified 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 144 shows an example of the User Notification panel:

```
BMC Software -------- User Notification for AAOCSM01 - code ------- 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
```

Figure 144. User Notification Panel

1. Enter any valid 1-7 character TSO user IDs for each user to be notified of error conditions in object processing in the `TSO SEND Users` fields.
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 CLIST EXECs* or *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs* manual.

3. In the Object Alert Queue field, enter the name of the ALERT queue that will receive ALERTs for this object from CSM
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 replace for the ALERT's key and queue is sent to all PATROL Enterprise Manager workstations that have subscribed to receive ALERTs from this AutoOPERATOR. If there is already an ALERT with that key and queue on a PATROL Enterprise Manager workstation, 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 AutoOPERATOR. If there is already an ALERT with that key and queue on a PATROL Enterprise Manager workstation, it is not deleted before writing the new ALERT with that key/queue.

  ADD is the default.

NO
: The ALERT is not written to the connected PATROL Enterprise Manager workstations even if they have subscribed to receive ALERTs.

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 it's 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 first before you can start defining objects. Refer to "Creating a Group" on page 256 for more information about creating a group. This section describes only the process of associating the object with one or more groups.

Figure 145 shows an example of the Object Groups panel:

```
BMC Software ---------- Object Groups for AAOCSM01 - Add --------- 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                         23  Group List for SJSE
___ KMZ6        Member            4  Group List for SYSD
___ KMZ7                         80  test group 7
****************************** Bottom of data ******************************
```

Figure 145. Object Groups Panel

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.

1. To associate an object with a group, enter the (M)ember line command next to the group name you want to associate an object with.
2. 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.
3. To see all the objects within a group, enter the (L)ist line command next to the group name. Figure 146 shows an example of all the objects in a group named KMZ2.

```
BMC Software ---------- Objects in Group List KMZ2     ----------- AutoOPERATOR
COMMAND ===>                                                 SCROLL ===> CSR
                                                                TGT ---  KMZ1
Enter END or ENTER to continue

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

Figure 146. Object Groups Panel

4. 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 272) 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 back-ups or you need more system resources) with the Moveable 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.

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

   This example shows object AAOCSM01 is a member of group KMZ1 but moveable to KMZ2 and KMZ6

```
BMC Software ---------- Object Groups for AAOCSM01 - Add --------- 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                         23  Group List for SJSE
o__ KMZ6                          4  Group List for SYSD
___ KMZ7                         80  test group 7
****************************** Bottom of data ******************************
```

Figure 147. Object Groups Panel: Creating a Moveable Object

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 323

2. Press ENTER to continue to the next object definition panel, Object Requisites.

Defining the Object's Parents

Figure 148 shows the Object Requisites panels for AAOCSM01 where the object's parents can be added.

```
BMC Software --------- Object Requisites for AAOCSM01 - Add --------- 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_____      KMZ1____    THE BIG GIANT HEAD
****************************** Bottom of data ******************************
```

Figure 148. Object Requisites for AAOCSM01 Panel

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

```
BMC Software --------- Object Requisites for AAOCSM01 - Add --------- 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                                |
_ |                                                                           |
* ---------------------------------------------------------------------------*
```

Figure 149. Add Object Requisites for AAOCSM01 Panel

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

3. Press Enter to continue to the next object definition panel, Client/Server Relationships.

Defining the Object's Servers

Figure 150 shows the Client/Server Relationships of 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)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 ********************************
```

Figure 150. Client/Server Relationships for AAOCSM01 Panel

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

```
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                                 |
 * ---------------------------------------------------------------------------*
```

Figure 151. Add Client/Server Relationships for AAOCSM01 Panel

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

4. Press Enter to continue to the next object definition panel, Object Scheduling.

Figure 152 shows the Object Scheduling panel for AAOCSM01

```
BMC Software -------- Object Scheduling for AAOCSM01 -  Add ------- 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 ********************************
```

Figure 152. Object Scheduling for AAOCSM01 Panel

For this example, a schedule is not created for object AAOCSM01. Refer to "Modifying an Object's Routine Schedule" on page 294 for information about creating a schedule for an object.

5. Press Enter to return to the CSM Object Detail Control panel (Figure 153).

```
BMC Software ------------ Object Detail Control - Add ------------ 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 ********************************
```

Figure 153. Object Detail Control Panel

6. Press PF3/END to return to the CSM Global Overview panel ().

```
BMC Software ------------------ Global Overview ------------------ 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
```

Figure 154. CSM Global Overview Panel

Object AAOCSM01 is added as the child of MVS.

Repeat this procedure for all the new Normal objects 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 259 through 277). The differences are:

- 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 Transient object (pages 259 through 277).

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

The following table summarizes the similarities and differences for the three object types.

Table 21. 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)epeat 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)epeat line command:

1. From the Global Overview panel (Figure 155), enter the (R)epeat line command in the LC field next to the object that you want to duplicate.

```
BMC Software ------------------ Global Overview ------------------ 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

LC  Object    Group    System   Actual       Desired      Ind    Type
    ________  ________ ________ ____________ ____________ _____  ________
___ AAOCMS01  AO6A              UNKNOWN      UNKNOWN              NORM
___ AAOCSM02  AO6A              UNKNOWN      UNKNOWN              NORM
___ CSBEMSTR  AO62       SJSE   STARTING     UP           XE      NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 155. Repeating an Object

The Repeat Object Definition panel is displayed (Figure 156).

```
BMC Software ------------------ Global Overview ------------------ 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
___  AAOCSM02  AO6A                UNKNOWN       UNKNOWN             NORM
___  CSBEMSTR  AO62      SJSE      STARTING      UP            XE    NORM
___  MVS
****************************** Bottom of data ******************************
```

Figure 156. Repeat Object Definition Panel

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

```
BMC Software ------------------ Global Overview ------------------ AutoOPERATOR
C | BMC Software            Repeat Object Definition           AutoOPERATOR
  | COMMAND ===> ___________________________________________ TGT --- AO6A      | A
P |                                                                          |
  |                                                                          |
  |                                                                          |
  | Old Object Name ===> AAOCMS01                                            |
L |                                                                          |
  | New Object Name ===> AAOCSM03                                            |
  |                                                                          |
  |                                                                          |
L | Press ENTER to continue or END / CANcel to abort                         |
  |__________________________________________________________________________|
R__  AAOCMS02  AO6A                UNKNOWN       UNKNOWN             NORM
___  AAOCSM02  AO6A                UNKNOWN       UNKNOWN             NORM
___  CSBEMSTR  AO62      SJSE      STARTING      UP            XE    NORM
___  MVS
****************************** Bottom of data ******************************
```

Figure 157. Add New Object Panel: Repeat Line Command Example 2

3. Press Enter to process the changes on this panel. The Object Detail Control panel appears (Figure 158).

```
BMC Software ----------- Object Detail Control - Repeat ---------- 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
```

Figure 158. Object Detail Control - Repeat

4. Follow the sequence of the panels be pressing Enter and fill in the information for new object AAOCSM03.

5. After all the information for object AAOCSM03 is completed, the Object Detail Control panel is re-displayed. To process the creation of object AAOCSM03, press PF3/END.

6. The Global Overview panel is re-displayed with the new object.

```
BMC Software ------------------ Global Overview ------------------ 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
     ________ ________ ________ ____________ ____________ _____ ________
___ AAOCMS01  AO6A              UNKNOWN      UNKNOWN              NORM
___ AAOCSM02  AO6A              UNKNOWN      UNKNOWN              NORM
___ AAOCSM03  AO6A              UNKNOWN      UNKNOWN              NORM
___ CSBEMSTR  AO62     SJSE     STARTING     UP            XE     NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 159. AAOCSM03 Object Created

Modifying Object Definitions

It may 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 MVS 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:

1. Display the Global Overview panel (Figure 160).

```
BMC Software ------------------ Global Overview ------------------ 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              UNKNOWN      UNKNOWN             NORM
___ AAOCSM09  AO6A              UNKNOWN      UNKNOWN             NORM
___ AAOCSM10  AO6A              UNKNOWN      UNKNOWN             NORM
___ CSBEMSTR  AO62     SJSE     STARTING     UP           XE     NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 160. Global Overview Panel: Adding a Parent

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

The Object Detail Control panel is displayed.

```
BMC Software ------------ Object Detail Control - Edit ----------- 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
```

Figure 161. Adding a Parent - Step 2

3. Enter the REQ primary command to skip the panels in the ring and display the Object Requisites panel (Figure 162).

```
BMC Software ------- Object Requisites for AAOCSM08 - Edit ------- 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 ******************************
```

Figure 162. Adding a Parent - Step 3

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

```
BMC Software --------- Object Requisites for AAOCSM08 - Add --------- 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                                |
_ |                                                                           |
* ---------------------------------------------------------------------------*
```

Figure 163. Adding a Parent - Step 4

5. Type in the new object name and the group name of the new parent to be added to CSM object AAOCSM08:

```
BMC Software --------- Object Requisites for AAOCSM08 - Add --------- 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                                |
_ |                                                                           |
* ---------------------------------------------------------------------------*
```

Figure 164. Adding a Parent - Step 5

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

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

```
BMC Software ------------------ Global Overview ------------------ 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                                      NORM
___ AAOCSM09  AO6A     SJSC                                      NORM
___ AAOCSM10  AO6A     SJSC                                      NORM
___ CSBEMSTR  AO62     SJSE      STARTING      UP          XE   NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 165. Adding a Parent - Step 6

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 166 shows the result for object AAOCSM08.

```
BMC Software ------ Object Requisites for AAOCSM08 - Browse ------ 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 ******************************
```

Figure 166. Adding a Parent - Step 7

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

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

Deleting Parents

In this example, parent MVS 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:

1. Display the Global Overview panel (Figure 167).

```
BMC Software ------------------ Global Overview ------------------ 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                                      NORM
___ AAOCSM09  AO6A     SJSC                                      NORM
___ AAOCSM10  AO6A     SJSC                                      NORM
___ CSBEMSTR  AO62     SJSE     STARTING      UP          XE    NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 167. Global Overview Panel: Deleting a Parent

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

   Figure 168 is displayed and shows the two parents of AAOCSM08; MVS and AAOCSM10.

```
BMC Software ------- Object Requisites for AAOCSM08 - Edit ------- 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 ******************************
```

Figure 168. Deleting a Parent: Step 2

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

The line for AAOCSM10 disappears from the panel (Figure 169).

```
BMC Software ------- Object Requisites for AAOCSM08 - Edit ------- 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 ********************************
```

Figure 169. Deleting a Parent: Step 3

4. Press PF3/END. The Object Detail Control panel is re-displayed.
5. Press PF/3END again. The Global Overview panel is re-displayed.

```
BMC Software ------------------ Global Overview ------------------ 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                                      NORM
___ AAOCSM09  AO6A     SJSC                                      NORM
___ AAOCSM10  AO6A     SJSC                                      NORM
___ CSBEMSTR  AO62     SJSE     STARTING     UP            XE   NORM
___ MVS
****************************** Bottom of data ********************************
```

Figure 170. Deleting a Parent: Step 5

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 may be necessary from time to time to make additions or deletions to an object's list of servers.

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:

1. Display the Global Overview panel (Figure 171).

```
BMC Software ------------------ Global Overview ------------------ 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                                      NORM
___ AAOCSM09  AO6A     SJSC                                      NORM
___ AAOCSM10  AO6A     SJSC                                      NORM
___ CSBEMSTR  AO62     SJSE     STARTING     UP            XE   NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 171. Global Overview Panel: Adding a Server

2. Enter the E line command (for (E)dit) next to AAOCSM10.
3. The Object Detail Control panel for AAOCSM10 is displayed.

```
BMC Software ------------ Object Detail Control - Edit ----------- 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 BAOKMZ1  on 01/30/2001 at 17:36

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

Figure 172. Adding a Server - Step 3

4. Enter the primary command CLS to display the Client/Server Relationships panel for AAOCSM10 (Figure 173)

```
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 ********************************
```

Figure 173. Adding a Server - Step 4

5. Enter the ADD primary command. The Add Client / Server Relationships panel is displayed.

```
BMC Software ---- Client/Server Relationships for AAO41 - 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                                  |
 * ----------------------------------------------------------------------------*
```

Figure 174. Adding a Server - Step 5

6. Enter the Server name, the Group Name, and Connect and Disconnect commands for the new server. Prefix the commands with a pound sign (#) if it is to be issued through MVS.

   The completed panel looks like Figure 175.

```
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 ********************************
```

Figure 175. Adding a Server - Step 6

7. Press PF3/END. The Object Detail Control panel is re-displayed.
8. Press PF/3END again. The Global Overview panel is re-displayed.

```
BMC Software ------------------ Global Overview ------------------ 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                                      NORM
___ AAOCSM09  AO6A     SJSC                                      NORM
___ AAOCSM10  AO6A     SJSC                                      NORM
___ CSBEMSTR  AO62     SJSE     STARTING     UP            XE   NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 176. Global Overview Panel: Server Added

9. To view the servers for object AAOCSM10, enter the line command c(L)ients next to AAOCSM10. Figure 177 is displayed.

```
BMC Software    -------------- Clients of AAOCSM10  ------------- AutoOPERATOR
COMMAND ===>                                                 TGT --- AO6A

     AAOCSM09  Child of MVS
****************************** Bottom of data ******************************
```

Figure 177. Clients of AAOCSM10 Panel

10. Press PF/3END to return to the Global Overview panel.
11. 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:

1. Display the Global Overview panel (Figure 178).

```
BMC Software ------------------ Global Overview ------------------ 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                                      NORM
___ AAOCSM09  AO6A     SJSC                                      NORM
___ AAOCSM10  AO6A     SJSC                                      NORM
___ CSBEMSTR  AO62     SJSE     STARTING     UP            XE   NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 178. Global Overview Panel: Deleting a Server

2. Enter the E line command (for (E)dit) next to AAOCSM09.
3. The Object Detail Control panel for AAOCSM09 is displayed.

```
BMC Software ------------ Object Detail Control - Edit ----------- 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
```

Figure 179. Deleting a Server - Step 3

4. Enter the primary command CLS to display the Client/Server Relationships panel for AAOCSM09. Figure 180 is displayed and shows that AAOCSM09 is the only server assigned to AAOCSM10.

```
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
                            Connect Command
                            Disconnect Command
_    AAOCSM10    AO6A____
                            #F AAOCSM09,Connect AAOCSM10_____________
                            #F AAOCSM09,Disconnect AAOCSM10__________

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

Figure 180. Deleting a Server - Step 4

5. Enter the D line command (for (D)elete Server) next to AAOCSM10.

   The line for AAOCSM10 disappears from the panel (Figure 181).

```
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
                            Connect Command
                            Disconnect Command
****************************** Bottom of data ******************************
```

Figure 181. Deleting a Server - Step 5

6. Press PF3/END. The Object Detail Control panel is re-displayed.

7. To save this change, issue the SAVE primary command on the Global Overview panel.

8. Press PF/3END again. The Global Overview panel is re-displayed.

```
BMC Software ------------------ Global Overview ------------------ 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                                      NORM
___ AAOCSM09  AO6A     SJSC                                      NORM
___ AAOCSM10  AO6A     SJSC                                      NORM
___ CSBEMSTR  AO62     SJSE     STARTING     UP            XE   NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 182. Global Overview Panel: Server Deleted

9. To view the servers for object AAOCSM10, enter the line command c(L)ients next to AAOCSM10. Figure 183 is displayed.

```
BMC Software    -------------- Clients of AAOCSM10  ------------- AutoOPERATOR
COMMAND ===>                                                 TGT --- AO6A

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

Figure 183. Clients of AAOCSM10 Panel

10. Press PF/3END to return to the Global Overview panel.

11. 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 pm to 7:00 pm
- Fridays from 2:00 am to 3:00 am
- Fridays from 6:00 am to 9:00 am
- Sundays from 3:00 pm to 6:00 pm

Because an object cannot be available when its parent is unavailable, your schedule for the object also must reflect the down times of its parent(s).

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:

1. Enter the B line command (for (B)rowse) next to the object AAOCSM10 (Figure 184).

```
BMC Software ------------------ Global Overview ------------------ 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                                      NORM
___ AAOCSM08  AO6A     SJSC                                      NORM
___ AAOCSM09  AO6A     SJSC                                      NORM
B__ AAOCSM10  AO6A     SJSC                                      NORM
___ CSBEMSTR  AO62     SJSE     STARTING     UP            XE   NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 184. Checking the Schedule of a Parent

2. The Object Detail Control - Browse panel is displayed (Figure 185).

```
BMC Software ------------ Object Detail Control - Browse ----------- 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 BAOKMX1M  on 01/30/2001 at 17:36

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

Figure 185. Checking the Schedule of a Parent: Step 2

3. Enter the SCH primary command to display the Object Scheduling panel for AAOCSM10.

   Figure 186 shows that AAOCSM10 is scheduled to be DOWN on Mondays from 3:00 pm to 5:00 pm.

```
BMC Software ------- Object Scheduling for AAOCSM10 - Browse ----- 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 ******************************
```

Figure 186. Checking the Schedule of a Parent: Step 3

4. Make a note of this time and press PF3/END to leave this panel and return to the Object Detail Control panel.

5. Press PF/3END again to return to the Global Overview panel.

6. 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 pm to 5:00 pm. You also want to schedule AAOCSM07 to be DOWN at the following times:

- Wednesdays from 3:00 pm to 7:00 pm
- Fridays from 2:00 am to 3:00 am
- Fridays from 6:00 am to 9:00 am
- Sundays from 3:00 pm to 6:00 pm

To avoid scheduling conflicts with the parent (AAOCSM10), you must also include Monday 3:00 pm to 5:00 pm 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 304.

To modify AAOCSM07's schedule:

1. From the CSM Global Overview panel, enter the E line command (for (E)dit) next to AAOCSM07.
2. The Object Detail Control panel for AAOCSM07 is displayed.

```
BMC Software ------------ Object Detail Control - Edit ----------- 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
```

Figure 187. Modifying an Object Schedule - Step 1

3. Enter the primary command SCH to display the Object Scheduling panel for AAOCSM07 (Figure 188).

```
BMC Software ------- Object Scheduling for AAOCSM07 - Edit ------- 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 ********************************
```

Figure 188. Modifying an Object Schedule - Step 2

4. Enter the schedule for AAOCSM07's DOWN times using the ADD primary command. Figure 189 is displayed.

```
BMC Software ------- Object Scheduling for AAOCSM07 - Edit ------- 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                                  |
  |----------------------------------------------------------------------------
```

Figure 189. Modifying an Object Schedule - Step 3

5. Use 24-hour clock format to enter all times. Repeat until all DOWN times are added.

   Figure 190 on page 298 shows the Object scheduling panel with all the DOWN times filled in.

Note that the DOWN time for parent object AAOCSM10 is also included (Monday from 3:00 PM to 5:00 PM).

```
BMC Software ------- Object Scheduling for AAOCSM07 - Edit ------- 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 ********************************
```

Figure 190. Modifying an Object Schedule - Step 4

AAOCSM07 will be DOWN:

- Fridays from 2:00 am to 3:00 am
- Fridays from 6:00 am to 9:00 am
- Mondays from 3:00 pm to 5:00 pm (because AAOCSM07's parent, AAOCSM10 is down at that time)
- Sundays from 3:00 pm to 6:00 pm
- Wednesday from 3:00 pm to 7:00 pm

6. Press PF3/END. The Object Detail Control panel is re-displayed.
7. Press PF/3END again. The Global Overview panel is re-displayed.

```
BMC Software ------------------ Global Overview ------------------ 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                                      NORM
___ AAOCSM08  AO6A     SJSC                                      NORM
___ AAOCSM09  AO6A     SJSC                                      NORM
___ AAOCSM10  AO6A     SJSC                                      NORM
___ CSBEMSTR  AO62     SJSE     STARTING     UP            XE   NORM
___ MVS
****************************** Bottom of data ********************************
```

Figure 191. Global Overview Panel: Schedule Modified

8. 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 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 am to 3:00 am.

To modify AAOCSM09's schedule:

1. Enter the E line command (for (E)dit) next to AAOCSM09.

2. The Object Detail Control panel for AAOCSM09 is displayed.

```
BBMC Software ------------ Object Detail Control - Edit ----------- 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
```

Figure 192. Using Object Schedule Tags

3. Enter the primary command SCH to display the Object Scheduling panel for AAOCSM09 (Figure 193).

```
BMC Software ------- Object Scheduling for AAOCSM09 - Edit ------- 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 ******************************
```

Figure 193. Using Object Schedule Tags - Step 3

4. Enter the primary command ADD. Figure 194 is displayed.

```
BMC Software ------- Object Scheduling for AAOCSM09 - Edit ------- 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                                   |
  |-----------------------------------------------------------------------------
```

Figure 194. Modifying an Object Schedule - Step 4

5. Enter the word `DLY` to specify that this object is supposed to be DOWN every day (Monday through Sunday) and the times, using 24-hour clock format (`02:00` and `03:00`) as shown in Figure 195.

```
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                                   |
  |-----------------------------------------------------------------------------
```

Figure 195. Using Object Schedule Tags - Step 5

You can enter additional DOWN times using the ADD primary command. Press PF3/END to redisplay the Object Scheduling panel.

6. Press PF3/END. The Object Detail Control panel is re-displayed.

7. Press PF/3END again. The Global Overview panel is re-displayed.

```
BMC Software ------------------ Global Overview ------------------ 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                                      NORM
___ AAOCSM08  AO6A     SJSC                                      NORM
___ AAOCSM09  AO6A     SJSC                                      NORM
___ AAOCSM10  AO6A     SJSC                                      NORM
___ CSBEMSTR  AO62     SJSE     STARTING     UP            XE   NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 196. Global OverviewPanel: Schedule Modified

8. To save the changes, enter the SAVE primary command.

   Changes take effect when you enter the primary command (BU)ild on the COMMAND line for the group.

The changes take effect when the BUILD command is issued for the group this object is associated with. CSM will notice if there is a 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-hand corner when you try to exit from the scheduling panel.

For example, suppose you have completed the schedule panel for AAOCSM07 as shown in Figure 197 where the object will be DOWN on Wednesdays, Fridays, and Sundays.

```
BMC Software ------- Object Scheduling for AAOCSM07 - Edit ------- 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 ********************************
```

Figure 197. Schedule Conflicts: Example 1

Now suppose you did not perform the task described in “Determining the Schedule of the Object’s Parent” on page 294 and you did not know that AAOCSM07’s parent (AAOCSM10) is scheduled to be DOWN on Mondays from 3:00 pm to 5:00 pm.

When you press PF3/END to save this schedule, you will receive a short error message as shown in the upper right-hand corner of Figure 198.

```
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 ********************************
```

Figure 198. Schedule Conflicts: Example 2

1. Press PF1 to see the entire error message.

   The conflict is described as shown in Figure 199.

```
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 **************************
```

Figure 199. Schedule Conflicts: Example 3

2. To leave this panel, you must erase all the entries you just entered for AAOCSM07.

3. Press PF3/END to return to the CSM panel.

4. To successfully enter the schedule for AAOCSM07, you must perform the procedure described in “Determining the Schedule of the Object’s Parent” on page 294, note all the times the parent AAOCSM10 is DOWN, and then perform the steps in “Modifying the Object’s Schedule” on page 296.

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 pm to 24:00
```

To access the Global Calendar Override panel:

1. Issue the CALendar primary command from the CSM Global Overview panel.

   The Global Calendar Override panel is shown (Figure 200).

```
BMC Software ------------- Global Calendar Override -------------- 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
_   ________   ________       ________ _____     _____ ______________________
                                                        ________________
```

Figure 200. Global Calendar Override Panel

   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).

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 down time 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-hand corner when you press PF3/END.

Figure 201. shows an example of a Global Calendar Override panel filled in.

```
BMC Software ------------- Global Calendar Override -------------- 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   AO6A           01/12/01  18:00     24:00   Backup copies

 **************************BOTTOM OF DATA **************************
```

Figure 201. Example of Global Calendar Override for Object AAOCSM07

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 object MVS:

1. To edit the MVS object, enter E (for (E)dit) in the line command column next to MVS on the Global Overview panel (Figure 202).

```
BMC Software ------------------ Global Overview ------------------ 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      STARTING      UP            XE    NORM
___ AAOCSM10  AO6A      SJSC      STARTING      UP            NXE   NORM
___ AO6B      AO6A                UNKNOWN       UNKNOWN             NORM
___ CSBEMSTR  AO62      SJSE      UP            UP                  NORM
_E_ MVS
******************************* Bottom of data ********************************
```

Figure 202. Editing the MVS Object to Enable Cross-System Management

The Object Detail Control - Edit panel for the MVS object is displayed (Figure 203).

```
BMC Software ------------ Object Detail Control - Edit ----------- 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
```

Figure 203. Object Detail Control - Edit Panel for MVS

The Object Detail Control - Edit panel for the MVS panel includes the field `Enable PLEX ===> Y_` where the default setting is `YES`.

2. To enable cross-system management, ensure that `YES` is entered in this field.
3. Press PF3/END.
4. To complete enabling cross-system management, you must repeat this process on the other BBI-SS PAS targets.

Routine Object Tasks

The following section describes routine tasks you can perform on CSM objects from the Global Overview panel:

Task	Pages
Starting objects	311
Stopping objects	314
Bouncing objects	316
Resetting objects	318
Canceling objects	321
Moving objects	323
Issuing an EXEC associated with a CSM ALERT	326

Starting an Object

The purpose of CSM is to help manage the availability of objects in your system. This 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 whose Actual and Desired states are DOWN from the Global Overview panel.

1. Display the Global Overview panel (Figure 204).

```
BMC Software ------------------ Global Overview ------------------ 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         DOWN                NORM
___ AAOCSM09 AO6A     SJSC     STARTING     UP           XE     NORM
___ AAOCSM10 AO6A     SJSC     STARTING     UP           NXE    NORM
___ CSBEMSTR AO62     SJSE     STARTING     UP           XE     NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 204. Global Overview Panel: Starting an Object

2. Enter the S line command (for (S)tart) next to AAOCSM07.

   Figure 205 is displayed.

```
BMC Software ------------------ Global Overview ------------------ AutoOPERATOR
C | BMC Software          Confirm: Start for AAOCSM07          AutoOPERATOR|
  | COMMAND ===>                                          TGT --- AO6A     |  A
P |                                                                        |
  |                                                                        |
  |                                                                        |
  |  Please confirm START for AAOCSM07   ? N (Y/N)                         |
L |                                                                        |
  |  START all PARENTS of AAOCSM07       ? Y (Y/N)                         |
  |                                                                        |
  |  START now, but STOP as scheduled    ? Y (Y/N)                         |
L |                                                                        |
  |                                                                        |
S |                                                                        |
_ |                                                                        |
_ |  Press ENTER to continue or END / CANcel to cancel request             |
_ |________________________________________________________________________|
___  MVS
****************************** Bottom of data ******************************
```

Figure 205. Object Start Confirmation Panel

3. Reply to the start confirmation panel; the following 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 CSM's control 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 shutdown at the scheduled time. The object will show an actual state of UP and a desired state of UPFORCE.

4. Press ENTER.

   Figure 206 is displayed.

```
BMC Software ------------------ Global Overview ------------------ 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     STARTING     UPEARLY            NORM
___ AAOCSM09 AO6A     SJSC     STARTING     UP           XE    NORM
___ AAOCSM10 AO6A     SJSC     STARTING     UP           NXE   NORM
___ CSBEMSTR AO62     SJSE     STARTING     UP           XE    NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 206. Global Overview Panel: Object Started

The Global Overview now shows that the actual state of AAOCSM07 is STARTING and the desired state is UPEARLY. As soon as the object completes starting, 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. Objects that are parents of the affected object are not being started by CSM. This usually is 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 start-up 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.

1. Display the Global Overview panel (Figure 207).

```
BMC Software ------------------ Global Overview ------------------ 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     UP           UP           XE    NORM
___ AAOCSM10 AO6A     SJSC     STARTING     UP           NXE   NORM
___ CSBEMSTR AO62     SJSE     STARTING     UP           XE    NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 207. Global Overview Panel: Stopping an Object

2. Enter the P line command (for sto(P)) next to AAOCSM09.

   Figure 208 is displayed.

```
BMC Software ------------------ Global Overview ------------------ 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 |                                                                          |
  |                                                                          |
_ |                                                                          |
P |                                                                          |
_ |  Press ENTER to continue or END / CANCEL to cancel request               |
_ |__________________________________________________________________________|
___  MVS
****************************** Bottom of data ******************************
```

Figure 208. Object Stop Confirmation Panel

3. Reply to the stop confirmation panel; the following 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 CSM's control 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.

4. Press ENTER.

   Figure 209 is displayed.

```
BMC Software ------------------ Global Overview ------------------ 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
    ________ ________ ________ ____________ ____________ _____ ________
___ AAOCSM07 AO6A     SJSC     DOWN         UP                  NORM
___ AAOCSM09 AO6A     SJSC     STOPPING     DOWNFORCE    XE     NORM
___ AAOCSM10 AO6A     SJSC     STARTING     UP           NXE    NORM
___ CSBEMSTR AO62     SJSE     STARTING     UP           XE     NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 209. Global Overview Panel: Object Stopped

The Global Overview now shows that the actual state of AAOCSM09 is STOPPING and the desired state is DOWNFORCE. As soon as the object completes stopping, the actual state will be DOWN.

Hint

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.

Objects that are children of the affected object are not being stopped by CSM. This usually is 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.

Bouncing an Object

From time to time 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. This 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):

1. Display the Global Overview panel (Figure 210).

```
BMC Software ------------------ Global Overview ------------------ 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     UP           UP           NXE   NORM
___ CSBEMSTR AO62     SJSE     STARTING     UP           XE    NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 210. Global Overview Panel: Bouncing an Object

2. Enter the O line command (for b(O)unce) next to AAOCSM10.

   Figure 211 is displayed.

```
BMC Software ------------------ Global Overview ------------------ AutoOPERATOR
C | BMC Software          Confirm: Bounce for AAOCSM10         AutoOPERATOR |
  | COMMAND ===>                                          TGT --- AO6A      | A
P |                                                                         |
  |                                                                         |
  |                                                                         |
  |  Please confirm Bounce for AAOCSM10    ? N (Y/N)                        |
L |                                                                         |
  |                                                                         |
  |                                                                         |
  |                                                                         |
L |                                                                         |
  |                                                                         |
_ |                                                                         |
O |  Press ENTER to continue or END / CANcel to cancel request              |
_ |_________________________________________________________________________|
___  CSBEMSTR  AO62      SJSE      STARTING      UP            XE    NORM
___  MVS
****************************** Bottom of data ******************************
```

Figure 211. Confirm Bounce Panel

3. Reply to the bounce confirmation panel; the following describes the replies that you can enter on this panel:

   – To bounce AAOCSM10, enter `Y`.

     CSM bounces AAOCSM10.

   – To cancel bouncing AAOCSM10, accept the default setting of `N` for `Please confirm BOUNCE for AAOCSM10`.

     The object bounce request is rejected.

4. Press ENTER.

   Figure 212 is displayed.

```
BMC Software ------------------ Global Overview ------------------ 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         UP           NXE   NORM
___ CSBEMSTR AO62     SJSE     STARTING     UP           XE    NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 212. Global Overview Panel: Object Bounce

The actual state of the object is DOWN. When the object restarts after being DOWN, the actual state will appear as UP.

Hint

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.

This 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.

Resetting an Object

Occasionally, objects need to be reset back to schedule control. This is necessary 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 may 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:

1. Display the Global Overview panel (Figure 213).

```
BMC Software ------------------ Global Overview ------------------ 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     UP           UP           NXE   NORM
___ CSBEMSTR AO62     SJSE     STARTING     UP           XE    NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 213. Global Overview Panel: Resetting an Object

2. Enter the T line command (for rese(T)) next to AAOCSM10.

   Figure 214 is displayed.

```
BMC Software ------------------ Global Overview ------------------ 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  AO62      SJSE      STARTING      UP            XE    NORM
___  MVS
****************************** Bottom of data ******************************
```

Figure 214. Confirm Resetting an Object Panel

3. Reply to the reset confirmation panel; the following 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. • BMC Software-supplied time-initiated Rule ACMRC060 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 is the default setting.

  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.

4. Press ENTER.

Figure 215 is displayed.

```
BMC Software ------------------ Global Overview ------------------ 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     UP           UP           NXE   NORM
___ CSBEMSTR AO62     SJSE     STARTING     UP           XE    NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 215. Global Overview Panel: Object Reset

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 MVS object.

By indicating a scope of C, CSM returns all of the children of MVS to schedule control. Every object in the CSM database is a child of MVS 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 may 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:

1. Display the Global Overview panel (Figure 216).

```
BMC Software ------------------ Global Overview ------------------ 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     UP           UP           NXE   NORM
___ CSBEMSTR AO62     SJSE     STARTING     UP           XE    NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 216. Global Overview Panel: Canceling an Object

2. Enter the C line command (for (C)ancel) next to AAOCSM10.

   Figure 217 is displayed.

```
BMC Software ------------------ Global Overview ------------------ AutoOPERATOR
C | BMC Software          Confirm: Cancel for AAOCSM10         AutoOPERATOR|
  | COMMAND ===>                                            TGT --- AO6A     |  A
P |                                                                          |
  |                                                                          |
  |                                                                          |
  |  Please confirm Cancel for AAOCSM10    ? N (Y/N)                         |
L |                                                                          |
  |                                                                          |
  |  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 ******************************
```

Figure 217. Confirm Cancel Panel

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.
4. Press ENTER.

   Figure 218 is displayed.

```
BMC Software ------------------ Global Overview ------------------ 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 AO62     SJSE     STARTING     UP           XE    NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 218. Global Overview Panel: Object Cancel

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. This is called making an object moveable (refer to "Moving an Object Between Groups" on page 273 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 the following conditions:

- The destination BBI-SS PAS has CSM installed and operating
- BBI-SS to BBI-SS PAS communications between the two subsystems is active
- The 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 prefered 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 on that group when it becomes active. It will remain on the group it was moved to until moved to another group.

Note: You cannot move an object to another BBI-SS PAS where it 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 that BBI-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.

1. Display the Global Overview panel (Figure 219).

```
BMC Software ------------------ Global Overview ------------------ 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     UP           UP           NXE   NORM
___ CICSA    AA6A     SYSA     UP           UP                 NORM
___ CSBEMSTR AO62     SJSE     STARTING     UP           XE    NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 219. Global Overview Panel: Moving an Object

2. Enter the M line command (for (M)ove) 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 you specify as shown in Figure 220.

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.

Confirm Move Panel is displayed (Figure 220).

```
BMC Software           Confirm: Move for CICSA          AutoOPERATOR
| COMMAND ===>                                          TGT --- NS61|
|                                                                   |
|                                                                   |
|                                                                   |
|  Please confirm MOVE for CICSA ? Y    (Y/N)                   |
|                                                                   |
|  Group id to MOVE CICSA to     ? NS62                         |
|                                                                   |
|                                                                   |
|  Press ENTER to continue or END / CANcel to cancel request        |
-------------------------------------------------------------------
```

Figure 220. Confirm Move Panel

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.

1. Press ENTER. The Global Overview panel is redisplayed..

```
BMC Software ------------------ Global Overview ------------------ 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     UP           UP           NXE   NORM
___ CICSA    AA6A     SYSA     DOWN         DOWNFORCE          NORM
___ CSBEMSTR AO62     SJSE     STARTING     UP           XE    NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 221. Global Overview Panel: Moving an Object

The CICSA 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 the object has clients.
N Indicates the object has children (dependents).
X Indicates 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 e(X)ec line command next to the object in the LC column.

Some CSM objects may 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 procedure describes how to do this.

To schedule an EXEC associated with a CSM ALERT:

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.

2. Enter the EX line command (for (EX)ec) in the LC column next to AAOCSM09 (Figure 222).

```
BMC Software ------------------ Global Overview ------------------ 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 AO62     SJSE     STARTING     UP           XE    NORM
___ MVS
****************************** Bottom of data ******************************
```

Figure 222. Global Overview Panel: Scheduling an EXEC

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 allow you to manually track and to 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's control. For more information about using the EXECs, refer to "Monitoring CSM Object States Manually" on page 330.

"What the Possible Desired States Are" on page 327 and "What the Possible Actual States Are" on page 330 describe the possible desired and actual state a CSM object can have.

What the Possible Desired States Are

Table 22 on page 328 shows the possible desired states 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:

1. The first column lists possible desired states.
2. The second column provides a brief description of the desired state.
3. The third column shows what command was issued against the object.
4. 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 22. Possible Desired States an Object Can Have

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 22. Possible Desired States an Object Can Have (Continued)

Resulting Desired State	Description	Command That Was Issued	Original Desired Object State
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

Note: *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.

What the Possible Actual States Are

Table 23 on page 330 shows the possible actual (or current) states 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:

1. The first column lists the name of the actual state.
2. The second column describes the state.

Table 23. Possible Actual States

Actual State Name	Description
UP	Object has started successfully
STARTING	Intermediate state of UP: 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, then the object probably has encountered an error.
UPMANUAL	Object has been brought UP (from a desired state of DOWN) outside of CSM's 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 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's 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

  Allows you to start or stop CSM objects without using the CSM dialog panels.

Using the CSMINFO EXEC to Manually Display CSM Object States

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, 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 UBBPARM(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 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 for REXX EXECs* book.

Using the CSMACT EXEC to Manually Modify Object States

Under ordinary circumstances, CSM should control the objects according to their schedules. The CSMACT EXEC allows you to change an object's state without using the CSM dialog panels. For example, you may 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's control and CSM can still track the state of the object. CSMACT can be scheduled on the command line of any 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 332.
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 is a list of valid commands you can use with the CSMACT EXEC to affect the state of an object. They are listed in alphabetical order.

Command	Definition
BOUNCE	Stops and immediately restarts an object if the original desired state for the object is UP. When used on an object that is UP, UPEARLY, UPFORCE, UPLOGICAL, or UPLOGICALE, the resultant desired state is UP. When issued on an object that is COMPLETE, the resultant desired state is COMPLETE.
CANCEL	Issues the object's cancel command. When issued on an object in any state, the resultant desired state is DOWNFORCE.
DOWN	Stops an object. When issued on an object that is DOWN, the resultant desired state is DOWN. When issued on an object that is in any state except DOWN, the resultant desired state is DOWNFORCE. Requires the DEP or NODEP parameter.
DOWNEARLY	Stops an object and continues under CSM's control. When issued on any object, the resultant desired state is DOWNEARLY. Requires the DEP or NODEP parameter.

Command	Definition
DOWNLOGICAL	Stops a grouping object and removes it from CSM's control. When issued on an object that is in any state, the resultant desired state is DOWNLOGICAL. Requires the DEP or NODEP parameter.
DOWNLOGICALE	Stops a grouping object and continues under CSM's control. When issued on an object that is in any state, the resultant desired state is DOWNLOGICALE. Requires the DEP or NODEP parameter.
MOVE	Moves an object to another subsystem. Requires the 4-character subsystem ID of the moved to subsystem. When an object that has an actual state UP is moved, the resultant actual state on the moved to BBI-SS PAS is UPFORCE and is DOWN on the moved from BBI-SS PAS.
RESET	Sets an object's desired state based on its schedule and returns an object to CSM's control. When issued on an object that is UP, the resultant desired state is UP. When issued on an object that is DOWN, the resultant desired state is DOWN. You can use the DEP or REQ parameters but they are not required.
RESETC	Resets an object and the command counter to zero. When issued on an object that is UP, the resultant desired state is UP. When issued on an object that is DOWN, the resultant desired state is DOWN. You can use the DEP or REQ parameters but they are not required.
UP	Starts an object. When issued on an object that is UP, the resultant desired state is UP. When issued on an object that is in any state, the resultant desired state is UPFORCE. Requires the REQ or NOREQ parameter.
UPEARLY	Starts an object and continues under CSM's control. When issued on an object that is in any state, the resultant desired state is UPEARLY. Requires the REQ or NOREQ parameter.

Command	Definition
UPLOGICAL	Starts a grouping object. When issued on an object that is in any state, the resultant desired state is UPLOGICAL or COMPLETE. Requires the REQ or NOREQ parameter.
UPLOGICALE	Starts a grouping object. When issued on an object that is in any state, the resultant desired state is UPLOGICALE or COMPLETE. Requires the REQ or NOREQ parameter.

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 223 shows the settings for group KMZ1.

.

```
BMC Software  ----------- Group Detail Control - Edit   ---------- AutoOPERATOR
COMMAND ===>                                                      TGT --- KMZ1

Group Name        ===> KMZ1
Command Prefix    ===> CSMKM___________

Rule information:
Ruleset Dataset   ===> BAOKMZ.AAO62.KMZ1.UBBPARM____________________
Ruleset Id        ===> G2      Rule Prefix ===> ACM1     Suffix ===> 0001

Object Statistics:
Audit             ===> Y (Y/N)
Dataset Name      ===> BAOKMZ.KMZ1.OFFLOAD__________________________
Disposition       ===> MOD

Application information:
Author            ===> BAOKMZ2
Description       ===> Group List for SYSD Again______

Last Built    by BAOKMZ2  on 07/07/2000 at 12:48:52
Last Modified by BAOKMZ2  on 06/23/2000 at 11:43:33

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

Figure 223. Command Prefix Settings for Group KMZ1

Note that the command prefix specified for this group is CSMKM. Therefore, any commands to group KMZ1 via the interface must be prefixed with CSMKM.

In this example, where a CSM-enabled 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 24. 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 DEP NODEP
CANCEL	Cancel execution of an object	None	None
BOUNCE	Stop 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 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 will be 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 it’s 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 object 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 it’s 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 24 on page 336, 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 224 shows an example of the Global Overview panel for CSM. Refer to “What the CSM Global Overview Panel Is” on page 249 for more information about this panel.

```
BMC Software ------------------ Global Overview ------------------ 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                                                         NORM
___ 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
```

Figure 224. Global Overview Panel

To view additional information about each object, scroll left by pressing PF10/PF22 (see Figure 225)

```
BMC Software ------------ Global Overview - Desc/Stats ----------- 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                     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
___ 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
```

Figure 225. Global Overview Panel: Scrolled LEFT

Field Descriptions

Table 25 describes the fields for the CSM Global Overview panel..

Table 25. Field Descriptions: CSM Global Overview Panel

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	MVS SYSNAME where the group is enabled. If the group is disabled, this column will be blank. For objects that exist in another repository or a pre 6.1.0 system, this field contains the value REMOTE. This 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 327 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 327 for more information about object states.

Table 25. Field Descriptions: CSM Global Overview Panel (Continued)

Field	Description
Ind	Indicates object relationships. Possible values are: **L** Indicates the object has clients. **N** Indicates the object has children (dependents). **X** Indicates 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 e(X)ec line commands next to the object in the LC column.
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	Date the last time the object was modified.
Time	Time the last time the object was modified.

Primary Commands

Table 26 on page 340 describes the primary commands for the CSM Global Overview panel.

Table 26. Primary Commands: CSM Global Overview Panel

Command	Description
Exception|EXC	Causes the CSM Exceptions panel to be displayed. The (MON)itor and (EXC)eption primary commands toggle the display between the CSM Global Overview panel and the CSM Exceptions panel. On the CSM Global Overview panel, the (EXC)eption command is shown. On the CSM Exceptions panel, the (MON)itor command is shown.
Group	Displays the Object Groups application, where you can manage CSM Groups.
Sort|SORT	Sorts data in (up to) two columns of the CSM display in either ascending or descending order. The default sort sequence is ascending alphabetical order based on the data in the Object column.

Table 26. Primary Commands: CSM Global Overview Panel (Continued)

Command	Description
Save|SAVE	Saves changes but does not activate them until CSM has been reinitialized or a BUILD command is issued. SAVE **does not** make changes effective immediately. Issue the BUILD command for changes to be saved and activated.
Add|ADD	Use this primary command to add new objects to be controlled by CSM. The added object automatically assumes that MVS is the parent. Enter ADD in the `COMMAND====>` area of the CSM panel when defining new objects. ADD takes effect after you issue the BUILD command.
Calendar|CAL	Invokes the Global Calendar Override panel from which you can define special schedules for one-time-only situations.

Cancel|CAN	Causes CSM to issue the (CAN)cel command for the selected object.
	
	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.
	
	Before canceling the object, CSM prompts you for verification:
	
	`Please confirm Cancel for object`
	`Cancel?? N (Y/N)`
	
	This command takes effect immediately.

Unsort|UN	Causes CSM to return the display to the default sequence of ascending object names. The UNSort has the same effect as specifying: `COMMAND ===> SORT OBJECT A` The UNsort command refreshes the CSM display; it **does not** affect the database.
CONVERT	Used to convert CSM database from AutoOPERATOR release 4.1 or 5.1 into a new Group with objects in the CSM repository.
Build|BU	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. This command requires the groupname parameter. When you issue this command, a confirmation panel appears requesting verification. BUILD makes changes effective immediately.
Locate|L	Locates a specified object on the CSM display. **Format**: `L object name`

Table 26. Primary Commands: CSM Global Overview Panel (Continued)

Command	Description
CMDSHOW	Allows 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.
Xref|X	Invokes the CSM Diagnostic Cross-Reference facility to create a cross-reference of the CSM database. The cross-reference report is written to the data set specified from which the report can be browsed or printed. To obtain a printed copy of the report: 1. Change the default value `Y` for Browse to `N` 2. Specify the SYSOUT class; for example: – Data Set Name==>`'ABC.CSMXREF.RS31'` – Browse report after generation (Y/N)? `N` – SYSOUT class for printed report? `R` 3. Press END to generate and/or print the cross-reference report.
SAVERULE	Requires Group name as a parameter. The command generates the CSM Ruleset which is defined for the specified Group.
SHUTSYS	Shuts down all the objects in a Group with a single command. The command requires that a group name is specified and can have an object as an optional parameter. The default for object is MVS, therefore if object is not specified, all objects from the Group are stopped. If an object, other than MVS is specified, the shutdown occurs for this object and all of it's children (dependents). **Format** SHUTSYS <groupid> <object> **Example** SHUTSYS PROD - CSM will shutdown all objects in group PROD SHUTSYS PROD TSO - CSM will shutdown TSO and all it's children

Table 26. Primary Commands: CSM Global Overview Panel (Continued)

Command	Description
Exceptions|EXC	When issued on the CSM Global Overview panel, causes the CSM Exceptions panel to be displayed. The (EXC)eption and (MON)itor primary commands toggle the display between the CSM Global Overview panel and the CSM Exceptions panel. On the CSM Global Overview panel, the (EXC)eption command is shown. On the CSM Exceptions panel, the (MON)itor command is shown.
Monitor|MON	When issued on the CSM Exceptions display, causes the CSM Global Overview panel to be displayed. The (EXC)eption and (MON)itor primary commands toggle the display between the CSM Global Overview panel and the CSM Exceptions panel. On the CSM Global Overview panel, the (EXC)eption command is shown. On the CSM Exceptions panel, the (MON)itor command is shown.

Line Commands

Table 27 describes the line commands for the CSM Global Overview panel.

Table 27. Line Commands: CSM Global Overview Panel

Command	Description

Start|S	Starts an object.
	
	CSM uses this command or the EXEC specified for this object.
	
	Before starting this object, CSM prompts you with three verification fields:
	
	`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)`
	
	If an object must start without starting its parents, specify `N` in the second verification field.
	
	If TSO/E and ISPF are not active and an object must be started, use the CSM EXEC, CSMACT.
	
	If the Start or Stop command needs to be issued in response to an outstanding WTOR, you can use the BMC Software-supplied EXEC CSMWTOR. The CSMWTOR EXEC works only in a JES2 environment. This EXEC has two parameters: `object name` and `reply`. The syntax is:
	
	`%CSMWTOR object_name reply_text`
	
	CSMWTOR can be defined in the Command Specifications panel as the Start or Stop command.
	
	This EXEC locates the outstanding reply from the object and responds to it with the reply text you specified.
	
	This command takes effect immediately.

Bounce|O	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. 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. The b(O)unce command affects only the selected object. It does not affect the object's parents or children. This command takes effect immediately.

Table 27. Line Commands: CSM Global Overview Panel (Continued)

Command	Description

Reset|T	Causes CSM to restore the object to its defined schedule.
	
	You might use this command if you have used the (S)tart or sto(P) line commands to start and stop objects.
	
	Before resetting this object, CSM prompts for the following verifications:
	
	`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)`
	
	The third verification field provides you three options:
	
	S Returns object to its defined schedule.
	P Returns object and its parents to their defined schedules.
	C Returns object and its children to their defined schedules.
	
	The rese(T) line command clears the status of an object in a manual state and returns it to CSM's control.
	
	To rese(T) all objects to their normal schedules, reset MVS and specify **C** as the scope, and rese(T) takes effect immediately.

Dependents|N	Displays the objects defined as being dependents (children) of the selected object. 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. If the selected object has no dependents defined, the Dependents of *object* panel contains no data.
Add|A	Add objects to the CSM database. 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. This command invokes the Object Detail Control panel. (A)dd takes effect after you issue the BUILD command.

Table 27. Line Commands: CSM Global Overview Panel (Continued)

Command	Description

Stop|P	Stops an object from the CSM application.
	
	CSM uses this command or the EXEC specified for this object.
	
	Before stopping this object, CSM prompts for the following verifications:
	
	`Please confirm STOP for object`
	`? N (Y/N)`
	`STOP all DEPENDENTS object`
	`? Y (Y/N)`
	`STOP now, but START as scheduled:`
	`? N (Y/N)`
	
	If a parent object must stop without stopping its children, specify `N` in the second verification field.
	
	If TSO/E and ISPF are not active and an object must be stopped, CSMACT should be used.
	
	If the Start or Stop command needs to be issued in response to an outstanding WTOR, you can use the BMC Software-supplied EXEC CSMWTOR. The CSMWTOR EXEC works only in a JES2 environment. This EXEC has two parameters: `object name` and `reply` and the syntax is:
	
	`%CSMWTOR object_name reply_text`
	
	CSMWTOR can be defined in the Command Specifications panel as the Start or Stop command.
	
	This EXEC locates the outstanding reply from the object and responds to it with the reply text you specified.
	
	This command takes effect immediately.

Exec|EX	Specifies that CSM will schedule a predefined Alert follow up EXEC for the object on the subsystem indicated in the target field. The E(X)ec command takes effect immediately.
Edit|E	Modifies the current control parameters for the selected object. • When you issue this command next to the MVS object, the Default Control Parameters panel is displayed. • 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.

Table 27. Line Commands: CSM Global Overview Panel (Continued)

Command	Description
Clients|L	Shows the clients of the selected object. 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). Clients continue to operate while the server is unavailable but might operate in a partially degraded state. If the selected object has no clients defined to CSM, the Clients of *object* panel contains no data.
Delete|DE	Eliminates the object from CSM's control. The selected object must be managed through some other means after this command is issued. If you attempt to (DE)lete an object which has children, CSM does not allow deletion of the object. Do one of the following to delete a parent object: • Delete the child objects and then delete the parent object. • Redefine the parent-child dependency relationship to be a client-server relationship. • Define the child objects to another parent. (DE)lete modifications take effect after you issue the BUILD command. If you delete an object by mistake, issue the CANCEL primary command.

Cancel|CAN	Causes CSM to issue the (CAN)cel command for the selected object.
	
	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.
	
	Before canceling the object, CSM prompts you for verification:
	
	`Please confirm Cancel for object`
	`Cancel?? N (Y/N)`
	
	This command takes effect immediately.

Move|M	Allows you to move CSM objects from the local BBI-SS PAS to a remote BBI-SS PAS where CSM is installed. For more information about moving objects, refer to "Moving an Object: Using the Moveable Attribute" on page 323.

Table 27. Line Commands: CSM Global Overview Panel (Continued)

Command	Description
Repeat|RE	Allows you to duplicate an existing object's definition to create a new object with the same information. Use this when you want to define objects with similar attributes. The only limitation is that no two objects can have the same name. Refer to "Using the Repeat Line Command to Create Objects" on page 279 for more information about using the Repeat line command.
Browse|B	You can browse the control parameters for the selected object. The control parameters cannot be modified with this line command. The (B)rowse command displays the Object Detail Control panel.
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 **does not** change the status of objects that are in a manual state.

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:

1. Object Detail Control
2. Command Specification
3. Conditional Commands
4. Event Specification 1
5. Event Specification 2
6. User Notification
7. Object Groups
8. Object Prerequisites
9. Object Scheduling
10. Client/Server Relationships

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.

```
BMC Software ------------ Object Detail Control - Add ------------ 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
```

Figure 226. Object Detail Control Panel Description

Field Descriptions

Table 28 describes the fields for the Object Detail Control panel.

Table 28. Field Descriptions: Object Detail Control Panel

Field	Description
Object Name	Specify 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. **Format** 8-character, alphanumeric **Default** None
Object Type	The type of the object, either Normal or Transient. 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. **Format**: NORM or TRAN. Default is NORM
Verify Force Down	Prevents accidental stopping of the object by requiring command verification. **Format**: `Y` or `N` **Y** CSM automatically asks for the stop request to be verified. You must reply to ALERT ACM705A before CSM will stop the object. **N** CSM stops the object without confirmation.

Table 28. Field Descriptions: Object Detail Control Panel (Continued)

Field	Description
Restart Only Control	Specifies whether or not CSM waits for the scheduling product to start the object. 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. When `YES` 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. Once started by the scheduling product, CSM manages the availability of the object by automatically restarting the object if it ends out of schedule. 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. **Format**: `Y` or `N` **Y** CSM manages only restart functions. **N** CSM starts, stops, and restarts the object. Objects that are defined as Restart Only can be manually started or stopped one of the following ways: • By using the MVS START or STOP command • By issuing the UP or DOWN command with the CSMACT EXEC • By changing the object's definition with the object definitions panels.
Author	TSO ID of the person who modified the object.
Description	Any free-form text you choose to describe this object. 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. **Format**: 1 to 40 alphanumeric characters of free-form text. The text should not contain quote marks. **Example**: `IRLM for IMSPROD1 on SYSB` The description is not used for any object management function.

Commands

You can issue the following commands on the COMMAND line from the Object Detail Control panel or in any of the panels listed to display another panel.

Table 29. Shortcut Commands: Object Detail Control Panel

Command	Displays the Panel....
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.

```
BMC Software ------ Command Specifications for object ----------- 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
```

Figure 227. Command Specification Panel Description

Field Descriptions

Table 30 describes the fields for the Command Specifications panel.

Table 30. Command Specification Panel Fields

Field	Description

Start Command/EXEC	This command or EXEC starts this object. If you specify a command, enter the same command used to start the object from the MVS console. Prefix the Start command with the appropriate command recognition character. For example, use a pound sign (#) as a prefix for MVS commands:
	
	`#S JES2`
	
	If the command has any embedded quotation marks, you must use two single quotation marks on each side of the command specified in the Start Command/EXEC field; for example:
	
	`#S ACF,PARM=''data''`
	
	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:
	
	`STARTIMS PROD1 WARM`
	
	When the EXEC STARTIMS is scheduled, it will be passed the two parameters `PROD1` and `WARM`.
	
	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 AutoOPERATOR subsystem address space.

Note: A Start Command/EXEC is not required for transient objects.	
User Post Start EXEC	This is the name of an optional, user-specified EXEC that CSM will schedule when CSM receives an object's start message. When CSM receives the object's start message, the parameters CSM passes to the User Post Start EXEC include: • Post Start EXEC name • Object name • Object status (normal, forced, or abnormal up/down) • Start message text-ID • Remaining words in start message The User Post Start EXEC must reside in a library in the SYSPROC concatenation of the AutoOPERATOR subsystem address space.

Table 30. Command Specification Panel Fields (Continued)

Field	Description
Stop Command/EXEC	This command or EXEC stops this object normally. If you specify a command, enter the same command used to end the object from the MVS console. Prefix the Stop command with the appropriate command recognition character. For example, use a pound sign (#) as a prefix for MVS commands: `#F CICSPAY,CEMT P SHUT` 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. 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. 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 AutoOPERATOR subsystem address space. The Cancel command/EXEC is automatically issued if the Stop command/EXEC times out. See below.
Note: A Stop Command/EXEC is not required for transient objects.	
User Post Stop EXEC	This is the name of an optional, user-specified EXEC that CSM will schedule when CSM receives an object's stop message. 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. 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 AutoOPERATOR subsystem address space.

Table 30. Command Specification Panel Fields (Continued)

Field	Description
Cancel Command/EXEC	The command or EXEC used to cancel this object. If you specify a command, enter the same command used to cancel the object from the MVS console. Prefix the Cancel command with the appropriate command recognition character. For example, use a pound sign (#) as a prefix for MVS commands: `#Z NET,CANCEL` If the command has any embedded quotation marks, you must use two single quotation marks on each side of the command specified in the Cancel Command/EXEC field. 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. The Cancel EXEC must reside in a library in the SYSPROC concatenation of the AutoOPERATOR subsystem address space. The Cancel command/EXEC is automatically issued if the Stop command/EXEC times out.
Recovery Command/EXEC	This is the name of an optional, user-specified EXEC that CSM will schedule when CSM receives an object's abnormal termination message. 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 Recovery Command/EXEC field. When CSM receives the object's abnormal termination message, the parameters CSM passes to the Recovery Command/EXEC include: • Recovery EXEC name • Object name • Object status (which is ABENDDOWN) • Abnormal termination message text-ID • Remaining words in the message The Recovery EXEC must reside in a library in the SYSPROC concatenation of the AutoOPERATOR subsystem address space.
Start Command Time Out	CSM provides automatic verification of Start commands. 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. 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. **Format**: *mmmm*, where *m*=minutes **Range**: 1 through 1440 minutes

Table 30. Command Specification Panel Fields (Continued)

Field	Description
Stop Command Time Out	CSM provides automatic verification of Stop commands. 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. **Format**: *mmmm*, where *m*=minutes **Range**: 1 through 1440 minutes 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.
Start Command Limit	Specifies how many times during a 24-hour period that CSM will automatically issue the start commands for an object. 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. The count of retries is reset when the object stops normally, at midnight every night, and when you issue the line command `Reset`. For example: • If you specify 0, CSM does not ever attempt to start the object. • If you specify 1, CSM attempts to start the object once with no retries. • If you specify 4, CSM attempts to start the object once and then tries three more times if the object terminates unexpectedly. If restarting fails at any attempt, an ALERT is issued. 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. **Format**: *xxxx* for 4 positions **Range**: 0 to 9999

Table 30. Command Specification Panel Fields (Continued)

Field	Description
Start Limit Reset	To prevent an abnormally terminating object from endlessly restarting and terminating CSM holds a Start Command Limit for the object. 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. Format: Y or N YES - 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. NO - Do not reset the command count for this object at midnight.
Conditional Start	Sometimes an object may require more than one command in order to be started or a user may want to issue an additional command if the object does not start within a prescribed period of time. Conditional commands allow users to issue additional commands on behalf of an object until all of the specified commands are scheduled or the object's Start event is recognized. Format: Y or N YES - Always user to specify Conditional Start commands. NO - Do not allow user to specify Conditional Start commands.
Conditional Stop	Sometimes an object may require more than one command in order to be stopped or a user may want to issue an additional command if the object does not stop within a prescribed period of time. Conditional commands allow users to issue additional commands on behalf of an object until all of the specified commands are scheduled or the object's stop event is recognized. Format: Y or N YES - Always user to specify Conditional Stop commands. NO - Do not allow user to specify Conditional Stop commands

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 353), the Conditional Commands Specification panel is displayed. Figure 228 shows the panel where you specify conditional Start commands.

```
BMC Software --------- Conditional Command Specification --------- 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 228. Conditional Command Specifications Panel: Start

Figure 229 shows an example of the Stop commands panel.

```
BMC Software --------- Conditional Command Specification --------- 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
```

Figure 229. Conditional Command Specifications Panel: Stop

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 31 describes the fields for the Conditional Commands Specifications panels.

Table 31. Object Groups Panel Fields

Field	Description
After ____ minutes	Enter a number of minutes to elapse before the Conditional Commands should be used.
Start Command/ EXEC	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 If this is an MVS command, the command must be preceded by the MVS command recognition character (#). Always use the pound sign for MVS commands. Maximum number of characters for commands submitted to MVS is 126. 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 AutoOPERATOR subsystem address space.
Stop Command/ EXEC	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. For example, you can enter a command: #P IMSCNTL1 If this is an MVS command, the command must be preceded by the MVS command recognition character (#). Always use the pound sign for MVS commands. Maximum number of characters for commands submitted to MVS is 126. 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 AutoOPERATOR subsystem address space.

Event Specifications 1 Panel

Pressing Enter from the Command Specifications panel displays the Event Specifications 1 panel.

```
BMC Software ------- Event Specifications 1 for object ----------- 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
```

Figure 230. Event Specifications 1 Panel Example

Field Descriptions

Table 32 describes the fields for the Event Specifications 1 panel.

Table 32. Event Specifications 1 Panel Fields

Field	Description
Generate	Specifies whether CSM automatically generates Start and Stop Rules for CSM objects. Possible values are as follows: **Yes** CSM will automatically generate Start and Stop Rules. **No** CSM will not automatically generate Start and Stop Rules. 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. 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. 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.
Start Events: Text-ID	Specifies the ID of the message that indicates the object has been started. 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. When this message is issued by an object, the Rules Processor schedules EXEC CSMUP to inform CSM of the object's state. **Format**: 1 to 16 characters **Example**: $HASP373 or IEF403I
Type	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. Must be one of the following event types: MSG, CMD, ALRT, JRNL 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 Jobname equal to the object name.
Job Type	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. Must be one of the following job types: STC, JOB, TSO

Table 32. Event Specifications 1 Panel Fields (Continued)

Field	Description
Job Name	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. Must be a valid jobname. Rules Processor wildcard characters are permitted.

Table 32. Event Specifications 1 Panel Fields (Continued)

Field	Description
Stop Events Text-ID	Specifies the ID of the message that indicates the object has been stopped. 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. When this message is issued by an object, the Rules Processor schedules EXEC CSMDOWN to inform CSM of the object's state. **Format**: 1 to 16 characters **Example**: $HASP395 or IEF404I 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. For example, the expected stop messages may not be produced if the MVS FORCE or SYSPROG EXEC commands are used to terminate an object. The End-of-Memory EXEC CSMEOM allows CSM to monitor an object stopping regardless of how the object stops. Because AutoOPERATOR allows only one End-of-Memory EXEC in each AutoOPERATOR subsystem, the CSM End-of-Memory EXEC is designed to be called from the EXEC, IMFEOM. **Parameters passed to the EXEC** Two parameters are passed to the CSM End-of-Memory EXEC: • The address space ID of the terminated address space • A character string which can have one of two values: **Parameter Value** — **Description** **NORMAL** — Indicates normal address space termination **ABNORMAL** — Indicates address space was terminated by passing it to RTM This may happen when using the SYSPROG EXIT command or the MVS FORCE command. This is not an indication that the address space abended with a system or user abend code. If the installation already uses an IMFEOM EXEC, you can add a call to the CSMEOM EXEC by adding: `IMFEXEC SELECT(CSMEOM IMFOASID EOMSTAT)` to the IMFEOM EXEC. 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.

Event Specifications 2 Panel

Pressing Enter from the Event Specifications1 panel displays the Event Specifications 2 panel.

```
BMC Software ------- Event Specifications 2 for EV1 - Add -------- 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
```

Figure 231. Event Specifications 2 Panel Example

Field Descriptions

Table 33 describes the fields for the Event Specifications 2 panel.

Table 33. Event Specifications 2 Panel Fields

Field	Description
Abnormal Termination events Text-id	Specifies the text-ID of the message that indicates the object has terminated abnormally (abended). This is also known as an abnormal termination event. 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. 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. **Format**: 1 to 16 characters **Example**: IEF450I
Job Name	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. Must be a valid jobname. Rules Processor wildcard characters are permitted.

Table 33. Event Specifications 2 Panel Fields (Continued)

Field	Description
Job Type	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. Must be one of the following job types: STC, JOB, TSO
Text String	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. 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.
Word	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. 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. Valid values are numeric, up to 4 positions.
Op	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. Valid values must be one of the following operands: EQ, LT, GT, NE
Event Type	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. Must be one of the following event types: MSG, CMD, ALRT, JRNL

User Notification Panel

Pressing Enter from the Event Specifications 2 panel displays the User Notification panel.

```
BMC Software ---------- User Notification for object ---------- 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
```

Figure 232. User Notification Panel Example

Field Descriptions

Table 34 describes the fields for the User Notification panel.

Table 34. User Notification Panel Fields

Field	Description
TSO Send User ID	CSM automatically sends a TSO message if an exceptional condition for the managed object exists. 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. 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). **Format**: `userid01` **Range**: Any valid 1- to 7-character TSO user ID
Contact	Specify in the Contact 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.
Info	The Information 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.

Table 34. User Notification Panel Fields (Continued)

Field	Description
Object Alert Queue	The name of the ALERT queue in the ALERT Management Facility that will receive ALERTs for this object from CSM. **Format**: 1 to 8 alphanumeric characters **Example**: MAIN or CSMMVS
Alert Publish Mode	Specifies whether the ALERT will be published.

Object Groups Panel

Pressing Enter from the User Notification panel displays the Object Groups panel.

```
BMC Software ------------ Object Groups object ---------------- 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 ********************************
```

Figure 233. Object Groups Panel Example

Field Descriptions

Table 35 describes the fields for the Object Groups panel.

Table 35. 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 36 describes the primary commands for the CSM Object Groups panel.

Table 36. Primary Commands: Object Groups Panel

Command	Description
Locate	Allows you to locate a specific CSM Group in the Object Groups display. This command works similar to Locate command in ISPF display.
CMDSHOW	Allows 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	Allows you to cancel your changes.

Line Commands

Table 37 describes the primary commands for the CSM Object Groups panel.

Table 37. Line Commands: Object Groups Panel

Command	Description
Remove|R	Removes 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 this is the only group which has the object as a member.
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.

```
BMC Software ---------- Object Requisites for object ------------ 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 ********************************
```

Figure 234. Object Requisites Panel Example

Field Descriptions

Table 38 describes the fields for the Object Requisites panel.

Table 38. 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 39 describes the primary commands for the CSM Object Requisites panel.

Table 39. 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	Allows 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 40 describes the primary commands for the CSM Object Requisites panel.

Table 40. 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.

```
BMC Software ----- Client/Server Relationships for object -------- AutoOPERATOR
COMMAND ===>                                                 SCROLL ===> CSR
                                                                TGT ---  NS61
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 ********************************
```

Figure 235. Client/Server Relationships Panel Example

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 41 describes the fields for the Client/Server Relationships panel.

Table 41. Client/Server Relationships Panel Fields

Field	Description
Server	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. This is the same name that is started from the MVS console or submitted as a batch job. The server name must be an object under the control of CSM. This field is modifiable.
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.

Table 41. Client/Server Relationships Panel Fields (Continued)

Field	Description

Connect Command	A command or EXEC routine to connect the client object to the server object. This is usually, though not restricted to, an MVS Modify command or a VTAM Vary command.
	
	Prefix this command with the appropriate command recognition character, or CSM reads the input as an EXEC. Use a pound sign (#) for MVS commands.
	
	Sample Modify command:
	
	`#F IMSPROD1,START`
	
	Sample VTAM Vary command:
	
	`#V NET,ACT,ID=IMSPROD1`
	
	If an EXEC name is used, it must reside in the library concatenation of the SYSPROC DD name of the AutoOPERATOR subsystem address space.

Disconnect Command	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 MVS Modify command or a VTAM Vary command.
	
	Prefix the Disconnect command with the appropriate command recognition character, or CSM reads the input as an EXEC. Use a pound sign (#) for MVS commands.
	
	Sample Modify command:
	
	`#F IMSPROD1,STOP`
	
	Sample VTAM Vary command:
	
	`#V NET,INACT,ID=IMSPROD1`
	
	If an EXEC name is used, it must reside in the library concatenation of the SYSPROC DD name of the AutoOPERATOR subsystem address space.

Primary Commands

Table 42 describes the primary commands for the CSM Client/Server Relationships panel.

Table 42. Primary Commands: Client/Server Relationships Panel

Command	Description
ADD	Enter ADD to add a new server for an object. Displays the Add Client/Server Relationships where you can add the Server name of the object Group Name of the object Connect command for the object Disconnect command for the object

Table 42. Primary Commands: Client/Server Relationships Panel (Continued)

Command	Description
END	Enter END to return to the Object Detail Control panel.
CMDSHOW	Allows 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 43 describes the primary commands for the CSM Client/Server Relationships panel.

Table 43. Line Commands: Client/Server Relationships Panel

Command	Description
Edit|E	Displays the Edit Client/Server Relationships panel. On this panel, edit the Server name of the object Group Name of the object Connect command for the object Disconnect command for the object
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.

```
BMC Software ---------- Object Scheduling for object ----------- 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 ********************************
```

Figure 236. Object Scheduling Panel Example

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 44 describes the fields for the Object Scheduling panel.

Table 44. 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	The time the object is scheduled to come down.
	
	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 pm on Sunday night to 2:00 am on Monday morning, the scheduled down time for that object would be specified as:
	
	`SUNDAY  20:00  24:00`
	
	`MONDAY  00:00  02:00`
	
	Format: *hh:mm*, where *hh*=hours and *mm*=minutes
	
	Range: 00:00 to 24:00
	
	The 00:00 is the beginning of the day; the 24:00 is the end of the day.

Table 44. Object Scheduling Panel Fields (Continued)

Field	Description
Down To	The time the object starts after a scheduled down time. CSM uses the 24-hour clock for managing down times. **Format**: *hh:mm*, where *hh*=hours and *mm*=minutes **Range**: 00:00 to 24:00 The 00:00 is the beginning of the day; the 24:00 is the end of the day.

Primary Commands

Table 42 describes the primary commands for the CSM Object Scheduling panel.

Table 45. Primary Commands: Object Scheduling Panel

Command	Description
ADD	Use to add a scheduled DOWN time. Displays the Add Object Scheduling panel where you can add • The name for the group for which this schedule is created. The object must be associated with this group. • The days of the week the object is supposed to be down – 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. – 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). – Saturday, Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Weekday, Weekend, and Daily • The Down From and Down To times (in HH:MM format)
END	Enter END to return to the Object Detail Control panel.
CMDSHOW	Allows 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 46 describes the primary commands for the CSM Object Scheduling panel.

Table 46. Line Commands: Object Scheduling Panel

Command	Description
Edit|E	Displays the Edit Object Scheduling panel. On this panel, edit • The name for the group for which this schedule is created. The object must be associated with this group. • The days of the week the object is supposed to be down – 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. – 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). – Saturday, Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Weekday, Weekend, and Daily. • The Down From and Down To times (in HH:MM format)
Delete|D	Use to delete a scheduled DOWN time.

Global Calendar Override Panel

CSM allows 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 237 is displayed).

```
BMC Software ------------- Global Calendar Override -------------- 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 ******************************
```

Figure 237. Global Calendar Override Panel Example

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 47 describes the fields for the Global Calendar Override panel.

Table 47. Global Calendar Override Panel

Field	Description
Object	Object name to which this special schedule override applies. 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. When the object is returned to operation, the dependent objects are restarted and the client objects are connected to the server object automatically. **Format**: `objname`
Group/SYSNAME	Group and MVS SYSNAME where this override is supposed to occur.
Calendar Date	Specific date for which the special schedule applies. 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. **Format**: 8 characters *mm/dd/yy*, where *mm*=month, *dd*=day, *yy*=year

Down From	Time CSM begins the process of stopping the selected object and its dependents.
	
	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.
	
	For example, if an IMS Control Region object is scheduled to be DOWN from 8:00 pm on Sunday night to 2:00 am on Monday morning, the scheduled down time for that object would be specified as:
	
	`SUNDAY 22:00 24:00`
	
	`MONDAY 00:00 02:00`
	
	Format: *hh:mm*, where *hh*=hours and *mm*=minutes
	
	Range: 00:00 to 24:00
	
	The 00:00 is the beginning of the day; the 24:00 is the end of the day.

Table 47. Global Calendar Override Panel (Continued)

Field	Description
Down To	Time CSM attempts to start the object, except in the case of MVS. If MVS is not active, then AutoOPERATOR and CSM are not active. However, when AutoOPERATOR and CSM are initialized, CSM automatically starts all the objects that should be active at that time. **Format**: *hh:mm*, where *hh*=hours and *mm*=minutes **Range**: 00:00 to 24:00 The 00:00 value is the beginning of the day; the 24:00 value is the end of the day.
Reason for Down Time	Lets you document the reason for the special down time for the affected objects. **Format**: 1 to 21 positions of free-form text
Contact	Group or individual responsible for scheduling the special down time for the affected objects. **Format**: 1 to 16 positions of free-form text

Primary Command

Table 48 describes the primary commands for the Global Calendar Override panel.

Table 48. Global Calendar Override Primary Commands

Command	Description
ADD	Add scheduled down times for the selected objects. Type ADD in the `COMMAND===>` 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. ADD takes effect after you issue the BUILD command.
CANcel	Return to the screen without processing changes. Type CANcel in the `COMMAND===>` area of the display to cancel any changes and return to the screen.
END	Process changes and return to the screen. Type END in the `COMMAND===>` area of the display to process your changes and return to the screen.
CMDSHOW	Allows 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 49 describes the line commands for the Global Calendar Override panel.

Table 49. Global Calendar Override Line Commands

Command	Description
Insert Time|I	Add an additional object and special schedule to the Global Calendar Override panel. To insert a new schedule for an object, type the line command in the line command `(LC)` column and press ENTER. 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. (I)nsert modifications take effect after you issue the BUILD command.
Delete Time|D	Remove the special schedule for an object. To use this line command, use the tab key to position the cursor in the line command `(LC)` column for the desired object and type `D.` Press ENTER to remove the special schedule for the selected object. If you delete a schedule by mistake, issue the CANCEL primary command. (D)elete modifications take effect after you issue the BUILD command.

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 1- to 8-characters long but it is recommended that you name the first group the SSID of the subsystem that you bring CSM up on.

Figure 238 shows an example of the Object Groups panel.

```
BMC Software ------------------- Object Groups ------------------- 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
     ________    ________    ____    ________   ______________________________
```

Figure 238. Object Groups Panel Example

Field Descriptions

Table 50 describes the fields for the Object Groups panel.

Table 50. 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 MVS 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 51 describes the primary commands for the CSM Object Requisites panel.

Table 51. Primary Commands: Object Requisites Panel

Command	Description
ADD	Allows you to add a new Group to the current CSM repository. CSM will display the Group Detail Control panel where you can specify all Group definitions (refer to Figure 239). The ADD Primary command can be entered by itself, or with a Group name; for example: ADD GROUP123 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.
Locate	Allows you to locate a specific CSM Group in the Object Groups display. This command works similar to Locate command in ISPF display.
CMDSHOW	Allows 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.
BUILD	Allows you to issue Build against CSM Group. Build saves all changes made online and activates them immediately. The BUILD command requires parameter of Group Name.

Line Commands

Table 52 describes the primary commands for the CSM Object Requisites panel.

Table 52. Line Commands: Object Requisites Panel

Command	Description
Build|BU	Allows 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 **does not** change the status of objects that are in a manual state.

Table 52. Line Commands: Object Requisites Panel (Continued)

Command	Description
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 CSM's control.
List|L	Display all objects, associated with the selected group.
Repeat|R	Allows 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	You can 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 239 shows an example of the Group Detail Control panel.

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

Figure 239. Group Detail Control Example

Field Descriptions

Table 53 describes the fields for the Group Detail Control panel.

Table 53. Group Detail Control Panel Fields

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 CSM's 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 allow 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 AutoOPERATOR variables.
Rule Dataset	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 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 Dataset Name in this field. For proper operation, the data set specified in this entry must be in the BBIPARM DD concatenation of the AutoOPERATOR BBI-PAS.
Ruleset ID	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. The Ruleset identified by this suffix must be present in one of the data sets in the BBIPARM concatenation in AutoOPERATOR's BBI-PAS.
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	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 Statistics Dataset Name field. 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 Statistics Dataset Name field. **Note:** The CSM object statistics dataset should be a fixed block sequential data set with the maximum record length of 80 bytes.

Table 53. Group Detail Control Panel Fields (Continued)

Field	Description
Disposition	Used in conjunction with the Dataset Name 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.

Part 4. Using Other Basic Automation Applications

This part contains chapters that describe:

- ALERTs application
- Automation Reporter application
- EXEC Management application
- General service applications

Chapter 12. Managing by Exception: Using AutoOPERATOR ALERTs 391
Introduction 391
Overview to the ALERT Management Facility 393
What ALERTs Are 393
How ALERTs Are Kept 394
Where ALERTs Are Stored. 394
How to Display ALERTs 394
What ALERTs Can Do 394
Where to Go from Here. 395
Obtaining Statistical ALERT Information: ALERT Queue Statistics Application 396
Panel Description 397
Displaying ALERT Data Graphically: ALERT Overview Application 399
Using a Profile. 402
Showing ALERTs in Text Format: Using the ALERT Detail Application 404
Using a Profile. 408

Chapter 13. Monitoring Automation with the Automation Reporter 415
Introduction: What the Automation Reporter Application Does 415
How to Activate the Automation Reporter. 416
Activating the Automation Reporter with the Dynamic Parameter Manager 417
Step 1: Accessing the Automation Reporter Parms Panel in the DPM Application. . . . 417
Step 2: Allocate a Data Set 419
Step 3: Specifying the Data Set Disposition 420
Step 4: Specifying Collection Intervals. 421
Step 5: Specifying the Offload Interval 422
Step 6: Specifying an AutoOPERATOR ALERT Queue Name 423
Step 7: Activating the Automation Reporter. 424
Stopping the Automation Reporter 426
Creating Automation Reporter Reports 427
What Kind of Data Is Collected 427
How the Data Is Saved 429
What the Record Looks Like: Basic Data Record Design 431
Describing a Sample Data Record 434
What Record Types Are Available 435
Tracking Actions (ACTN). 437
Tracking AutoOPERATOR ALERTS (ALRT) 443
Tracking Events (EVENT) 445
Tracking AutoOPERATOR EXECs (EXEC) 446
Tracking AutoOPERATOR Rules Activity (RULE). 447
Tracking AutoOPERATOR Rule Sets (RULESET: AUTO). 448
Tracking AutoOPERATOR Rule Sets (RULESET: SET). 449
Tracking Resources (RES) 451

Tracking Tape Device Sharing Activity (TAPE: LOCAL) 453
Tracking Tape Device Sharing Activity (TAPE: PLEX) 455

Chapter 14. Managing EXECs Using the EXEC Management Application 457
Introduction 457
Accessing the EXEC Management Application 458
Scrolling Right to Display Additional Information 459
Viewing All Currently Active EXECs: Using the EXPAND Command 459
Using Primary Commands 460
Describing Fields in the EXEC Status Area 460
Describing Fields in the EXEC Description Area 462
Displaying Currently Active EXECs 466
Using Line Commands 466
Describing Fields 467
Documenting Your EXECs 468

Chapter 15. General Services 469
Creating Cyclic Displays: CYCLE SETUP 469
Specifying Application Names in CYCLE SETUP 471
Saving a Refresh Cycle 477
Viewing Current Operational Events: Log Display 480
Displaying System Messages: MESSAGES 483
Using PF Keys: KEYS 483
Requesting Time-Initiated EXECs: TIMEXEC Application 484
Describing the TIMEXEC Application 485
Purging an Active CALLX Request 489

Chapter 12. Managing by Exception: Using AutoOPERATOR ALERTs

This chapter provides:

- An introduction to ALERTs and your data center
- An overview to using the AutoOPERATOR ALERT Management Facility
- Discussions for using the applications in the ALERT Management Facility for:
  - Obtaining statistical ALERT information
  - Displaying ALERT data graphically
  - Showing ALERTs in text format

To create AutoOPERATOR ALERTs, you either must use the IMFEXEC ALERT statement in CLISTs or REXX EXECs, or AutoOPERATOR Rules that generate ALERTs in response to automation events occurring in the system. For information about how to create ALERTs, refer to Table 54 on page 393.

Introduction

The following sections discuss general topics such as:

- What are some problems in message management
- How AutoOPERATOR has solutions to these problems by providing:
  - Automation and ALERT generation with the AutoOPERATOR Rules Processor
  - ALERT management features in the ALERT Management Facility

What Are the Problems?

Today's data center operations personnel are inundated with information from MVS consoles, IMS logs, CICS system transient data (TD) queues, various system monitors, and so on. Messages also arrive to different physical locations. Of all these messages, only a small percentage might require operator attention and intervention.

Solutions

Following are some possible solutions.

Message Suppression: One of the first things you can do is implement message suppression with the AutoOPERATOR Rules Processor applications. By lessening the overall number of messages, you would allow more important messages to be more easily seen.

Console Consolidation: Console consolidation (reducing the number of physical terminals associated with the information providers) is the next step. Potentially, all sources could route important messages to a single terminal and utilize the AutoOPERATOR Logging Facility. However, operations personnel often are responsible for different areas of the

systems. For example, Network, DASD, MVS, or subsystem operators generally deal with issues that are specific to a particular area.

A better situation is one where message traffic from all the different sources are categorized and routed to the appropriate personnel. The AutoOPERATOR ALERT Management Facility allows you to do this.

Management By Exception: With AutoOPERATOR ALERTs and the ALERT Management Facility, you can implement something that is commonly referred to as "Management by Exception".

The ALERT Management Facility helps you to:

- Filter the messages originating from the many sources
- Classify and categorize information with priorities
- Create subclasses of information for those individuals handling the messages

The ALERT Management Facility provides the following features:

- Message interception and classification
- Generation of ALERTs containing the original (or reworded text) associated with routing and priority characteristics
- Flexible display facilities that allow controlled subsets of this information to be displayed

The ALERT Management Facility also provides ALERTs that continue to exist across BBI-SS PAS restarts and MVS IPLs or can be customized to escalate in severity over time.

Overview to the ALERT Management Facility

The following sections contain discussions that introduce important concepts about AutoOPERATOR ALERTs and the ALERT Management Facility.

What ALERTs Are

An AutoOPERATOR ALERT consists of a number of components both textual (visible) and user-specified (machine accessible). AutoOPERATOR ALERTs are messages that AutoOPERATOR creates to notify operators of exceptional situations that might require manual intervention and attention.

AutoOPERATOR ALERTs can originate from:

- The AutoOPERATOR Rules Processor
- AutoOPERATOR CLIST or REXX EXECs
- NetView address space issuing the NAIEXEC ALERT command

The way an ALERT looks and functions depends on what you specify when the ALERT is created from any of these sources.

For documentation about how ALERTs are generated from these sources, refer to the following table.

Table 54. Where to Find More Information about ALERTs

To read about...	**See...**
Creating ALERTs based on various system events with AutoOPERATOR Rules “Creating Rules for Events: Using the Rule Creation Panels” on page 73 contains complete information about how ALERTs are created from a Rule.	Chapters 3 through 11 in this manual
Creating ALERTs from AutoOPERATOR CLIST EXECs	*MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs*
Creating ALERTs from AutoOPERATOR REXX EXECs	*MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*
Creating ALERTs from the NetView address space	“AutoOPERATOR ACCESS NV” section in the *MAINVIEW AutoOPERATOR Options User Guide*

In addition, AutoOPERATOR maintains information about:

- The date and time ALERTs originated
- Where the ALERTs originated from
- The ALERT's priority and color
- The ALERT's queuing characteristics

- Target user information
- Any user-specified data associated with a particular ALERT

How ALERTs Are Kept

ALERTs reside in **queues**. At any one time, there can be only one ALERT with a particular key in a particular queue. This means that the queue identifier, together with the ALERT key, uniquely identify an ALERT.

Regardless of the source of an ALERT, one attribute of an ALERT which must be specifically defined for each ALERT is the ALERT key. The key is the attribute which allows an ALERT to be distinguished from other ALERTs.

A new ALERT created with the same queue and key identifier replaces an existing ALERT with the same queue and key identifier.

Where ALERTs Are Stored

ALERTs are maintained in extended storage private to an AutoOPERATOR-SS but can be routed from one subsystem to another. In this case, the receiving subsystem may be classified as a focal point of the ALERTs from both systems.

How to Display ALERTs

The ALERT Management Facility provides three different applications to allow the display of ALERTs in several ways:

- The ALERT Overview Statistics application shows a general breakdown of all ALERT queues with the corresponding number of ALERTs by priority and storage usage
- The ALERT Overview application is a graphical display of the number of ALERTs in user-specified queues
- The ALERT Detail application provides the actual text content of selected ALERTs.

The panels can be modified using their profile panels which allow you to specify only the information pertinent to a particular user or user-group is shown on the panel.

What ALERTs Can Do

AutoOPERATOR ALERTs are programmable. You can create ALERTs that allow you to have automatic execution of follow-up EXECs (also known as ALERT-initiated EXECs) to take automation steps as a result of the ALERT.

You can also opt to associate additional help information with an ALERT thus allowing you to retrieve specific information about an individual problem from an ALERT.

The interaction of ALERTs with other BMC Software products, such as the various online monitors, can be tailored so that they provide additional information for the ALERT without you having to be aware of the underlying details.

Where to Go from Here

Once ALERTs are created (either from Rules, EXECs, or NetView), you can use the ALERT Management Facility, which is documented in this chapter, to manage ALERT functions with different ALERT applications.

Obtaining Statistical ALERT Information: ALERT Queue Statistics Application

This application provides information about all ALERT queues active in the AutoOPERATOR subsystem addressed by the TGT field. The main portion of the panel shows a scrollable listing of:

- The individual queues
- The number of ALERTs contained within the queues broken up by priority
- The sum of all ALERTs within each queue
- Information about the amount of storage consumed

Use the information presented here to gain an understanding of what the storage requirements for ALERTs are to fine-tune the parameters in the BBPARM member AAOALS00. For more information, refer to the chapter for the Dynamic Parameter Manager in the *MAINVIEW AutoOPERATOR Customization Guide*.

Select Option 0 from the ALERT Management Facility menu to invoke the ALERT Queue Statistics application as shown in Figure 240:

```
BMC Software ------------------- ALERT Queue Stats ------------- AutoOPERATOR
COMMAND ===>                                                TGT ===> SYSC
Interval ==> 1                                              Date --- 01/03/30
Status   --- INPUT                                          Time --- 11:22:23
(S)elect  (Del)ete
             -------------------- Counts ------------------       Total Avg.
Cmd   Queue    Clrng   Info.   Warn  Minor  Major  Crit.  Total    Size  Size
___  NETWORK     0      0       1      0      0      0      1       206  206
___  MVS         0      0       0      0      1      0      1       233  233
___  IMS         0      0       1      0      0      0      1       222  222
___  DASD        0      0       0      0      0      1      1       221  221
___  CICS        0      0       1      1      0      0      2       419  209
___  A B         1      0       0      0      0      0      1       205  205
     ***************************** END OF QUEUES *****************************
```

Figure 240. ALERT Queue Statistics Panel

This application provides an overview of all ALERT activity and is designed for use by the automation specialist who develops site-specific solutions involving ALERT management. It may also serve as a diagnostic tool and a quick shortcut when, for example, a test queue with hundreds of ALERTs needs to be cleared.

The ALERT Queue Statistics application is the only online application that gives you a complete picture of all the ALERT queues. It can provide you with information about unexpected ALERT queues, such as an ALERT queue caused by an EXEC gone awry.

Panel Description

The following describes the fields on the panel.

Field	Contents
Queue	Name of the ALERT queue that the information on this line applies to.
CLRNG	Number of ALERTs in this queue prioritized as CLEARING.
INFO	Number of ALERTs in this queue prioritized as INFORMATIONAL.
WARN	Number of ALERTs in this queue prioritized as WARNING.
MINOR	Number of ALERTs in this queue prioritized as MINOR.
MAJOR	Number of ALERTs in this queue prioritized as MAJOR.
CRIT	Number of ALERTs in this queue prioritized as CRITICAL.
TOTAL	Total number of ALERTs in this queue. For example, the sum of ALERTs in the CLRNG, INFO, WARN, MINOR, MAJOR, and CRIT queues.
TOTAL SIZE	Total amount of storage in bytes consumed by all ALERTs in this queue.
AVG. SIZE	Average size of one ALERT in this queue expressed as TOTAL SIZE / TOTAL.

Line Commands

Line commands that you can enter in the input fields are:

Command	Function
Select|S	Invokes the ALERT Detail Application displaying all ALERTs in the indicated queue
DELete|DEL	Deletes all ALERTs in the indicated queue and removes the queue itself **Note:** Deleting all ALERTs contained in a particular queue from the ALERT Detail panel does not cause the actual queue headers to be removed. The queue name itself may still be retrieved using the IMFEXEC ALERT FUNCTION(LISTQ) command. This application represents the only means to interactively remove the ALERT queue header itself.

Primary Commands

Primary commands that you can enter in the COMMAND line are:

Command	Function
Sort	Sorts the ALERT queues shown by any of the column headings in either ascending or descending sequence This command has 2 parameters: the first is required, the second is optional. • The first parameter specifies a column heading name. The following names may be used: – QUEUE – CLRNG – INFO – WARN – MINOR – MAJOR – CRITICAL – TOTAL – TOTSIZE – AVGSIZE The default is QUEUE. • The second parameter specifies the sort order, for example, Ascending (A) or Descending (D); for example: `SORT TOTSIZE A`

Displaying ALERT Data Graphically: ALERT Overview Application

The ALERT Overview application shows a graphical representation of up to 6 queues or queue groups. Use this application whenever you want to see a comparison of the number and severity of ALERTs in (up to) 6 different queues.

To display the panel, select Option 1 from the ALERT Management Menu to display the ALERT Overview application:

```
BMC Software -------------------- ALERT OVERVIEW ---------------- AutoOPERATOR
COMMAND ===>                                                TGT ===>  CHICAGO
INTERVAL ==> 2                                              DATE --- 01/03/30
STATUS   --- INPUT             Profile suffix ===> 00       TIME --- 09:33:34

        MVS         DASD      NETWORK       IMS*        CICS       MAIN
   +-------------------------------------------------------------------------+
   |  (     1 )   (     4 )   (     2 )   (     2 )   (     3 )   (     5 )   |
   |                                                                         |
   |                                                                         |
   |                                                                         |
   |                                                                         |
   |                                                                         |
   |                                                                         |
   |                                                                         |
   |                                                                         |
   |                                                                 555555  |
   |               444444                                            333333  |
   |               222222                                 444444     222222  |
   |               111111      444444      666666      444444     222222  |
   |   222222      111111      222222      444444      333333     111111  |
   +-------------------------------------------------------------------------+

       Critical-1  Major-2  Minor-3  Warning-4  Informational-5  Clearing-6
```

Figure 241. ALERT Overview Application Panel

Each one of these stacks represents a queue and is referred to as a bar graph. On terminals supporting extended attributes, these bar graphs are shown in color in reverse video.

The display can contain up to 6 bar graphs or queues. These sections are either color-coded (on terminals supporting reverse video) or represented by a number series.

How Bar Graphs Are Scaled

A line or set of lines within a bar graph represents the number of ALERTs with specific priority within the queue. Each bar graph is constructed from bottom to top with the ALERTs of the highest priority appearing on the bottom of the panel. The bar graphs are automatically scaled using the following rules:

- All bar graphs on the panel are scaled by the same factor.
- The factor is calculated as a divisor that allows the tallest bar graph (that is, the largest queue) to fit into the constraints of the physical terminal.
- However, if the divisor would cause any section (representing the number of ALERTs with a particular priority) to be dropped from the panel, the bar graph is adjusted so that the smallest section will still be displayed, possibly causing the bar graph to extend beyond the screen window and be truncated.

The last rule ensures that, even though a queue may contain a large number of ALERTs with a lesser priority and only a few with a high priority, the section related to the ALERTs with a high priority will still be visually represented.

How Colors and Numbers Are Used

The following table shows the correlation of digits and colors of the ALERTs to the priority of the ALERTs. The colors will be correctly represented only on a terminal supporting extended attributes.

Also, the colors shown here are strictly related to priorities and not the colors associated with the ALERTs themselves. ALERTs may have color attributes associated with them at creation time; these colors are shown only in the ALERT Detail application.

Color	**Digit**	**Priority**
Red	1	Critical
Pink	2	Major
Yellow	3	Minor
Dark Blue	4	Warning
Light Blue	5	Informational
Green	6	Clearing

The following fields are shown on the ALERT Overview panel:

Field	**Contents**
Command	Standard command input field for BBI global or application specific commands
Interval	Screen refresh interval when issuing GO command
Status	Indicator to show whether the application has been entered into automatic screen refresh: • Input: Application accepts input • Running: Application is in auto-refresh mode
Profile Suffix	Profile suffix used to construct panel Use the SET primary command or overtype this value to select a different configuration. To construct a new setup, simply enter a new suffix. You will be prompted for the required values.

The ALERT Overview Application processes some commands which are cursor-position sensitive. This means that by placing the cursor within the display and into the column for a queue (the exact location is not relevant), certain primary commands may be issued in relation to a particular queue.

These primary commands are assigned to a queue on the ALERT Overview Profile panel which is discussed in “Using a Profile” on page 402.

The primary commands are:

Command	Function
EXPAND	Invoke the ALERT Detail Application (refer to “Showing ALERTs in Text Format: Using the ALERT Detail Application” on page 404) showing the textual content for this ALERT queue or queue group. (Requires cursor positioning.) **Note:** The EXPAND command is designed to be assigned to program function keys and act as a hot-key rather than to be explicitly entered in the COMMAND field.
TRANSFER	Execute the primary command sequence defined in the profile for this queue. (Requires cursor positioning.) **Note:** The TRANSFER command is designed to be assigned to program function keys and act as a hot-key rather than to be explicitly entered in the COMMAND field.
SET	Selects a different or new profile to configure the panel. To construct a new profile, enter a new suffix. You will be prompted for the required values.
PROFILE	Invokes the profile setup for the suffix currently active (for example, entered in the field labeled Profile suffix). The panel shown on page 402 will be displayed.
GO	Places the application into screen refresh. Press the PA1 or Attention key to exit this mode.

Displaying Hierarchical Arrangement of Overview Applications

By using a combination of primary commands and application transfer commands, you can create a hierarchical arrangement of panels that show differing levels of ALERT queue information.

In other words, you can create one Overview panel that displays ALERTs of *several different queues in one bar graph* Then, you can have an secondary Overview panel that displays each of those different queues with its own bar graph.

A combination of the primary commands TRANSFER and SET and the application transfer command ALE allows you to accomplish this by:

- Using an initial profile and having each bar chart representing groups or categories of queues

  For example, you can have a queue named CICS. On the Overview panel, the bar graph for CICS represents a summary of several queues of ALERTs for active CICS systems at a site.

- Assigning a primary command that invokes another profile suffix under a new application level; for example, ALE;SET xx, where xx is a new profile suffix

  This expands the queue named CICS and the new panel shows the individual queues. That is, the new panel will display each of the individual queues with CICS-related ALERTs with a bar graph for each queue.

This approach allows zooming into problem areas simply by positioning the cursor into a bar of the graph and pressing a PF key that has been assigned the command string TRANSFER.

By implementing a new application level, the underlying information is saved and the summary profile will be redisplayed when you press PF3.

Using a Profile

A profile application drives the Overview application and determines which ALERT queues are displayed and which primary commands are associated with each queue.

The ALERT Overview Application must be configured using a profile before it can be used. The profile determines which ALERT queues are shown. To display the ALERT Overview Profile panel, issue the command PROFILE on the command line. An example of the panel is shown in “Overview to the ALERT Management Facility”.

```
BMC Software -------------- ALERT OVERVIEW PROFILE ------------- AutoOPERATOR
COMMAND ===>
                     Profile suffix 00                       DATE --- 01/03/08
                                                             TIME --- 09:33:54
NAME ===> MVS        PCMD ===> MAO;STATUS

NAME ===> DASD       PCMD ===>

NAME ===> NETWORK    PCMD ===>

NAME ===> IMS*       PCMD ===>

NAME ===> CICS       PCMD ===>

NAME ===> MAIN       PCMD ===>



     Press END to SAVE Profile and return to application
```

Figure 242. ALERT Overview Profile Application Panel

This panel allows the specification of up to 6 queue names or queue name patterns. Optionally, any AutoOPERATOR application transfer may be entered in the fields labeled PCMD to the right of the queue names. The specified command is executed when, in the ALERT Overview Application, the cursor is positioned into the corresponding queue's area and the TRANSFER command is issued.

As an example, you might be monitoring ALERTs pertaining to a specific CICS system. Once a problem is indicated on the ALERT Overview Display, you may use the TRANSFER feature to directly transfer into a MAINVIEW for CICS session that shows technical details involving this CICS system.

All commands that can be processed in the COMMAND field of the ALERT Overview Application when the session is run under TSO, VTAM, or EXCP mode may be specified in the PCMD field.

The information on this panel is automatically saved when you press the PF3 key. Enter CANCEL on the COMMAND field to discard any changes.

You can also specify that a command is executed on a remote system by specifying `SYSTEM` and a remote BBI-SS PAS target. For example, if you specify:

```
PCMD ===>  CICS;EX TRAN
```

the `TRAN` command will be executed against the current default target for CICS.

If you specify:

```
PCMD ===>  CICS;EX TRAN;SYSTEM CICSA
```

the `TRAN` command will be executed against the CICS target CICSA, which is the name of a remote CICS target.

Showing ALERTs in Text Format: Using the ALERT Detail Application

Selecting Option 2 from the ALERT Management Menu, issuing the EXPAND primary command in the ALERT Overview Application, or using the Select|S line command in the ALERT Overview Statistics application causes the following panel to be invoked:

```
BMC Software -------------------- ALERT DETAIL -------------------------------
COMMAND ===>                                                TGT ===> CHICAGO
INTERVAL ==> 3     Suffix ===> 00   COL.   0 - 53           DATE --- 01/03/28
STATUS   --- INPUT                  Queue(s) ===> NETWORK
TIME --- 11:44:41

RSP TIME  IND ORIGIN   ---------------------------------------- Sorted by QUEUE
___ 11:44 ech DETROIT  AMF001I - INITIATOR NOT AVAILABLE FOR FINANCE PRODUCT
___ 11:44 eh  DETROIT  AMF020E - LINES DOWN:
                          - BERLIN
                          - CHICAGO
___ 11:44 h   CHICAGO  ALM0100 8100 COMMUNICATION LINE DOWN:
                       -CHI998A21
___ 11:43     CHICAGO  COMMUNICATION LINES DOWN:
                               - DALLAS
                               - CHICAGO
___ 11:43 e   CHICAGO  COMM CONTROLLER HAS I/O ERROR
                     ********************** END OF ALERTS *********************
```

Figure 243. ALERT Detail Application Panel

While the other ALERT applications concern themselves with information regarding the counts of ALERTs (in effect, providing an abstract picture of the general health of a system), this application allows the display of the nature of a problem. It provides a scrollable listing of the textual content of all ALERTs or a subset of ALERTs present in the target AutoOPERATOR-SS.

The text is formatted according to the line feed and color instructions provided during ALERT generation.

The fields on this panel are:

Field	Contents
Interval	Rate in seconds in which the contents of this panel will be refreshed if placed into auto-refresh using the GO primary command.
Suffix	Profile suffix used to construct panel. Use the SET primary command or overtype this value to select a different configuration. To construct a new setup, enter a new suffix. You will be prompted for the required values.
Col.	Columns of ALERT text displayed. The first value may be overtyped, thus effectively scrolling the text horizontally. The second value is automatically computed from the first.

Field	Contents
Status	Indicator whether application has been entered into screen refresh: • Input: Application accepts input • Running: Application is in screen refresh
Queues	Input field accepting overrides for the standard profile backing this application. If any contents are present in this field, they take precedence, thus temporarily disabling the profile. Valid input values are queue names and queue name patterns.
Rsp	Response action field for the ALERT it precedes. It serves two different functions: • Allows you to delete the indicated ALERT by erasing the underscores and pressing ENTER • Allows you to enter up to 3 characters in this field and, when you press ENTER, a follow-up EXEC is scheduled (if one was specified at ALERT creation) The 3 characters get passed to the EXEC as a positional parameter.
Time	Time the ALERT was issued
Ind	ALERT contents indicator: • E: Follow-up EXEC is available (EXEC keyword was specified at ALERT creation) • C: Command transfer is available (PCMD keyword was specified at ALERT creation) • H: ALERT specific help is available (HELP keyword was specified at ALERT creation) When a help panel is associated with an ALERT, the help text member must be included the BBPLIB concatenation for the terminal session.

Field	Contents
Origin	Origin associated with this ALERT. The origin of an ALERT is one of the following: • **For EXEC-initiated ALERTs**: the default value is the originating EXEC's IMFSYSID; for example, the originating job name for the EXEC • **For Rule-generated ALERTs**: the default value is the subsystem identifier of the subsystem the ALERT was created on • **For either EXEC or Rule generated ALERTs**: the origin may be explicitly specified through the ORIGIN keyword at ALERT creation
Sorted by	Input field designating the sort order in which the ALERTs on this panel are shown. Acceptable values are TIME and QUEUE, and PRIORITY. When sorted by TIME, all ALERTs are shown by creation time in descending order (the ALERTs created most recently appear at the top of the listing). When sorted by QUEUE, all ALERTs are sorted by a combined key of queue and time. In other words, the ALERTs within a queue appear grouped together and sorted within the queue again by time.

The application supports the following specific primary commands:

Command	Function
PROFILE	Invokes the profile setup for the suffix currently active (in other words, entered in the field labeled Profile suffix). The panel shown in page 408 will be displayed.
TRANSFER	Performs the command transfer specified for a particular ALERT. This command requires the cursor to be placed within the text area of an ALERT that bears a C in the Ind. column, indicating for this ALERT a command association is present. **Note:** The TRANSFER command is designed to be assigned to program function keys and act as a hot-key rather than to be explicitly entered in the COMMAND field.

Command	Function
EXPAND	Retrieves and displays help information associated with a particular ALERT. This command requires the cursor to be placed within the text area of an ALERT that bears an H in the Ind. column, indicating help information is present for this ALERT. **Note:** The EXPAND command is designed to be assigned to program function keys and act as a hot-key rather than to be explicitly entered in the COMMAND field. When a help panel is associated with an ALERT, the help text member must be included the PNLLIB concatenation for the terminal session.
SET	Selects a new profile to define the subset of ALERTs displayed. To construct a new setup, enter a new suffix. You will be prompted for the required values.
GO	Places the application into screen refresh. Press the PA1 or Attention key to exit this mode.
DETAIL	Displays all information associated with a particular ALERT. This command requires the cursor to be placed within the text area of an ALERT.

Using a Profile

A profile application drives the ALERT Detail application and allows you to restrict the ALERTs shown on the ALERT Detail panel. An example of an ALERT Detail Profile panel is shown in Figure 244.

```
BBMC Software  -------------- Alert Detail Profile --------------- AutoOPERATOR
COMMAND ===>
                                                                 Date --- 01/03/28
             Profile suffix 00                                   Time --- 12:07:43

Specify INCLUDE(s) below:

User     ===> *             / *             / *             / *
Queue    ===> *             / *             / *             / *
Target   ===> *             / *             / *             / *
Priority ===> *             / *             / *             / *

Specify EXCLUDE(s) below:

User     ===>               /               /               /
Queue    ===>               /               /               /
Target   ===>               /               /               /
Priority ===>               /               /               /


Initial Alert sort order ===> PRIO     ( QUEUE, TIME or PRIO )

Enter CANCEL to cancel, END to save
```

Figure 244. ALERT Detail Profile Panel

Up to 4 patterns each may be specified for the ALERT fields USER, QUEUE, TARGET, and PRIORITY. Only ALERTs matching at least 1 of the 4 criteria for each category become eligible for display.

Field Name	Description
User	Is the 1- to 8-character name of the user ID the ALERT is addressed to; for example, a 1- to 8-character TSO user ID.
Queue	Is the name of the queue to access or into which to place the ALERT.
Target	Is the target associated with a single AutoOPERATOR only. It is not possible to display the ALERTs from multiple AutoOPERATORs on the same ALERT Detail panel. To see ALERTs from another AutoOPERATOR, create a different profile using target definitions from that AutoOPERATOR.
Priority	Is the priority of the ALERT. For example, you can specify one of the following priorities for an ALERT: CRITICAL(RED) MAJOR(PINK) MINOR(YELLOW)

WARNING(DKBLUE)

INFORMATIONAL(LTBLUE)

CLEARING(GREEN)

These fields also are described as keywords for the IMFEXEC ALERT command statement in the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs* guides. Refer to this discussion to learn how to create and construct AutoOPERATOR ALERTs.

Use the area under `Specify INCLUDE(s) below` to list the ALERTs that you want to display on the ALERT DETAIL panel.

Use the area under `Specify EXCLUDE(s) below` to list the ALERTs that you want to exclude from being displayed on the ALERT DETAIL panel. For example, you may have specified that ALERTs will be published to PATROL Enterprise Manager (PATROL EM) consoles but you might not necessarily want to see these ALERTs displayed on the ALERT DETAIL panel. You can use these EXCLUDE fields to specify that those ALERTs will not be displayed on the ALERT DETAIL panel while they are still being sent to PATROL EM consoles.

In addition to specifying eligibility criteria, this panel defines the initial sort order in which the selected ALERTs will be shown. The contents of the fields in Figure 244 preset the contents of the corresponding field on the ALERT Detail panel.

Getting Extended Help

When using the EXPAND primary command to retrieve extended help information associated with an ALERT, a scrollable listing of the associated data set member is shown.

An example of a possible display is the following:

```
BMC Software -------------- Extended ALERTs -------------------- AutoOPERATOR
COMMAND ===>                                                     TGT  ===> SYSC
                                                                 DATE --- 01/03/30
                                                                 TIME --- 13:20:57

TIME  IND ORIGIN   ------------------------------------------------------------
11:44 ech DETROIT  AMF001I - INITIATOR NOT AVAILABLE FOR FINANCE PRODUCTION
 **************************** EXTENDED ALERT TEXT ***************************
 AMF001I - Initiator not available for FINANCE production

     This ALERT is issued only when there was an error detected in
     in the FINANCE backup job series - FINlljj.  Where ll is the
     datacenter location and jj is the job number in the backup series.

 PROBLEM DETERMINATION:

     Review the job run book for the FINANCIAL backup procedures.
     Pay particular attention to the section on NON-ZERO return codes.

 ACTION:

     If the return code is one that provides WARNING information,
     place the cursor on the ALERT and press the TRANSFER key.  This
     will start the necessary INITIATORS to run the FINANCIAL
```

Figure 245. ALERT Help Information (Extended ALERT Text) Panel

When a help panel is associated with an ALERT, the help text member must be included the BBPLIB concatenation for the terminal session.

Seeing ALERT Detail Information

To see all information pertinent to a single ALERT, which is particularly useful for debugging purposes:

1. Enter the DETAIL primary command in the COMMAND field of the ALERT Detail panel.
2. Position the cursor on the ALERT you want to investigate.
3. Press ENTER.

A panel similar to the following will be displayed:

```
BMC Software -------------- ALERT detail information ----------- AutoOPERATOR
COMMAND ===>
Time     : 12:36:02           Date: 30-MAR-01     Help: AMF001I
Origin   : DETROIT       System ID: RE1X         Alarm: NO     Tgt switch:  NO
Priority : MAJOR              User: RAE1         Color: PINK       Retain: YES
Queue    : MVS              Target: RE1X       Creator: EXEC  / DEMOALRT
Key      : MVS
Text     : AMF001I - INITIATOR NOT AVAILABLE FOR FINANCE PRODUCTION         :
         :                                                                  :
         :                                                                  :
         :                                                                  :
EXEC     : ALRTINIT                                                         :
         :                                                                  :
         :                                                                  :
         :                                                                  :
Command  : #$SI25                                                           :
         :                                                                  :
         :                                                                  :
         :                                                                  :
User data:                                                                  :
         :                                                                  :
         :                                                                  :
         :                                                                  :
                  Hit ENTER to view rest of Alert detail or END to exit
```

Figure 246. ALERT Detail Information Panel

Pressing ENTER displays the next panel.

```
BMC Software ----------- ALERT detail information (cont) -------- AutoOPERATOR
COMMAND ===>
Disposition            :                                                    :
Escalation intervals 1 :                                                    :
Escalation intervals 2 :                                                    :
Escalation intervals 3 :                                                    :
Escalation intervals 4 :                                                    :
Escalation intervals 5 :                                                    :
Escalation intervals 6 :                                                    :
Escalation direction   :                                                    :
Escalation EXEC/parms  :                                                    :
                       :                                                    :
                       :                                                    :
                       :                                                    :




                      Enter END to return to previous panel
```

Figure 247. ALERT Detail Information Panel: Secondary Panel

These panels show all the parameters that can be associated with an ALERT when an ALERT is created either in the Rule Processor or the IMFEXEC ALERT command.

The following describes the fields on the primary panel:

Field	**Contents**	**Associated IMFEXEC Keyword**
Time	Time the ALERT was created	N/A
Date	Date the ALERT was created	N/A
Help	Name of any associated help member (retrieved using EXPAND command) When a help panel is associated with an ALERT, the help text member must be included the BBPLIB concatenation for the terminal session.	HELP
Origin	Origin of the ALERT (the default origin is subsystem from where the ALERT was issued)	ORIGIN
System ID	The subsystem identifier (SSID) where the ALERT originated from	N/A
Alarm	Indicates whether or not the ALERT is accompanied by a warning beep	ALARM
Tgt switch	Indicates whether or not target switching was specified for a PCMD that was associated with the ALERT	SYSTEM
Priority	ALERT priority	PRIO
User	ALERT target user	USER
Color	Color in which ALERT should be displayed	COLOR
Retain	Indicates whether or not the ALERT is coded to be retained between restarts of the BBI-SS PAS	RETAIN
Queue	Queue in which the ALERT resides	QUEUE
Target	ALERT subsystem target (the default is subsystem where ALERT originated)	TARGET
Creator	Programmatic origin of the ALERT; can be: • As a result of an EXEC or a RULE. • If created by an EXEC, the indicator EXEC is followed by the name of the EXEC. If created by a Rule, the indicator RULE is followed by the identifier of the Rule.	N/A
KEY	ALERT identifier for retrieval reference positional parameter	KEY

Field	Contents	Associated IMFEXEC Keyword
TEXT	ALERT text displayed in ALERT Detail application positional parameter	TEXT
EXEC	Name of follow-up EXEC and any parameters	EXEC
Command	Associated command	PCMD
User Data	Display user-defined data associated with the ALERT. The contents can be retrieved using the READQ function	UDATA

The following describes the fields on the secondary panel. Note that the basic information relating to the ALERT is repeated at the top of the panel.

Field	Contents	Associated IMFEXEC keyword
Disposition	Disposition of the ALERT after the final interval elapses.	DISPOSE
Escalation Intervals 1 through 6	Allows you to see the user-specified time periods over which ALERTs can either increase or decrease in severity. When you create an ALERT, you can specify up to six separate intervals of time over which the priority of the ALERT can be changed. When the final priority is reached, the action specified by the DISPOSE parameter is taken. For more information about creating ALERTs, refer to Table 54 on page 393.	INTERVAL
Escalation Direction	Shows whether the ALERT will escalate in severity (UP) or be downgraded (DOWN) in severity. Default is UP.	ESCALATE
Escalation EXEC/parms	Shows the name of an EXEC and any additional parameters that will be invoked when the ALERT reaches its final priority level.	ESCEXEC

When you are viewing ALERT detail information for ALERTs created before AutoOPERATOR release 4.1, these fields will display `N/A`.

Chapter 13. Monitoring Automation with the Automation Reporter

This chapter provides:

- An introduction to monitoring automation with the Automation Reporter
- Additional discussions about the Automation Reporter application that include:
  - How to activate the Automation Reporter with the Dynamic Parameter Manager application
  - How to set timed intervals
  - How to set the data offload interval
  - What the data looks like and how to read it

Important
To use the Automation Reporter application, you must implement and activate the REXX/370 Alternate library. For more information about implementing the REXX/370 Alternate library, refer to the *MAINVIEW AutoOPERATOR Customization Guide*.

Introduction: What the Automation Reporter Application Does

The Automation Reporter application, once activated, tracks and collects data (in numerical values) for automation activities and the consumption of resources over user-specified time intervals.

An automation *activity* can be the:

- Number of TS users logged on during a time interval
- Number of ALERTs created during a time interval
- Number of ALERTs still active at the end of the time interval
- Number of events that occurred

A *resource* is anything a program needs to complete its task, such as CPU or virtual storage. Many resources are shared between several tasks, so it is important to minimize use and potential contention for resources between tasks.

Collecting data for these activities and resources (also collectively called data items) allows you to measure how efficiently the automation you have implemented is performing on your system over time.

You can store the collected data in a sequential data set. By storing the data in a data set, you can import the data into a report generator or a spreadsheet tool and create reports to demonstrate how much automation you have in place on your system.

"What Kind of Data Is Collected" on page 427 briefly describes the activities and resources that can be measured. "How to Activate the Automation Reporter" on page 416 provides an overview as to how the Automation Reporter is activated.

How to Activate the Automation Reporter

Activation of the Automation Reporter is accomplished through the AutoOPERATOR application Dynamic Parameter Manager. The Dynamic Parameter Manager (DPM) is a panel-driven application that allows you to view or modify parameters in BBPARM members from a BBI-TS session.

The BBPARM members currently supported by DPM are:

AAOEXPxx	AutoOPERATOR EXEC-related parameters
AAOALSxx	AutoOPERATOR ALERT thresholds
AAOARPxx	Automation Reporter-related parameters
AAOTSPxx	TapeSHARE for AutoOPERATOR-related parameters.

The AAOARPxx member is the one that controls the Automation Reporter and it is where you specify information such as:

- **Time interval** at which the Automation Reporter is to record data about automation resources and activities

  For example, you can specify that the Automation Reporter records data every 15 minutes or every 1440 minutes (once a day). If you specify 15 minutes, then every 15 minutes the Automation Reporter records the information for the resources and activities as they exist at the end of the interval.

- **Offload interval** which determines how often the collected data is moved to a pre-allocated sequential data set

  For example, you can specify that the offload interval is once an hour (minimum) or once a day (maximum) and the collected data will be written to a sequential data set you have pre-allocated.

- **Name of the pre-allocated sequential data set** that the collected data is moved to

  You must pre-allocate a sequential data set and enter its name into DPM. At the offload interval, the collected data is written into this data set.

- **Disposition** of the sequential data set

  You can specify whether the sequential data set you allocated is going to have a disposition of MOD or SHR.

- AutoOPERATOR **ALERT queue name** where Automation Reporter ALERTs are queued

  Any AutoOPERATOR ALERTs generated by the Automation Reporter will be routed automatically to the ALERT queue name you specify here.

For more information about activating the Automation Reporter and specifying these parameters, refer to “Activating the Automation Reporter with the Dynamic Parameter Manager” on page 417.

Note: For information about the other BBPARM members that the DPM allows you to edit, refer to the *MAINVIEW AutoOPERATOR Customization Guide*.

Activating the Automation Reporter with the Dynamic Parameter Manager

This section describes all the steps you must go through to start (and stop) the Automation Reporter application. The Automation Reporter application is controlled by parameters in BBPARM member AAOARPxx which you can access through the Dynamic Parameter Manager. Use these parameters to:

- Specify a user-allocated sequential data set name and the disposition of the data set
- Specify the interval at which the Automation Reporter is supposed to record data
- Specify the offload interval at which the data is actually written to the data set
- Specify the name of an AutoOPERATOR ALERT queue where any AutoOPERATOR ALERTs are routed

Step 1: Accessing the Automation Reporter Parms Panel in the DPM Application

1. From the AutoOPERATOR Primary Option Menu, select Option 9 `Online AOPARM Administration` to access the Dynamic Parameter Manager application.

   The following panel is displayed:

```
BMC Software -- - ---------- Dynamic Parameter Manager ------------- AutoOPERATOR
COMMAND ===>                                                  TGT ===> MS@1
Primary command: Add                                          DATE --- 01/03/28
                                                              TIME --- 10:01:49
LC CMDS --- (S)elect, (A)ctivate

Cmd   Member        Lib VV.MM   Created      Changed      Size   ID
    ________
_   AAOALS00         1  01.04  97/02/07  01/03/26 13:07     28   MES1
_   AAOARP00         1  01.17  96/10/17  01/03/26 13:08      6   MES1
_   AAOEXP00         1  01.15  96/12/02  01/03/03 11:56     25   MES2
_   AAOTSP00         1  01.15  96/07/02  01/03/01 09:23     12   RYS1
     *************************** END OF MEMBERS ******************************
```

   Note that you can access and edit BBPARM members AAOALS00, AAOEXP00, AAOARP00, and AAOTSP00 but in this section only the parameters in AAOARP00 are discussed. For documentation about the other members, refer to the *MAINVIEW AutoOPERATOR Customization Guide*.

2. Select `AAOARP00` with the `(S)elect` line command to edit the Automation Reporter-related parameters in this member.

The Automation Reporter `Parms` panel is displayed:

```
BMC Software -- - ------------ Automation Reporter Parms ----------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- MS@1
Primary command: ADD                                          DATE --- 01/03/28
                                                              TIME --- 13:39:59
              Member             Current

   New DSN  ===>
                Current  ===

   Disp     ===>      MOD                   Dataset disposition (SHR, MOD)

   Interval ===>       15                   Interval minutes (10 - 1440, or 0)

   Off int  ===>     1440                   Offload interval minutes (60 - 1440)

Alert Queue ===>  MAIN                      Alert Queue

Press END to save changes,  CANCEL to cancel changes
```

The first time you bring up this panel, the display-only field `Current ===` is blank and all the values in the display-only fields of this panel (which is the right side) are blank.

This is because the first time you bring up the Automation Reporter, there are no current data set values to display in these fields. After you have used the Automation Reporter once, the next time you access this panel, the `Current ===` field and all the display fields will show the values you previously entered.

The left side of the panel will show the default values in BBPARM member AAOARPxx as shown in this panel.

Proceed to “Step 2: Allocate a Data Set”.

Step 2: Allocate a Data Set

Before you use the DPM application to activate the Automation Reporter, you must allocate a data set to which the data collected by the Automation Reporter is written.

Data set requirements: The data set should be a fixed block sequential data set and the maximum record length of 580 bytes.

This data set name is specified in the DPM application in BBPARM member AAOARPxx and then you can modify the other parameters that control how the Automation Reporter performs its data gathering functions.

After you have allocated the sequential data set, to specify the data set name in DPM:

1. In the `New DSN` field of the Automation Reporter Parms panel, enter the name of the sequential data set you have preallocated.

   See number **1** in the following figure.

```
BMC Software -- - ------------ Automation Reporter Parms ----------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- MS@1
                                                              DATE --- 01/03/28
                                                              TIME --- 13:39:59
              Member             Current

1 New DSN  ===> AAO41.SAVE.ARSTATS
                  Current  ===

    Disp     ===>      MOD                   Dataset disposition (SHR, MOD)

    Interval ===>       15                   Interval minutes (10 - 1440, or 0)

    Off int  ===>     1440                   Offload interval minutes (60 - 1440)

 Alert Queue ===>  MAIN                      Alert Queue

Press END to save changes,  CANCEL to cancel changes
```

In this example, the new sequential data set is `AAO41.SAVE.ARSTATS`. This data set must be a pre-allocated sequential data set and have a maximum record length of 580 bytes.

Proceed to “Step 3: Specifying the Data Set Disposition”.

Step 3: Specifying the Data Set Disposition

In this step, specify the disposition of the data set you specified in Step 1.

Valid values: Valid disposition values are SHR (share) and MOD (modify), which is the default value.

Specifying SHR causes the Automation Reporter to overwrite all existing data.

Specifying MOD causes the Automation Reporter to append data to the end of the data set from each offload cycle.

To specify the data set disposition:

1. At the `Disp` field of the Automation Reporter Parms panel, enter the disposition of the data set.

   See number **2** in the following figure.

```
BMC Software -- - ------------ Automation Reporter Parms ----------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- MS@1
                                                              DATE --- 01/03/28
                                                              TIME --- 13:39:59
              Member             Current

   New DSN  ===> AAO41.SAVE.ARSTATS
                 Current  ===

2  Disp     ===>      MOD                   Dataset disposition (SHR, MOD)

   Interval ===>       15                   Interval minutes (10 - 1440, or 0)

   Off int  ===>     1440                   Offload interval minutes (60 - 1440)

Alert Queue ===> MAIN                       Alert Queue

Press END to save changes,  CANCEL to cancel changes
```

In this example, the data set disposition was left as its default value, `MOD`.

Proceed to “Step 4: Specifying Collection Intervals”.

Step 4: Specifying Collection Intervals

In this step, specify the collection interval (in minutes) for the Automation Reporter to collect data. For example, if you specify 60, then every 60 minutes the Automation Reporter will record the data in the subsystem for the resources and activities.

Valid values: Valid collection intervals range from 10 minutes to 1440 minutes. The default is 15 minutes.

To specify the collection interval:

1. In the `Interval` field of the Automation Reporter `Parms` panel, enter the collection interval. See number **3** in the following figure.

```
BMC Software -- - ------------ Automation Reporter Parms ----------- AutoOPERATOR
COMMAND ===>                                                   TGT  --- MS@1
                                                               DATE --- 01/03/28
                                                               TIME --- 13:39:59
              Member             Current

    New DSN  ===> AAO41.SAVE.ARSTATS
                  Current  ===

    Disp     ===>      MOD                    Dataset disposition (SHR, MOD)

3   Interval ===>       60                    Interval minutes (10 - 1440, or 0)

    Off int  ===>     1440                    Offload interval minutes (60 - 1440)

 Alert Queue ===> MAIN                        Alert Queue

Press END to save changes,  CANCEL to cancel changes
```

In this example, the interval is specified as 60 minutes. Therefore, after this DPM member is activated, every hour the Automation Reporter records the data as it exists at the end of the interval.

For example, if there were 10 ALERTs active during the 60 minute interval but only 5 ALERTs are active at the end of the 60 minute interval, the Automation Reporter records that there are 5 active ALERTs. The data is collected for each interval and saved in the subsystem and offloaded to the data set when the offload interval expires.

Note: The data is not available to be browsed until it is offloaded into the data set.

Hint
Choose the collection interval value carefully. You want to ask yourself whether you really need to have data collected once an hour? Once every 15 minutes? If you specify every 15 minutes, data is collected for all the resources and activities 4 times an hour, 24 hours a day. You may or may not need to collect data at this frequency. BMC Software recommends specifying twice a day collection (or every 720 minutes).

Proceed to “Step 5: Specifying the Offload Interval”.

Step 5: Specifying the Offload Interval

In this step, specify the offload interval (in minutes) for the Automation Reporter to offload the data it has collected to the data set. For example, if you specify 120, then every 2 hours the Automation Reporter will write the data it has collected to the sequential data set you specified in Step 1.

Valid values: Valid offloading intervals are 60 minutes to 1440 minutes. The default is 1440 (or once a day).

To specify the offload interval:

1. In the `Off int` field of the Automation Reporter Parms panel, enter the offload interval. See number **4** in the following figure.

```
BMC Software -- - ------------ Automation Reporter Parms ----------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- MS@1
                                                              DATE --- 01/03/28
                                                              TIME --- 13:39:59
              Member             Current

    New DSN  ===> AAO41.SAVE.ARSTATS
                  Current  ===

    Disp     ===>      MOD                    Dataset disposition (SHR, MOD)

    Interval ===>       60                    Interval minutes (10 - 1440, or 0)

4   Off int  ===>      120                    Offload interval minutes (60 - 1440)

Alert Queue ===> MAIN                        Alert Queue

Press END to save changes,  CANCEL to cancel changes
```

In this example, the offload interval is specified as 120. Therefore, after this DPM member is activated, the Automation Reporter records the collected data once every 2 hours, for the life of the BBI-SS PAS PAS, to the sequential data set.

Hint
The offload interval should always be larger than the collection interval. If you plan to collect data every 12 hours (or 720 minutes), BMC Software recommends specifying once a day offloading (or 1440 minutes).

Proceed to “Step 6: Specifying an AutoOPERATOR ALERT Queue Name”.

Step 6: Specifying an AutoOPERATOR ALERT Queue Name

In this step, specify the name of an AutoOPERATOR ALERT queue to which any AutoOPERATOR ALERTs created by the Automation Reporter will be queued to. For example, if you specify ARALRT, then any ALERTs will be queued to a queue named ARALRT.

Valid values: Any valid AutoOPERATOR ALERT queue name can be entered in this field (default queue name is MAIN). You do not have to enter the name of a pre-existing ALERT queue. You can enter the name of a new queue and it will be created for you when the Automation Reporter is activated and ALERTs are generated.

To specify the ALERT queue name:

1. In the `Alert Queue` field of the Automation Reporter Parms panel, enter the name of the ALERT queue that you want Automation Reporter ALERTs queued to.

   See number **5** in the following figure.

```
BMC Software -- - ------------ Automation Reporter Parms ----------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- MS@1
                                                              DATE --- 01/03/28
                                                              TIME --- 13:39:59
              Member             Current

      New DSN  ===> AAO41.SAVE.ARSTATS
                    Current  ===

      Disp      ===>      MOD                Dataset disposition (SHR, MOD)

      Interval  ===>      60                 Interval minutes (10 - 1440, or 0)

      Off Int   ===>     120                 Offload interval minutes (60 - 1440)

5    Alert Queue ===> ARALRT                  Alert Queue

Press END to save changes,  CANCEL to cancel changes
```

In this example, the ALERT queue name is ARALRT. This queue, if it has not already been created, will be created when the first ALERT is generated by the Automation Reporter.

Proceed to “Step 7: Activating the Automation Reporter”.

Step 7: Activating the Automation Reporter

The Automation Reporter is initialized and it begins gathering data at the intervals you specified when you use the Activate function of the DPM to activate BBPARM member AAOARPxx.

Note: You also may use the BBI command `.E P AAOARPXX` to activate the AAOARPxx member.

To activate the AAOARP00 member:

1. After you have completed steps 1 through 6, press PF3 to save the values you entered.

   The Dynamic Parameter Manager panel is displayed:

```
BMC Software -- - ---------- Dynamic Parameter Manager ------------- AutoOPERATOR
COMMAND ===>                                                TGT ===> MS@1
Primary command: Add                                        DATE --- 01/03/28
                                                            TIME --- 10:01:49
LC CMDS --- (S)elect, (A)ctivate

Cmd   Member        Lib VV.MM   Created      Changed      Size   ID
    ________
_   AAOALS00         1  01.04  97/02/07  01/03/26 13:07     28   MES1
_   AAOARP00         1  01.17  96/10/17  01/03/28 13:08      6   MES1
_   AAOEXP00         1  01.15  96/12/02  01/03/03 11:56     25   MES2
_   AAOTSP00         1  01.15  96/07/02  01/03/01 09:23     12   RYS1
     *************************** END OF MEMBERS ******************************
```

2. Activate the AAOARP00 member by entering an `A` (for the Activate command) in the `Cmd` column.

   See number **6** in the following figure.

```
BMC Software -- - ---------- Dynamic Parameter Manager ------------- AutoOPERATOR
COMMAND ===>                                                TGT ===> MS@1
Primary command: Add                                        DATE --- 01/03/28
                                                            TIME --- 10:01:49
LC CMDS --- (S)elect, (A)ctivate

Cmd   Member        Lib VV.MM   Created      Changed      Size   ID
    ________
_   AAOALS00         1  01.04  97/02/07  01/03/26 13:07     28   MES1
A 6 AAOARP00         1  01.17  96/10/17  01/03/28 13:08      6   MES1
_   AAOEXP00         1  01.15  96/12/02  01/03/03 11:56     25   MES2
_   AAOTSP00         1  01.15  96/07/02  01/03/01 09:23     12   RYS1
     *************************** END OF MEMBERS ******************************
```

   The Confirm Parameter Activation panel is displayed.

See numbers **7** and **8** in the following figure.

```
BMC Software -- - ---------- Confirm Parameter Activation ---------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- MS@1
                                                              DATE --- 01/03/29
                                                              TIME --- 10:02:21

Current TARGET:      MS@1

Current SSID:        MS@1


Last Parm:       7   N/A       (Values may have been temporarily modified)

Replaced by:     8   AAOARP00
Last update:
User:



Instructions:

  Press ENTER to confirm activation request.
     (The parameters in the current SS will be dynamically updated)

  Press END   to cancel activation request.
```

The first time you activate a member, the `Last Parm` field value is `N/A` (**7**) and the `Last Update` (**8**) field is blank. In the future, when you activate a member, the `Last Parm` field shows the date of the currently active AAOARPxx member. The `Last Update` field shows the date you activated the current member.

3. To activate the changes you entered, press ENTER.
4. To cancel activating this member, press PF3.

Once you activate the AAOARP00 member, the Automation Reporter will begin recording data and offloading it to a sequential data set at the intervals you just specified.

For information about stopping the Automation Reporter, refer to “Stopping the Automation Reporter” on page 426. For information about the data the Automation Reporter collects and what it looks like, refer to “Creating Automation Reporter Reports” on page 427.

Stopping the Automation Reporter

To stop the Automation Reporter from collecting data:

1. Display the Automation Reporter `Parms` panel and edit the current AAOARPxx member; for example:

```
BMC Software -- - ------------ Automation Reporter Parms ----------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- MS@1
                                                              DATE --- 01/03/28
                                                              TIME --- 13:39:59
              Member             Current

   New DSN  ===>
                 Current  ===  AAO41.SAVE.ARSTATS

   Disp     ===>                       MOD  Dataset disposition (SHR, MOD)

   Interval ===>  9 0                 60  Interval minutes (10 - 1440, or 0)

   Off int  ===>                       120  Offload interval minutes (60 - 1440)

Alert Queue ===>                    ARALRT  Alert Queue

Press END to save changes,  CANCEL to cancel changes
```

2. Enter a zero (0) in the `Interval` field (at **9**).

   Entering a 0 turns off all data recording.

3. Press PF3/END to save this change.

4. Activate this member.

   Follow the instructions in “Step 7: Activating the Automation Reporter” on page 424 to activate this AAOARPxx member.

Once the BBPARM member AAOARPxx is activated, the Automation Reporter, is turned off and data recording ceases.

For information about what the collected data looks like, refer to “Creating Automation Reporter Reports”.

Creating Automation Reporter Reports

With the Automation Reporter, you can store the data collected into a sequential data set, thereby creating a historical record of how AutoOPERATOR activities and resources performed over hours and days. By using SAS, COBOL, report writing tools, or spreadsheet tools, you can sort and organize the historical data to track and illustrate what automation is taking place on your system over specific time periods.

What Kind of Data Is Collected

Once the Automation Reporter is activated and begins collecting data, data is collected for all of the following items:

Data Item Name	Description
Actions (ACTN)	The ACTN data item contains 7 fields of data (action record data count) for each of 13 different action records (one built for each Event type): MSG CICS JRNL TIME ALRT DB2 EXT IMS CMD JES3 ALRM VAR MQS Within one ACTN record are 7 fields of data for these 13 events. The data contains values representing things such as how often the events occurred, how many were handled or suppressed, and so on. This data is collected in intervals.
AutoOPERATOR ALERTS (ALRT)	The ALRT data item contains data for six different priorities of AutoOPERATOR ALERTs: • CRITICAL • MAJOR • MINOR • WARN • INFO • CLEAR Within one ALRT record are eight fields of data for these six ALERT priorities. The data contains values representing things such as how many ALERTs were active, how many were created or deleted, and so on. This data is collected in intervals.

Data Item Name	Description
Events (EVNT)	One record is collected for each entry in the Event Statistics Table. The record contains information such as the Text-ID of the event, how many times the event occurs in the Event Statistics Table, the number of times the event was handled, and so on. This data is collected cumulatively.
AutoOPERATOR EXECs (EXEC)	One record is collected for each EXEC in the SYSPROC and the SYSEXEC concatenation. The record contains information such as the EXEC name, the number of times the EXEC ran, abended, and so on. This data is collected cumulatively.
AutoOPERATOR Rules (RULE)	One record is created for each enabled, disabled, and suspended Rule. The record contains information such as the name of the Rule, how many times it fired, the Rule Set it belongs to, and so on. This data is collected cumulatively.
AutoOPERATOR Rule Sets (RULESET)	There are two subtype records for RULESET: • SET Contains information about the Rule Set strategy (if an overall strategy of INDIVIDUAL is used in the Rules Processor), the number of Rules in the Rule Set, the fired count of the Rule Set, and so on. One SET record is created for every Rule Set that is active at the time offload interval time (set by the `Off Int` parameter). If 10 Rule Sets are active at the end of the interval, 10 SET records are created. • AUTO Contains information about Rules automation such as which overall automation strategy the Rules Processor is using, whether HONORMPF was turned on, and so on. One AUTO record is created at the offload interval time (set by the `Off Int` parameter). If 10 Rule Sets are active at the end of the interval, you will have 1 AUTO record and 10 SET records. Data for both these subtypes is collected cumulatively.
Resources (RES)	The record contains information about how much private storage (above and below) was allocated, how many PQEs were created, how many CPU seconds the subsystem used, and so on. This data is collected in intervals.

Data Item Name	Description
TapeSHARE for AutoOPERATOR (TAPE)	When you have the TapeSHARE for AutoOPERATOR component implemented in an MVS image (also called a partner) where Automation Reporter also is running, you will collect statistics for the tape sharing activity performed by TapeSHARE in TAPE records. There are two subtype records for TAPE: • LOCAL Contains information about tape sharing activity such as the total number of tape device requests, total time needed for a TAKE, how many devices were shared, and so on. One LOCAL record is created per interval where both TapeSHARE and Automation Reporter are operating. If 10 partners are participating in tape sharing activity with TapeSHARE at the end of the interval, you must save and collect the Automation Reporter TAPE: LOCAL records from each partner individually. • PLEX Contains information about partner to partner communication such as how often a partner requested devices from the other partners, how long it took for a device to be shared, and so on. One PLEX record is created per interval. Within one PLEX record, if 10 partners are active at the end of the interval, you will have 9 fields of data collected for each of the 10 partners. Data for both these subtypes is collected in intervals.

Refer to “What Record Types Are Available” on page 435 for a matrix that shows the data items, their subtypes, how the data is collected, and a brief description.

How the Data Is Saved

After a BBI-SS PAS is cold started and the Automation Reporter is activated, the Automation Reporter begins to collect and save data in the BBI-SS PAS before it is written to the sequential data set. If during the life of the BBI-SS PAS you issue the

```
.E STATS
```

command, all data collection for events and Rules is erased and the Automation Reporter collects data beginning from the time you issued this command.

For more information about collecting data and BBI-SS PASs, refer to “Managing Data Collection Over the Life of the BBI-SS PAS” on page 431.

Data is saved in **intervals** or **cumulatively**.

What Is Interval Data

Interval Data is data collected at user-specified intervals. It contains information about the activity (or resource) that was measured at the end of the interval. The interval duration is

specified on the `Interval` field of the Automation Reporter Parms panel (see “Step 4: Specifying Collection Intervals” on page 421).

When a new interval begins, data is collected **as if the previous interval did not occur.**

For example, if the Automation Reporter begins collecting data at 9:00 am in 30 minute intervals and you choose to dump the collected data at 10:30 am the same morning, you will have three records for each item: one for 9:00 to 9:30, from 9:30 to 10:00, and from 10:00 to 10:30.

To see how a specific item performed over the entire hour and half period, you would **add up the values in all three of these records.**

The items that the Automation Reporter collects in intervals are:

- ACTN
- ALRT
- RES
- TAPE (LOCAL)
- TAPE (PLEX)

What Is Cumulative Data

Cumulative data represents data collected since the BBI-SS PAS was cold-started (or the .E STATS or .E BLDL commands were issued) **up until the time the data was offloaded into the data set.**

For example, if the BBI-SS PAS is started at 8:30 am and the Automation Reporter is activated and begins collecting data at 9:00 am, the Automation Reporter collects data (without interruption) from 9:00 am until the data is written into the data set.

The value you specify in the `Off Int` field determines when and how often data is offloaded (see “Step 5: Specifying the Offload Interval” on page 422). If you specify 90 minutes in the `Off Int` field, the data is offloaded to the data set at 10:30 am the same day. One record is created for each record type and all the activity for the past 90 minutes is reflected in the data.

If the data is offloaded to the data set a second time at 12:00 noon of the same day (90 minutes later), one record, spanning the past three hours is created. All the activity for the three hours since the BBI-SS PAS was up and the Automation Reporter is collecting data is reflected in the values recorded.

The items that the Automation Reporter collects cumulatively are:

- EVENT
- EXEC
- RULE
- RULE SET, AUTO
- RULE SET, SET

Refer to "What Record Types Are Available" on page 435 for a table that shows all the items the Automation Reporter collects for, what kind of data is collected for the item (interval or cumulative), and a brief description.

Managing Data Collection Over the Life of the BBI-SS PAS

When the Automation Reporter is active and recording data, the data is saved in the BBI-SS PAS before it is written to the user-allocated sequential data set.

To make sure data is not lost during a BBI-SS PAS cold start, set the interval parameter to zero (0) prior to BBI-SS PAS termination to stop and save data collection. You can write a Rule to do this automatically for you.

The Rule should capture the `Purge` or `Cancel` command issued against the BBI-SS PAS and invoke an EXEC that issues two commands with a short wait specified between them. The first command the EXEC should issue is `.E P AAOARPxx`, where xx is the suffix of the BBPARM member AAOARPxx that is currently controlling the Automation Reporter.

Specify that the EXEC waits until the `DATA OFFLOAD` message is issued by the Automation Reporter then reissues the `Purge` or `Cancel` command against the BBI-SS PAS. This ensures that the already collected data is saved and the BBI-SS PAS can be stopped.

BBI-SS PAS abends: If the BBI-SS PAS abends unexpectedly, all cumulative data is lost but the interval data collected up until the abend is not lost.

When the BBI-SS PAS is restarted, the Automation Reporter is restarted automatically and continues to operate with all the same parameters as before the BBI-SS PAS abend. Data collection continues and interval data collected is appended to the interval data collected before the BBI-SS PAS abended. Cumulative data is collected anew and will be offloaded at the specified offload interval.

What the Record Looks Like: Basic Data Record Design

The basic record is comprised of data fields separated by commas, also known as Comma Separated Data (CSD) format. This format is parsed easily by many reporting packages, which is important because you may decide to create a PC spreadsheet with the collected data.

There is a header portion (always shown in capital letters) that contains consistent information for every record and a data portion that contains specific data for information collected for every interval. This is an example of a data record:

```
SYSID,SSID,VVR,SYSPLEX,"YYYYMMDD","HH:MM",INTERVAL,RECTYPE,SUBTYPE,
"YYYYMMDD","HH:MM",data,EOR
```

The individual fields of the record are described in Table 55.

Table 55. Header Fields Description

Field Name	Maximum Length (in Characters)	Data Type (Character or Numeric)	Description
`SYSID` System name	8	Character	Is the MVS system name from the CVT the BBI-SS PAS was executing on
`SSID` Subsystem identifier	4	Character	Is the subsystem identifier (SSID) of the BBI-SS PAS that created this record
`VVR` Record version number	3	Numeric	Is the version and release numbers of the record, for future reporter record compatibility Currently, the value is 410.
`SYSPLEX` Name of the MVS sysplex	8	Character	Is the sysplex name as defined in the ECVT control block
`"YYYYMMDD"` Interval date	8	Numeric	Is the date, in quotes, when the recording interval began
`"HH:MM"` Interval time	5	Numeric	Is the time, in quotes, when the recording interval began
`INTERVAL` Duration of the interval	8	Numeric	Is the length of the recording interval when this record was created in `MMMM` format, where `MMMM` can be any value up to 1440 minutes
`RECTYPE` Record type	8	Character	Is the record type identifier; possible values can be: ACTN ALRT EVNT EXEC RES RULE RULESET TAPE For more information about the different record types you can collect data records for, refer to “What Record Types Are Available” on page 435.
`SUBTYPE` Subtype	8	Character	Is the record subtype identifier For more information about the different record types and subtypes you can collect data records for, refer to “What Record Types Are Available” on page 435.

Table 55. Header Fields Description (Continued)

Field Name	Maximum Length (in Characters)	Data Type (Character or Numeric)	Description
`"YYYYMMDD"` Write date	8	Numeric	For data that is saved in intervals (interval data), it is the time (in quotes) the record was saved in the BBI-SS PAS For data that is saved cumulatively (cumulative data), it is the date the record was offloaded to the sequential data set
`"HH:MM"` Write time	5	Numeric	For data that is saved in intervals (interval data), it is the time (in quotes) the record was saved in the BBI-SS PAS For data that is saved cumulatively (cumulative data), it is the time the record was offloaded to the sequential data set
`data` Collected data	Variable	Alphanumeric	Is data collected for a specific record type and subtype Refer to “Varying Data Record Lengths” for more information about the variable collected data length
`EOR` End of Record	3	Character	Is a 3-character field denoting the end of the record

Varying Data Record Lengths

It is important to note that for any type of data sorting or report writing, the actual record length and the number of fields will vary. The header data is constant in length and content although the type of data collected (ALRT, EVNT, and so on) will cause the field lengths to vary.

The reason for varying length records and a non-determined set of record types is for expansion of the data elements and record types in future releases of the Automation Reporter.

Any data item field with a null or missing value will not be shown, but two commas (,,) indicate the absence of data between them. Therefore, two null or missing values are denoted by three commas (,,,) and so on. Refer to “Describing a Sample Data Record” for more information about what the data record looks like.

Describing a Sample Data Record

This sample record from May 1, 1997, shows a Rule (Rule ID RUL001) was fired 13 times between 8:00 and 8:20 while Automation Strategy was set to FIRST. It also records that the Rule Set name of the Rule is AAORUL01, that an EXEC named RULEXEC was fired, and that the event type that caused the Rule to fire is a MSG event.

The following example shows what the sample data record looks like.

```
  1    2   3      4         5       6      7   8     9      10      11
SYSA,SSA1,410,SYSAPLEX,"19970501","08:00",20,RULE,ONLY,"19970501","10:00"
,

  12       13   14    15     16   17      18  19
RULE001,FIRST,13,AAORUL01,,,,RULEXEC,MSG,EOR
```

The information in the header is described below:

1	SYSA	Is the SMF ID of the system on which the record was created.
2	SSA1	Is the BBI-SS PAS subsystem identifier from which this record was created.
3	410	Is the version and release number of the Automation Reporter record.
4	SYSAPLEX	Is the name of the sysplex created.
5	"19970501"	Is the date the interval began recording.
6	"08:00"	Is the time the interval began recording.
7	20	Is the number of minutes in the time interval.
8	RULE	Is the record type of RULE. For more information about record types and their subtypes, refer to “What Record Types Are Available” on page 435.
9	ONLY	Is the record subtype of ONLY For more information about record types and their subtypes, refer to “What Record Types Are Available” on page 435.
10	"19970501"	Is the date the record was written to the output data set
11	"10:00"	Is the time the record was written to the output data set
12	RUL0001	Is the identifier of the Rule that was fired
13	FIRST	Is the Automation Strategy that was in place in the Rules Processor when the Rule fired
14	13	Is the number of times the Rule was fired within the time interval of the record
15	AAORUL01	Is the name of the Rule Set the Rule belongs to

16	`,,,,`	Shows three blank data fields
17	`RULEXEC`	Is the name of the EXEC that was fired by the Rule
18	`MSG`	Is the event type that triggered the firing of the Rule
19	`EOR`	Signifies the end of the record

When data is missing, as there is at , there are commas to indicate the fields are blank. In this example, the blank data fields are for the Rule documentation fields that were left blank when the Rule was created. The data fields exclusive of the header can be of varying lengths.

What Record Types Are Available

The following table summarizes items the Automation Reporter collects data for, subtypes, whether the data is collected cumulatively or in single intervals, and gives a brief description.

Data Record Type	Subtype	Interval Data	Cumulative Data	Description	See page...
ACTN	ONLY	X		One record is collected per time interval; contains numeric counts for 10 different events within each record.	437
ALRT	ONLY	X		One record is collected per time interval; contains numeric counts for six different ALERT priorities within each record.	443
EVENT	ONLY		X	One record is collected for each entry in the Event Statistics Table. The record contains information such as the Text-ID of the event, how many items the event occurs in the Event Statistics Table, the number of times the event was handled, and so on	445
EXEC	ONLY		X	One record is collected for each EXEC in the SYSPROC and the SYSEXEC BLDL. The record contains information such as the EXEC name, the number of times the EXEC ran, abended and so on	446
RULE	ONLY		X	One record is created for each enabled, disabled, and suspended Rule. The record contains information such as the name of the Rule, how many times it fired, the Rule Set it belongs to, and so on.	447

Data Record Type	Subtype	Interval Data	Cumulative Data	Description	See page...
RULESET	AUTO		X	One AUTO record is created every time the data is offloaded. If 10 Rule Sets are active at the end of the interval, you will have 1 AUTO record and 10 SET records. Contains information about Rules automation such as which overall automation strategy the Rules Processor is using, whether HONOR MPF was turned on, and so on	448
RULESET	SET		X	One SET record is created for every Rule Set that is active at the time the data is offloaded. If 10 Rule Sets are active at the end of the interval, 10 SET records are created. Contains information about the Rule Set strategy (if an overall strategy of INDIVIDUAL is used in the Rules Processor), the number of Rules in the Rule Set, the fired count of the Rule Set, and so on.	449
RES	ONLY	X		One record is collected per time interval; contains information about how much private storage (above and below) was allocated, how many PQEs were created, and so on.	451
TAPE	LOCAL	X		One LOCAL record is created per interval. If 10 partners are participating in tape sharing activity with TapeSHARE at the end of the interval, you must save and collect the Automation Reporter TAPE: LOCAL records from each partner individually. Contains information about tape sharing activity such as how many total device requests there are, how many devices were requested, and so on	453
TAPE	PLEX	X		One PLEX record is created per interval. Within one PLEX record, if 10 partners are active at the end of the interval, you will have 9 fields of data for each of the 10 partners. Contains information about partner to partner communication such as how often a partner requested devices from the other partners, how long it took for a device to be shared, and so on.	455

The following sections contain tables for each record type (and subtype where applicable) and briefly describe possible uses for the collected data.

Tracking Actions (ACTN)

This table describes the data collected for the record type ACTN, subtype ONLY. ACTN records are collected in intervals.

Possible Usage The ACTN records contain 7 fields of data for 10 separate types of Events:

MSG MVS messages
CMD MVS commands
JRNL BBI-SS PAS Journal messages
TIME Timed events
ALRT AutoOPERATOR ALERTs
EXT Externally generated messages
IMS IMS messages
CICS CICS messages
DB2 DB2 messages
JES3 JES3 messages
ALRM ALRM messages
VAR VAR messages
MQS MQS messages

One of the useful things you can use this collected data for is to determine how many of these actions are performed by automation. For example, you can determine the message suppression rate by comparing the EVENT COUNT values for MSG versus the SUPPRESSED COUNT value and calculate what percentage of MSGs are suppressed by automation.

Data Item Name	Max. Field Length	Type of Data	Description
ACTION RECORD COUNT	2	Numeric	Total number of action types within the record
ACTION RECORD DATA COUNT	1	Numeric	Total number of fields of data for each action type
MSG	4	Character	Name of the action type (MSG)
MSG TOTAL	8	Numeric	Total number of MSGs counted during the interval
MSG HANDLED COUNT	8	Numeric	Handled count: in other words, the total number of times a MSG event caused one (or more) Rule to fire
MSG SUPPRESSED COUNT	8	Numeric	Total number of MSGs suppressed from their destination during the interval
MSG SCHEDULED EXEC COUNT	8	Numeric	Total number of EXECs scheduled by MSG events
MSG ALERTS CREATED COUNT	8	Numeric	Total number of ALERTs created by MSG events

Data Item Name	Max. Field Length	Type of Data	Description
MSG HARDCOPY LOG SUPPRESSED COUNT	8	Numeric	Total number of MSGs suppressed from the hardcopy log during the interval
CMD	4	Character	Name of the action type (CMD)
CMD TOTAL	7	Numeric	Total number of CMDs counted during the interval
CMD HANDLED COUNT	7	Numeric	Handled count: in other words, the total number of times a CMD event caused one (or more) Rule to fire
CMD SUPPRESSED COUNT	7	Numeric	Total number of CMDs suppressed from their destination during the interval
CMD SCHEDULED EXEC COUNT	7	Numeric	Total number of EXECs scheduled by CMD events
CMD ALERTS CREATED COUNT	7	Numeric	Total number of ALERTs created by CMD events
CMD HARDCOPY LOG SUPPRESSED COUNT	7	Numeric	Total number of CMD messages suppressed from the hardcopy log during the interval
JRNL	4	Character	Name of the action type (JRNL)
JRNL TOTAL	8	Numeric	Total number of BBI-SS PAS Journal messages counted during the interval
JRNL HANDLED COUNT	8	Numeric	Handled count: in other words, the total number of times a JRNL event caused one (or more) Rule to fire
JRNL SUPPRESSED COUNT	8	Numeric	Total number of JRNL messages suppressed from their destination during the interval
JRNL SCHEDULED EXEC COUNT	8	Numeric	Total number of EXECs scheduled by JRNL events
JRNL ALERTS CREATED COUNT	8	Numeric	Total number of ALERTs created by JRNL events
JRNL HARDCOPY LOG SUPPRESSED COUNT	8	Numeric	Total number of JRNL messages suppressed from the hardcopy log during the interval
TIME	4	Character	Name of the action type (TIME)
TIME TOTAL	4	Numeric	Total number of TIME events counted during the interval
TIME HANDLED COUNT	4	Numeric	Handled count: in other words, the total number of times a TIME event caused one (or more) Rule to fire
TIME SUPPRESSED COUNT	4	Numeric	Total number of TIME events suppressed from their destination during the interval

Data Item Name	Max. Field Length	Type of Data	Description
TIME SCHEDULED EXEC COUNT	4	Numeric	Total number of EXECs scheduled by TIME events
TIME ALERTS CREATED COUNT	4	Numeric	Total number of ALERTs created by TIME events
TIME HARDCOPY LOG SUPPRESSED COUNT	4	Numeric	Total number of TIME events suppressed from the hardcopy log during the interval
ALRT	4	Character	Name of the action type (ALRT)
ALRT TOTAL	7	Numeric	Total number of ALRT events counted during the interval
ALRT HANDLED COUNT	7	Numeric	Handled count: in other words, the total number of times an ALRT event caused one (or more) Rule to fire
ALRT SUPPRESSED COUNT	7	Numeric	Total number of ALRT messages suppressed from their destination during the interval
ALRT SCHEDULED EXEC COUNT	7	Numeric	Total number of EXECs scheduled by ALRT events
ALRT ALERTS CREATED COUNT	7	Numeric	Total number of ALERTs created by ALRT events
ALRT HARDCOPY LOG SUPPRESSED COUNT	7	Numeric	Total number of ALRT messages suppressed from the hardcopy log during the interval
EXT	4	Character	Name of the action type (EXT)
EXT TOTAL	4	Numeric	Total number of EXT events counted during the interval
EXT HANDLED COUNT	4	Numeric	Handled count: in other words, the total number of times an EXT event caused one (or more) Rule to fire
EXT SUPPRESSED COUNT	4	Numeric	Total number of EXT messages suppressed from their destination during the interval
EXT SCHEDULED EXEC COUNT	4	Numeric	Total number of EXECs scheduled by EXT events
EXT ALERTS CREATED COUNT	4	Numeric	Total number of ALERTs created by EXT events
EXT HARDCOPY LOG SUPPRESSED COUNT	4	Numeric	Total number of EXT messages suppressed from the hardcopy log during the interval
IMS	4	Character	Name of the action type (IMS)
IMS TOTAL	8	Numeric	Total number of IMS events counted during the interval

Data Item Name	Max. Field Length	Type of Data	Description
IMS HANDLED COUNT	8	Numeric	Handled count: in other words, the total number of times an IMS event caused one (or more) Rule to fire
IMS SUPPRESSED COUNT	8	Numeric	Total number of IMS messages suppressed from their destination during the interval
IMS SCHEDULED EXEC COUNT	8	Numeric	Total number of EXECs scheduled by IMS events
IMS ALERTS CREATED COUNT	8	Numeric	Total number of ALERTs created by IMS events
IMS HARDCOPY LOG SUPPRESSED COUNT	8	Numeric	Total number of IMS messages suppressed from the hardcopy log during the interval
CICS	4	Character	Name of the action type (CICS)
CICS TOTAL	6	Numeric	Total number of CICS events counted during the interval
CICS HANDLED COUNT	6	Numeric	Handled count: in other words, the total number of times a CICS event caused one (or more) Rule to fire
CICS SUPPRESSED COUNT	6	Numeric	Total number of CICS messages suppressed from their destination during the interval
CICS SCHEDULED EXEC COUNT	6	Numeric	Total number of EXECs scheduled by CICS events
CICS ALERTS CREATED COUNT	6	Numeric	Total number of ALERTs created by CICS events
CICS HARDCOPY LOG SUPPRESSED COUNT	6	Numeric	Total number of CICS messages suppressed from the hardcopy log during the interval
DB2	4	Character	Name of the action type (DB2)
DB2 TOTAL	8	Numeric	Total number of DB2 events counted during the interval
DB2 HANDLED COUNT	8	Numeric	Handled count: in other words, the total number of times a DB2 event caused one (or more) Rule to fire
DB2 SUPPRESSED COUNT	8	Numeric	Total number of DB2 messages suppressed from their destination during the interval
DB2 SCHEDULED EXEC COUNT	8	Numeric	Total number of EXECs scheduled by DB2 events
DB2 ALERTS CREATED COUNT	8	Numeric	Total number of ALERTs created by DB2 events
DB2 HARDCOPY LOG SUPPRESSED COUNT	8	Numeric	Total number of DB2 messages suppressed from the hardcopy log during the interval

Data Item Name	Max. Field Length	Type of Data	Description
JES3	4	Character	Name of the action type (JES3)
JES3 TOTAL	6	Numeric	Total number of JES3 events counted during the interval
JES3 HANDLED COUNT	6	Numeric	Handled count: in other words, the total number of times a JES3 event caused one (or more) Rule to fire
JES3 SUPPRESSED COUNT	6	Numeric	Total number of JES3 messages suppressed from their destination during the interval
JES3 SCHEDULED EXEC COUNT	6	Numeric	Total number of EXECs scheduled by JES3 events
JES3 ALERTS CREATED COUNT	6	Numeric	Total number of ALERTs created by JES3 events
JES3 HARDCOPY LOG SUPPRESSED COUNT	6	Numeric	Total number of JES3 messages suppressed from the hardcopy log during the interval
ALRM	4	Character	Name of the action type (ALRM)
ALRM TOTAL	6	Numeric	Total number of ALRM events from the MAINVIEW Alarm Manager component counted during the interval
ALRM HANDLED COUNT	6	Numeric	Handled count: in other words, the total number of times an ALRM event caused one (or more) Rule to fire
RESERVED FIELD	6	Numeric	This field is always 0 and no data is collected
ALRM SCHEDULED EXEC COUNT	6	Numeric	Total number of EXECs scheduled by ALRM events
ALRM ALERTS CREATED COUNT	6	Numeric	Total number of ALERTs created by ALRM events
RESERVED FIELD	6	Numeric	This field is always 0 and no data is collected
VAR	4	Character	Name of the action type (VAR)
VAR TOTAL	6	Numeric	Total number of VAR events counted during the interval. VAR events are counted by the Automation Reporter **only if** at least one VAR-initiated Rule is enabled during the collection interval. If no VAR-initiated Rules are enabled during the interval, this field will always equal zero (0).
VAR HANDLED COUNT	6	Numeric	Handled count: in other words, the total number of times a VAR event caused one (or more) Rule to fire
RESERVED FIELD	6	Numeric	This field is always 0 and no data is collected

Data Item Name	Max. Field Length	Type of Data	Description
VAR SCHEDULED EXEC COUNT	6	Numeric	Total number of EXECs scheduled by VAR events
VAR ALERTS CREATED COUNT	6	Numeric	Total number of ALERTs created by VAR events
RESERVED FIELD	6	Numeric	This field is always 0 and no data is collected
MQS	4	Character	Name of the action type (MQS)
MQS TOTAL	6	Numeric	Total number of MQS events counted during the interval. MQS events are counted by the Automation Reporter **only if** the BBI-SS PAS is connected to an MQS queue manager. If the BBI-SS PAS is not connected to an MQS queue manager, this field will always equal zero (0).
MQS HANDLED COUNT	6	Numeric	Handled count: in other words, the total number of times an MQS event caused one (or more) Rule to fire
MQS SUPPRESSED COUNT	6	Numeric	Total number of MQS messages suppressed from their destination during the interval
MQS SCHEDULED EXEC COUNT	6	Numeric	Total number of EXECs scheduled by MQS events
MQS ALERTS CREATED COUNT	6	Numeric	Total number of ALERTs created by MQS events
RESERVED FIELD	6	Numeric	This field is always 0 and no data is collected

Following is a sample data record for ACTN for 2 intervals.

```
/* SAMPLE RECORD FOR TYPE ACTN, SUB TYPE ONLY        */
SYSC,AOIA,410,BBPLEX01,"19970528","19:00",10,ACTN,ONLY,"19970528","19:00",
    10,6,MSG,65,0,0,0,0,0,CMD,34,0,0,0,0,0,JRNL,1,0,0,0,0,0,TIME,0,0,0,0,0,0,
    ALRT,0,0,0,0,0,0,EXT,0,0,0,0,0,0,IMS,0,0,0,0,0,0,CICS,0,0,0,0,0,0,DB2,0,0,
    0,0,0,0,JES3,0,0,0,0,0,0,ALRM,0,0,0,0,0,0,VAR,0,0,0,0,0,0,MQS,0,0,0,0,0,0,EOR
SYSC,AOIA,410,BBPLEX01,"19970528","18:50",10,ACTN,ONLY,"19970528","18:50",
    10,6,MSG,53,0,0,0,0,0,CMD,24,0,0,0,0,0,JRNL,1,0,0,0,0,0,TIME,0,0,0,0,0,0,
    ALRT,0,0,0,0,0,0,EXT,0,0,0,0,0,0,IMS,0,0,0,0,0,0,CICS,0,0,0,0,0,0,DB2,0,0,
    0,0,0,0,JES3,0,0,0,0,0,0,ALRM,0,0,0,0,0,0,VAR,0,0,0,0,0,0,MQS,0,0,0,0,0,0,EOR
```

Figure 248. Sample Data Record for ACTN, Subtype=ONLY
Note that these records, though split to four lines in this figure, are really created as a single line of data.

Tracking AutoOPERATOR ALERTS (ALRT)

This table describes the data collected for the record type ALRT (AutoOPERATOR ALERTs), subtype ONLY. ALRT records are collected in intervals.

Possible Usage: The ALRT records contain eight fields of data for six ALERT priorities:

Priority Name	Priority Value
Critical	1
Major	2
Minor	3
Warning	4
Informational	5
Clear	6

With the data in these records, you can approximate the average age of an ALERT. This may be especially important data for your Critical ALERTs because you would not want to see Critical ALERTs out on your system for long periods of time.

To approximate the average age of an ALERT:

1. Select an ALERT priority: for example; CRITICAL ALERTs.
2. Find the ACTIVE COUNT value to see how many CRITICAL ALERTs were active during an interval. Suppose this value is 50.
3. Find the ACTIVE MINUTES values for these ALERTs to see how long the ALERTs was active. Suppose this value is 100.
4. Divide the time by the number of active ALERTs: 100/50.
5. The average age of a critical ALERT is two minutes.

Note: This value does not include ALERTs that may have been created and deleted during the interval.

Data Item Name	Max. Field Length	Type of Data	Description
ALERT RECORD COUNT	1	Numeric	Number of ALERT priority types within the record
ALERT RECORD DATA COUNT	1	Numeric	Number of fields of data for each ALERT priority
PRIORITY	8	Character	Description of the ALERT priority
PRIORITY	1	Numeric	A numeric representation of the ALERT priority
ACTIVE COUNT	4	Numeric	Total number of ALERTS active at the end of the interval ALERTs that were active and then deleted during the interval are **not included in this count.**

Data Item Name	Max. Field Length	Type of Data	Description
CREATED COUNT	4	Numeric	Total number of ALERTS created (during the interval)
DELETED COUNT	4	Numeric	Total number of ALERTS deleted (during the interval) The number of ALERTs that were active during the interval but deleted before the end of the interval are not included in the ACTIVE COUNT numbers.
ACTIVE MINUTES	4	Numeric	Total number of active minutes for ALERTs that are still active at the end of the interval (when the ACTIVE COUNT values are calculated) This value does not include ALERTs that may have been created and deleted during the course of the interval.
EXEC FOLLOW UP COUNT	4	Numeric	ALERTs with follow-up EXEC
SENT TO COMMAND/POST COUNT	4	Numeric	ALERTs sent to COMMAND/POST

Following is a sample data record for ALRT for four intervals.

```
/* SAMPLE RECORD FOR TYPE ALRT, SUB TYPE ONLY         */
SYSC,AOIA,410,BBPLEX01,"19970528","19:00",10,ALRT,ONLY,"19970528","19:00",
      6,8,CRITICAL,1,0,0,0,0,0,0,MAJOR,2,0,0,0,0,0,0,MINOR,3,0,0,0,0,0,0,
           WARN,4,2,0,0,971,0,0,INFO,5,0,0,0,0,0,0,CLEAR,6,0,0,0,0,0,0,EOR
SYSC,AOIA,410,BBPLEX01,"19970528","18:50",10,ALRT,ONLY,"19970528","18:50",
      6,8,CRITICAL,1,0,0,0,0,0,0,MAJOR,2,0,0,0,0,0,0,MINOR,3,0,0,0,0,0,0,
           WARN,4,2,0,0,951,0,0,INFO,5,0,0,0,0,0,0,CLEAR,6,0,0,0,0,0,0,EOR
SYSC,AOIA,410,BBPLEX01,"19970528","18:40",10,ALRT,ONLY,"19970528","18:40",
      6,8,CRITICAL,1,0,0,0,0,0,0,MAJOR,2,0,0,0,0,0,0,MINOR,3,0,0,0,0,0,0,
           WARN,4,2,0,0,931,0,0,INFO,5,0,0,0,0,0,0,CLEAR,6,0,0,0,0,0,0,EOR
SYSC,AOIA,410,BBPLEX01,"19970528","18:30",10,ALRT,ONLY,"19970528","18:30",
      6,8,CRITICAL,1,0,0,0,0,0,0,MAJOR,2,0,0,0,0,0,0,MINOR,3,0,0,0,0,0,0,
            WARN,4,2,0,0,911,0,0,INFO,5,0,0,0,0,0,0,CLEAR,6,0,0,0,0,0,0,EOR
```

Figure 249. Sample Data Record for ALRT, Subtype=ONLY
Note that these records, though split to three lines in this figure, are really created as a single line of data.

Tracking Events (EVENT)

This table describes the data collected for the record type EVNT (AutoOPERATOR Events), subtype ONLY. There is no subtype for this record (denoted as SUBTYPE=ONLY). EVNT records are collected cumulatively.

Possible usage: Use the two fields EVENT COUNT and HANDLED COUNT of this data record to track information about how many Events occur and how many get handled (or automated) over periods of time.

For example, you might gather data once a day for several days before you implement new or additional automation and collect data after additional automation is implemented. By comparing the differences in these two fields you can see how the new automation is increasing the number of handled events.

You also can use this record to determine the sum of handled Events (HANDLED COUNT) and divide that by the EVENT COUNT, yielding a percentage of the Event activity handled by automation.

If this percentage drops sharply, you'll know to check if you have added an application that produces a lot of Events that are not handled by automation.

Data Item Name	Max. Field Length	Type of Data	Description
TEXT-ID	8	Character	Text ID (or first word) of the Event
EVENT COUNT	4	Numeric	Total number of Event occurrences
HANDLED COUNT	4	Numeric	Total number of times the Event was handled (in other words, the number of times the Event triggered and fired a Rule)
EVENT TYPE	4	Character	Type (or origin) of the Event
PARTIAL TEXT	50	Character	Partial Event text (contains the first 50 characters of the Event text)
PEAK ARRIVAL	4	Numeric	Peak arrival rate of the Event per interval

Following is a sample data record for EVNT for two data collection intervals.

```
/* SAMPLE RECORD FOR TYPE EVNT, SUB TYPE ONLY        */
SYSC,AOIA,410,BBPLEX01,"19970529","08:13",76,EVNT,ONLY,"19970529","09:29",
     "PM0221I",1,0,JRNL,"NO MULTIPLE SYSTEM APPLID SPECIFIED",27,EOR
SYSC,AOIA,410,BBPLEX01,"19970529","08:13",76,EVNT,ONLY,"19970529","09:29",
     "PM0216I",5,0,MSG,"BBI CONNECTIONS COMPLETED FOR 000 LINK(S)JO2X",27,EOR
```

Figure 250. Sample Data Record for EVNT, Subtype=ONLY
Note that these records, though split to two lines in this figure, are really created as a single line of data.

Tracking AutoOPERATOR EXECs (EXEC)

This table describes the data collected for the record type EXEC (AutoOPERATOR EXECs), subtype ONLY. There is no subtype for this record (denoted as SUBTYPE=ONLY). EXEC records are collected cumulatively.

Possible usage: With EXEC data records, you might save and track only those that have at least one significant value. For example, you might be interested in those records that show how many times a specific EXEC abended (ABEND COUNT field). A recursive abend in a service EXEC can seriously damage automation production and it may be difficult to detect this kind of problem during an online session.

Another important piece of information might be how many EXECs were dropped or not executed. The DROP COUNT field shows the number of EXECs that have not been processed, which can be very detrimental if you have automation depending on those EXECs. You may need to increase the number of threads available for EXEC processing, serialize some processes, or even defer some processes.

This record also can be used to analyze EXECs that have a high RUN COUNT. You may want to try to convert EXECs that are used frequently to another form of automation such as Rules. Rules generally use less CPU seconds to run and are easier to maintain.

Data Item Name	Max. Field Length	Type of Data	Description
EXEC NAME	8	Character	Name of the EXEC
RUN COUNT	4	Numeric	Total number of times the EXEC was run
TOTAL CPU SECONDS	4	Numeric (seconds)	Total seconds of CPU consumed by EXEC
MAX CPU SECONDS	8	Numeric (seconds)	Maximum amount of CPU used by this EXEC
ABEND COUNT	4	Numeric	Total number of times an EXEC abended
DROP COUNT	4	Numeric	Total number of EXECs dropped
QUEUED COUNT	4	Numeric	Total number of times this EXEC was queued
MAXIMUM QUEUED COUNT	4	Numeric	Maximum number of queued EXECs

Following is a sample data record for EXEC for two data collection intervals.

```
/* SAMPLE RECORD FOR TYPE EXEC, SUB TYPE ONLY        */
SYSC,AOIA,410,BBPLEX01,"19970529","08:12",77,EXEC,ONLY,"19970529",
"09:29",ARTHRESH,5,4,1,0,0,0,0,EOR
SYSC,AOIA,410,BBPLEX01,"19970529","08:12",77,EXEC,ONLY,"19970529",
"09:29",MSGMULEB,192,13,0,0,0,0,0,EOR
```

Figure 251. Sample Data Record for EXEC, Subtype=ONLY

Tracking AutoOPERATOR Rules Activity (RULE)

This table describes the data collected for the record type RULE (AutoOPERATOR Rules), subtype ONLY. These records are collected cumulatively.

Possible Usage: By selecting and sorting the RULE data records for a specific Rule, you can find out how often a Rule is fired and then move Rules that are fired more often to be higher up in the Rule Set or even into a new Rule Set. Use a search strategy of FIRST and you can cut down on the amount of search time used for every incoming Event.

For example, if you sort the records collected over a period of days or weeks based on the FIRED COUNT field, you will be able to determine which Rules get fired most often. Move these Rules up in the Rule Set (or into a new Rule Set) and use an Automation Strategy of FIRST, and you'll lessen the time used for searching for a match between incoming Events and Rules.

Data Item Name	Max. Field Length	Type of Data	Description
RULE NAME	8	Character	Rule name (Rule ID)
FIRED COUNT	4	Numeric	Fired count of the Rule (in other words, how many times the Rule fired during the interval)
SET NAME	8	Character	Rule Set name that the Rule belongs to
GROUP	8	Character	Rule documentation field: Group
FUNCTION	8	Character	Rule documentation field: Function
CODE	2	Character	Rule documentation field: Code
EXEC NAME	8	Character	Name of the EXEC that was scheduled by the Rule (if applicable)
EVENT TYPE	4	Character	Event type (ORIGIN) of the Rule

Following is a sample data record for RULE for four data collection intervals.

```
/* SAMPLE RECORD FOR TYPE RULE, SUB TYPE ONLY        */
SYSC,AOIA,410,BBPLEX01,"19970529","09:55",11,RULE,ONLY,"19970529","10:06",
                                "ARSTART",1,AAORULPM,REPORTER,START,,,JRNL,EOR
SYSC,AOIA,410,BBPLEX01,"19970529","09:55",11,RULE,ONLY,"19970529","10:06",
                                "RUL00001",0,AAORULPM,,,,,JRNL,EOR
SYSC,AOIA,410,BBPLEX01,"19970529","09:55",11,RULE,ONLY,"19970529","10:06",
                                "RUL00002",1,AAORULPM,,,,,JRNL,EOR
SYSC,AOIA,410,BBPLEX01,"19970529","09:55",11,RULE,ONLY,"19970529","10:06",
                                 "ARGPTRS",1,AAORULPM,REPORTER,,,,JRNL,EOR
```

Figure 252. Sample Data Record for RULE, Subtype=ONLY
Note that these records, though split to two lines in this figure, are really created as a single line of data.

Tracking AutoOPERATOR Rule Sets (RULESET: AUTO)

This table describes the data collected for the record type RULESET (AutoOPERATOR Rule Sets), subtype AUTO. These records are collected cumulatively.

Data Item Name	Max. Field Length	Type of Data	Description
AUTOMATION STRATEGY	10	Character	Automation strategy
HONOR MPF	3	Character	Whether HONORMPF was set to YES or NO
AUTOMATION STATUS	8	Character	Current automation status (ACTIVE/INACTIVE)

Following is a sample data record for RULESET, subtype AUTO, for four data collection intervals.

```
/* SAMPLE RECORD FOR TYPE RULESET, SUB TYPE AUTO     */
SYSC,AOIA,410,BBPLEX01,"19970529","08:12",77,RULESET,AUTO,"19970529","09:29",
   INDIVIDUAL,NO,ACTIVE,EOR
SYSC,AOIA,410,BBPLEX01,"19970529","09:55",11,RULESET,AUTO,"19970529","10:06",
   INDIVIDUAL,NO,ACTIVE,EOR
```

Figure 253. Sample Data Record for RULESET, Subtype=AUTO

Tracking AutoOPERATOR Rule Sets (RULESET: SET)

This table describes the data collected for the record type RULESET (AutoOPERATOR Rule Sets), subtype SET. These records are collected cumulatively.

Possible Usage: Use this record to determine ratios such as how many messages were seen by the Rules Processor versus how many of theses messages were handled by Rules.

To do this, examine the value in the FILTER COUNT field which shows the number of events that match Rule Set Selection Criteria. Compare this against the FIRED COUNT field. What you see is the total number of events seen by the AutoOPERATOR Rules Processor and the number of messages that were handled by Rules. Based on these numbers, you can better analyze if you need to write more Rules to handle events coming through the Rules Processor.

You also can look at the FILTER COUNT and FIRED COUNT fields to determine if you need to alter the order of your Rule Sets. If you find high numbers in the FILTER COUNT field and low numbers (or even zeroes) in the FIRED COUNT field, what you are seeing is a lot of searching through your Rule Sets but very few matches to Rules.

A possible remedy for this type of situation is to use Rule Set filtering criteria as described in “Filtering Events for Rule Sets” on page 188. This would prevent the scanning of entire Rule Sets for an event that does not match any events in the Ruleset. Or you can rearrange your Rule Sets to lessen the search time.

Data Item Name	Max. Field Length	Type of Data	Description
SET NAME	8	Character	Rule Set name
RULE COUNT	2	Numeric	Total number of Rules in the Rule Set
FIRED COUNT	4	Numeric	Total number of times that Rules within the Rule Set were fired during the interval
FILTER COUNT	8	Numeric	Total number of events that match and pass through filter criteria
SET ENABLE DATE	8	DATE "YYYYMMDD"	Rule Set ENABLED date
SET ENABLE TIME	5	TIME "HH:MM"	Rule Set ENABLED time
SET STRATEGY	8	Character	Rule Set Strategy; can be: FIRST, ALL, INDIVIDUAL, or QUALIFIED

Following is a sample data record for RULESET, subtype SET, for three data collection intervals.

```
/* SAMPLE RECORD FOR TYPE RULESET, SUB TYPE SET      */
SYSC,AOIA,410,BBPLEX01,"19970529","08:12",77,RULESET,SET,"19970529",
                     "09:29",AAORULJB,34,192,2198,"19970529","08:13",FIRST,EOR
SYSC,AOIA,410,BBPLEX01,"19970529","09:55",11,RULESET,SET,"19970529",
                     "10:06",AAORULPM,5,1,760,"19970529","09:57",ALL,EOR
SYSC,AOIA,410,BBPLEX01,"19970529","09:55",11,RULESET,SET,"19970529",
                      "10:06",RULAR001,6,0,760,"19970529","09:57",ALL,EOR
```

Figure 254. Sample Data Record for RULESET, Subtype=SET
Note that these records, though split to two lines in this figure, are really created as a single line of data.

Tracking Resources (RES)

This table shows the data collected for the record type RESOURCE. There is no subtype for this record (denoted as SUBTYPE=ONLY). RES records are collected in intervals.

Data Item Name	Max. Field Length	Type of Data	Description
PRIVATE STORAGE BELOW	16	Numeric	Private storage allocated by the SS (bytes current allocation) at the end of the interval
PRIVATE STORAGE ABOVE	16	Numeric	Private storage allocated by the SS (bytes current allocation) at the end of the interval
CSA PQE COUNT	8	Numeric	Total number of low storage PQEs at the end of the interval
CSA PQE BYTES	16	Numeric	Bytes of low storage in use by PQEs at the end of the interval
ECSA PQE COUNT	8	Numeric	Total number of high storage PQEs at the end of the interval
ECSA PQE BYTES	16	Numeric	Bytes of high storage in use by PQEs at the end of the interval
TOTAL CSA PQES	8	Numeric	Count of PQEs created this interval
TOTAL ECSA PQES	8	Numeric	Count of extended PQEs created this interval
VARIABLE COUNT CSA	8	Numeric	Total number of variables in CSA at the end of the interval
VARIABLE BYTES CSA	16	Numeric	Total number of bytes of CSA storage in use by variables at the end of the interval
NEW CSA VARIABLES COUNT	8	Numeric	New variables created in CSA storage during this interval
SUBSYSTEM CPU SECONDS	8	Numeric (seconds)	Subsystem CPU seconds used this interval
SSR SECONDS	8	Numeric (seconds)	Subsystem-request CPU seconds requested by address spaces which fired Rules during the interval
EXCP COUNT	8	Numeric	EXCP count this interval
TS USERS	4	Numeric	Total number of TS users at data collection time
TRANSACTION COUNT	6	Numeric	Transactions issued this interval by terminal session users
CPUS ONLINE	2	Numeric	Total number of CPUs online at data collection time

Following is a sample data record for RES for two intervals.

```
/* SAMPLE RECORD FOR TYPE RES, SUB TYPE ONLY         */
SYSC,AOIA,410,BBPLEX01,"19970528","18:50",10,RES,ONLY,"19970528","18:50",
                       1847296,937984,0,0,0,0,0,277,47,4413,0,1,2,207,0,0,02,EOR
SYSC,AOIA,410,BBPLEX01,"19970528","18:40",10,RES,ONLY,"19970528","18:40",
                        1847296,937984,0,0,0,0,0,544,47,4413,0,2,7,207,0,0,02,EOR
```

Figure 255. Sample Data Record for RES, Subtype=ONLY
Note that these records, though split to two lines in this figure, are really created as a single line of data.

Tracking Tape Device Sharing Activity (TAPE: LOCAL)

This table shows the data collected for the record type TAPE (TapeSHARE for AutoOPERATOR component), subtype LOCAL. These records are collected in intervals.

One LOCAL record is created per interval. If 10 partners are participating in tape sharing activity with TapeSHARE at the end of the interval, you must save and collect the Automation Reporter TAPE: LOCAL records from each partner individually.

Therefore, note that the information described here is for a single partner in a TapeSHARE PLEX where TapeSHARE is performing tape device sharing automation for a group of partners.

Data Item Name	Max. Field Length	Type of Data	Description
ALLOCATION REQUESTS	12	Numeric	Number of times this partner requested that TapeSHARE find one or more devices for the partner to use
ALLOCATION REQUESTS SATISFIED	12	Numeric	Number of times that TapeSHARE was able to satisfy this partner's request for a device
TOTAL DEVICE REQUESTS	12	Numeric	Number of devices that TapeSHARE obtained from other partners for this partner to use
TAKES NOT GIVEN TO THE REQUESTING JOB	12	Numeric	Number of times a device was TAKEN from another partner but not GIVEN to the requesting job In other words it's the number of times the partner requested a device and a device was found but the partner did not end up receiving the device. This could happen because the request timed out before the partner could actually TAKE the device.
TOTAL TIME NEEDED FOR A TAKE	12	Numeric	Amount of time (in seconds) this partner spent waiting for a device
REQUEST FOR GIVES	12	Numeric	Number of times other partners requested to share devices from this partner
GIVES	12	Numeric	Number of times this partner actually GAVE devices to other partners
TOTAL DEVICES GIVEN	12	Numeric	Number of devices that were GIVEN by this partner to other partners
SUCCESSFUL TAKE: TOTAL TIME	12	Numeric	Amount of time (in seconds) that a partner spent asking for devices and receiving them
SUCCESSFUL TAKE: RETRY TIME	12	Numeric	Amount of time (in seconds) spent in retrying to complete a successful TAKE
SUCCESSFUL TAKE: RETRY COUNT	12	Numeric	Number of retry attempts that were made and a successful TAKE was completed

Data Item Name	Max. Field Length	Type of Data	Description
FAILED TAKE: TOTAL TIME	12	Numeric	Amount of time (in seconds) that a partner spent asking for devices and did not receive any devices
FAILED TAKE: RETRY TIME	12	Numeric	Amount of time (in seconds) spent in retrying to complete a TAKE but was unsuccessful
FAILED TAKE: RETRY COUNT	12	Numeric	Number of retry attempts that were made and a TAKE was unsuccessful
TIMED OUT: REQUESTS	12	Numeric	Number of requests that could not be satisfied because the request timed out

Following is a sample data record for TAPE: LOCAL for one interval.

```
/* SAMPLE RECORD FOR TYPE TAPE, SUB TYPE LOCAL        */
SYSC,DSHC,410,LOCAL,"19970620","18:00",10,TAPE,LOCAL,"19970620","18:00",
     1,15,0,0,0,0,0,0,0,0,0,0,0,0,0,120,DEFAULT,EOR
```

Figure 256. Sample Data Record for TAPE, Subtype=LOCAL
Note that these records, though split to two lines in this figure, are really created as a single line of data.

Tracking Tape Device Sharing Activity (TAPE: PLEX)

This table shows the data collected for the record type TAPE (TapeSHARE for AutoOPERATOR component), subtype PLEX. These records are collected in intervals.

This record contains information about partner to partner communication such as how often a partner requested devices from the other partners, how long it took for a device to be shared, and so on.

Data Item Name	Max. Field Length	Type of Data	Description
NUMBER OF PARTNERS	8	Numeric	Number of partners that were in the TapeSHARE PLEX when the data was collected For every partner in this record, statistics will be kept in subsections of this record. The PARTNER NODE NAME field separates each subsection. Each subsection has 9 fields of data (see the field for NUMBER OF FIELDS FOR THIS PARTNER). If this value is 6, there will be 6 subsections of data.
NUMBER OF FIELDS FOR THIS PARTNER	2	Numeric	Number of fields of data for this partner
PARTNER NODE NAME	8	Character	Node name of the partner
TARGET PARTNER NAME	8	Character	Name of the TapeSHARE partner
TARGET SMF NAME	8	Character	SMF system name of the target partner
SYSPLEX NAME	4	Character	MVS SYSPLEX name of this partner as defined in the ECVT control block
TARGET SSID NAME	4	Character	SSID (subsystem ID) of the target partner
AO VERSION LEVEL	5	Numeric	Version level of AutoOPERATOR running on the target partner This level is the same value as in the header portion of the record.
TAPESHARE LEVEL	1	Numeric	Internal level of TapeSHARE that is running on the target partner
TIME WAITED	12	Numeric	The amount of time (in seconds) spent waiting for a TAKE
TAKE REQUESTS	12	Numeric	The number of TAKE requests made from this target
SUCCESSFUL TAKES	12	Numeric	The number of successful TAKE requests made for this target

Data Item Name	Max. Field Length	Type of Data	Description
GIVE REQUESTS	12	Numeric	The number of GIVE requests made from this target
SUCCESSFUL GIVES	12	Numeric	The number of successful GIVE requests made for this target

Following is a sample data record for TAPE: PLEX for one interval.

```
/* SAMPLE RECORD FOR TYPE TAPE, SUB TYPE PLEX        */
SYSC,DSHC,410,LOCAL,"19970620","18:00",10,TAPE,PLEX,"19970620","17:50",
   2,12,"SYSG","SYSG","$GCIS","$GCIS","$GCIS","410",0,0,1,1,3,1,
   "SYSF","SYSF","$FCIS","$FCIS","$FCIS","410",0,0,1,0,0,0,EOR
```

Figure 257. Sample Data Record for TAPE, Subtype=PLEX
Note that these records, though split to two lines in this figure, are really created as a single line of data.

Chapter 14. Managing EXECs Using the EXEC Management Application

The EXEC Management application allows you to enable, disable, schedule, browse, and test the AutoOPERATOR EXECs that you have stored in your SYSPROC or SYSEXEC concatenation.

Important
AutoOPERATOR EXECs where you have specified: `DISP(NO)` in the documentation box of the EXEC **do not** appear in this application.

For more information about AutoOPERATOR EXECs, including documentation boxes, refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.

Introduction

This chapter describes the EXEC Management application and how you can use it. The EXEC Management application allows you to:

- Schedule an EXEC to run
- Select an EXEC for testing
- Browse the contents of an EXEC
- Enable and disable currently executing EXECs

The EXEC Management application also allows you to see things such as:

- How many EXECs you have currently executing
- Which EXECs are currently executing
- How many EXECs you have queued to Normal or High priority queues
- How many times a specific EXEC has executed since the last subsystem start
- How much CPU time a specific EXEC has used
- How many times a specific EXEC has abended

Every time an EXEC is executed in AutoOPERATOR, CPU consumption statistics are saved. The only conditions where statistics are not kept for a specific EXEC is where the EXEC is initiated using:

- The TSO command EX (for example, `TSO EX EXECABC`)
- The REXX CALL command
- The CLIST SYSCALL command

The CPU time used for these EXECs is added to the calling EXEC.

Individual statistics also are not kept for EXECs called as external REXX functions.

Accessing the EXEC Management Application

This section describes the EXEC Management application and its two views, which you can see by scrolling left and right. “Displaying Currently Active EXECs” on page 466 describes the application when you use the EXPAND primary command to see statistics for only currently active EXECs.

Select Option 8 from the Primary Option Menu and Option 7 from the Automation Menu to display the EXEC Management panel as shown in Figure 258.

```
BMC Software ---------------- EXEC Management ----------------- AutoOPERATOR
COMMAND ===>                                                     TGT ===>  FK5X
INTERVAL ==> 3                                                   DATE --- 01/03/16
STATUS   --- INPUT              Scroll  right/left               TIME --- 16:04:59
Primary command: Sort
         EXECs defined   418     Scheduled    11            Enabled   418
 High Priority running     0        Queued     0
 Norm Priority running     0        Queued     0

 PRESS EXPAND TO VIEW EXEC ACTIVITY
 (B)ROWSE, (E)NABLE, (D)ISABLE, (S)ELECT EXEC FOR EXECUTION, (T)EST EXEC

LC  NAME     STATUS   GROUP    FUNCTION CODE AUTHOR   DESCRIPTION
    ________ _        ________ ________ __   ________ ___________________________
___ $AAA     ENABLED  TEST     QUEUE         CIM4X    TEST THRESHOLD
___ $COLORS  ENABLED  DEMO     LOOP          CIM4     TEST SELECT EXEC WITHIN A
___ $DFS970I ENABLED                         $AOSRLO
___ $HASP373 ENABLED                         JDB1
___ $HASP605 ENABLED                         CIM4X
___ $IM9175I ENABLED                         CIM4X
___ $SEND    ENABLED                         CIM4
___ @CATCHU1 ENABLED                         CIM1
___ @CHE     ENABLED                         CIM4X
```

Figure 258. EXEC Management Panel with EXEC Description—Documentation View

The middle portion of the panel shows statistics about all the EXECs. Refer to “Example of the Status Area of the EXEC Management Panel” on page 460 for more information about these fields.

The bottom portion of the panel shows the documentation information that was included when the EXEC was first coded.

You can qualify the display to show subsets of the list with the input fields below each column heading. For example, you can enter `D` in the `STATUS`, and the display shows only those EXECs that are DISABLED.

Scroll right to see more information about the EXECs, such as how many times an EXEC has executed, how much CPU it has used, how many times it has abended, and so on.

Scrolling Right to Display Additional Information

Figure 259 shows an example of the panel when you scroll right.

```
BMC Software ---------------- EXEC Management ----------------- AutoOPERATOR
COMMAND ===>                                                    TGT ===>  FK5X
INTERVAL ==> 3                                                  DATE --- 01/0316
STATUS   --- INPUT              Scroll  right/left              TIME --- 16:05:04
Primary command: Sort
         EXECs defined   418     Scheduled    11            Enabled   418
 High Priority running     0        Queued     0
 Norm Priority running     0        Queued     0

 PRESS EXPAND TO VIEW EXEC ACTIVITY
 (B)ROWSE, (E)NABLE, (D)ISABLE, (S)ELECT EXEC FOR EXECUTION, (T)EST EXEC

LC  NAME     STATUS    EXECD   ABEND  TOTCPU  AVGCPU  MAXCPU     CHANGED
    ________ _
___ $AAA     ENABLED                                              00/11/04 16:14
___ $COLORS  ENABLED                                              00/10/25 17:33
___ $DFS970I ENABLED                                              00/11/01 15:54
___ $HASP373 ENABLED                                              00/03/22 15:00
___ $HASP605 ENABLED                                              00/07/30 15:01
___ $IM9175I ENABLED                                              00/11/18 18:38
___ $SEND    ENABLED                                              00/01/25 15:28
___ @CATCHU1 ENABLED                                              00/08/13 23:35
___ @CHE     ENABLED        1                       .08      .08  00/03/12 19:19
___ @DAILY   ENABLED                                              00/04/12 11:49
```

Figure 259. EXEC Management Panel with EXEC Statistics—Resource View

This panel shows individual statistics for each EXEC, such as how many times an EXEC was executed, how many times it abended, and so on.

Similar to View 1 of the panel, you can use the input fields under the column headings to qualify the display and show a subset of all the EXECs.

Viewing All Currently Active EXECs: Using the EXPAND Command

The EXEC Management application also has a panel that displays information for all currently executing EXECs. To see this panel, use the EXPAND primary command to display only those EXECs that are currently running and track the progress of those EXECs.

“Displaying Currently Active EXECs” on page 466 contains more information about using this panel. The EXPAND command toggles the display between the EXEC Management panels.

Using Primary Commands

You can use the following primary commands on the COMMAND line of this panel.

Command	Description
EXPAND	Toggles the display between the EXEC Description portion of the panel and the EXEC Activity portion that shows only the EXECs currently running
GO	Starts the panel refresh
SORT	Allows you to sort the display of EXECs in either ascending or descending order The default is descending for all numeric fields. For example, all values in the `SCHED` field are sorted with the largest values at the top of the display. The default is ascending for all alphanumeric fields.

Describing Fields in the EXEC Status Area

Following is a description of the status area of the EXEC Management panel (see Figure 260). This area appears on all EXEC Management panels.

```
          EXECs defined   465     Scheduled   115        Enabled   465
  High Priority running     0        Queued   0
  Norm Priority running     0        Queued   0
```

Figure 260. Example of the Status Area of the EXEC Management Panel

Field Name	Description
EXECs defined	Is the total number of EXECs that are written to the SYSPROC concatenation. This number includes EXEC that specify `DISP(NO)` in the documentation box but these EXECs are not displayed in the bottom portion of the panel.
Scheduled	Is the number of all the EXECs which have been scheduled to run since the last subsystem start. A REXX program executed through a CALL statement is not counted.
Enabled	Is the number of EXECs enabled.
High Priority	Shows information about EXECs scheduled to run in high priority; for example: **Running** Number of EXECs currently executing on high priority threads **Queued** Number of EXECs queued to run on high priority threads

Norm Priority Shows information about EXECs scheduled to run in normal priority; for example:

Running Number of EXECs currently executing on normal priority threads

Queued Number of EXECs queued to run on normal priority threads

Describing Fields in the EXEC Description Area

The bottom portion of the panel contains one of the following items list:

- A list of defined EXECs and their EXEC Descriptions (as seen in View 1 in Figure 261)
- A list of defined EXECs and information regarding how much CPU each EXEC has used, how many times an EXEC has abended, and so on (as seen in View 2 in Figure 262)
- A list of currently active EXECs and related statistics (if you use the EXPAND command); see page 457

Describing Fields in View 1: The EXEC Description area fields of View 1 are shown in Figure 261.

```
 PRESS EXPAND TO VIEW EXEC ACTIVITY
 (B)ROWSE, (E)NABLE, (D)ISABLE, (S)ELECT EXEC FOR EXECUTION, (T)EST EXEC

LC  NAME     STATUS   GROUP    FUNCTION CODE AUTHOR   DESCRIPTION
    ________ _        ________ ________ __   ________ ___________________________
___ $AAA     ENABLED  TEST     QUEUE         CIM4X    TEST THRESHOLD
___ $COLORS  ENABLED  DEMO     LOOP          CIM4     TEST SELECT EXEC WITHIN A
___ $DFS970I ENABLED                         $AOSRLO
___ $HASP373 ENABLED                         JDB1
___ $HASP605 ENABLED                         CIM4X
___ $IM9175I ENABLED                         CIM4X
___ $SEND    ENABLED                         CIM4
___ @CATCHU1 ENABLED                         CIM1
___ @CHE     ENABLED                         CIM4X
```

Figure 261. Bottom Portion of EXEC Management Panel—Documentation View

The first line of this area:

```
PRESS EXPAND TO VIEW EXEC ACTIVITY
```

indicates you can use the EXPAND command to view the EXEC activity of all currently active EXECs. Use the EXPAND command by assigning a PF key to EXPAND, or typing EXPAND in the COMMAND line and pressing ENTER.

You can enter the following line commands in the LC column.

Command	Description
BROWSE|B	Allows you to browse the complete contents of an EXEC
ENABLE|E	Allows you to enable the EXEC
DISABLE|D	Allows you to disable the EXEC
SELECT|S	Allows you to submit the EXEC and, optionally, specify parameters
TEST|T	Allows you to select the EXEC for testing with the EXEC Testing application Refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs* for more information about EXEC Testing.

The EXEC Description area fields of View 1 are:

Field Name	Description
LC	Allows you to enter a one character line command.
NAME	Is the name of the EXEC (member name in SYSPROC). You can qualify the display by EXEC name.
STATUS	Is the current status of the EXEC: can be either ENABLED or DISABLED.
GROUP	Is the group associated with the EXEC as specified in the EXEC documentation section.
FUNCTION	Is the function associated with the EXEC as specified in the EXEC documentation section.
CODE	Is the code associated with the EXEC as specified in the EXEC documentation section.
AUTHOR	Is the author associated with the EXEC as specified by the ISPF statistics. If an AUTHOR is specified in the documentation section, then this name overrides the ISPF data.
Description	Is the description associated with the EXEC as specified in the EXEC documentation section.

Describing Fields in View 2: The EXEC Description area fields of View 2 are shown in Figure 262.

```
 PRESS EXPAND TO VIEW EXEC ACTIVITY
 (B)ROWSE, (E)NABLE, (D)ISABLE, (S)ELECT EXEC FOR EXECUTION, (T)EST EXEC

LC  NAME     STATUS    EXECD   ABEND  TOTCPU   AVGCPU    MAXCPU      CHANGED
    ________ _
___ $AAA     ENABLED                                                 00/11/04 16:14
___ $COLORS  ENABLED                                                 00/10/25 17:33
___ $DFS970I ENABLED                                                 00/11/01 15:54
___ $HASP373 ENABLED                                                 00/03/22 15:00
___ $HASP605 ENABLED                                                 00/07/30 15:01
___ $IM9175I ENABLED                                                 00/11/18 18:38
___ $SEND    ENABLED                                                 00/01/25 15:28
___ @CATCHU1 ENABLED                                                 00/08/13 23:35
___ @CHE     ENABLED        1                      .08      .08     00/03/12 19:19
___ @DAILY   ENABLED                                                 00/04/12 11:49
```

Figure 262. Bottom Portion of EXEC Management Panel—Resource View

The EXPAND command and the line commands are explained in “Describing Fields in View 1:” on page 462.

The EXEC Description area fields of View 2 are:

Field	Description
NAME	Is the name of the EXEC (member name in SYSPROC).
STATUS	Is the status of the EXEC, either ENABLED or DISABLED.
EXECD	Is the number of times the EXEC has completed execution. Each time an EXEC completes execution, this count is incremented. Also, if you have a situation where EXECA calls EXECB (with an IMFEXEC SELECT statement and WAIT(YES) is specified), both EXECA and EXECB are counted in the EXECD count when they complete.
ABEND	Is the number of times the EXEC has abended since the SS started.
TOTCPU	Is the sum of CPU time used for all executions of the EXEC since the SS was started. If the EXEC schedules another EXEC (with an IMFEXEC SELECT statement where WAIT(YES) is specified), then CPU collection for the first EXEC is suspended until the selected EXEC returns control. If a REXX EXEC executes another REXX EXEC using the REXX CALL facility, the CPU time is charged to the calling EXEC.

AVGCPU Is the value of the TOTCPU divided by the EXECD value (TOTCPU/EXECD) since the BBI-SS PAS was last started.

MAXCPU The greatest amount of CPU time the EXEC used during any single execution since the BBI-SS PAS was last started.

Changed The date and time that the EXEC was last changed. This information will not be shown if SPF STATS is turned off.

Displaying Currently Active EXECs

By using the EXPAND command, you can toggle between the two panel views shown in Figure 258 on page 458 and in Figure 259 on page 459 and Figure 263.

```
BMC Software ----------------- EXEC Management ----------------- AutoOPERATOR
COMMAND ===>                                                     TGT ===>  AOJB
INTERVAL ==> 3                                                   DATE --- 01/03/17
STATUS   --- INPUT              Scroll  right/left               TIME --- 16:11:49
Primary command: Sort
         EXECs defined   438     Scheduled    10            Enabled   438
 High Priority running     0        Queued     0
 Norm Priority running     4        Queued     0

                                     PRESS EXPAND TO VIEW EXEC DESCRIPTIONS
 (C)ANCEL EXEC

LC  NAME     SEQ.# TIME     ORIGIN   TYPE     STATE   CPU  CPU% CPUL   CPUL%

___ $TOTSIO      7 16:11:47 JDB1     NORMAL   RUNNING     0   0%     1   0%
___ EXECIO       8 16:11:48 JDB1     NORMAL   RUNNING     0   0%     1   0%
___ $TOTCPU      9 16:11:48 JDB1     NORMAL   RUNNING     0   0%     1   0%
___ $JOBCPU     10 16:11:48 JDB1     NORMAL   RUNNING     0   0%     1   0%
              ****************** END OF EXECS *****************
```

Figure 263. EXEC Management Display of All Currently Active EXECs.

This figure shows the EXEC Management panel when you enter the EXPAND primary command to show only the currently active EXECs.

The EXEC status area of this panel is the same as on other EXEC Management panels. Use this panel to see information about the currently active EXECs such as:

- What time the EXEC was executed
- The origin of the EXEC
- How much CPU the EXEC is using

Using Line Commands

You can enter the following line command in the LC field.

Command	Description
CANCEL|C	Cancel the selected EXEC

Describing Fields

Following is a description of the fields in the EXEC Activity area.

Field	Description
LC	Allows you to enter a one character line command.
NAME	Is the name of the EXEC (member name in SYSPROC).
SEQ#	Is the sequence number that identifies the EXEC.
TIME	Is the time the EXEC was initiated.
ORIGIN	Is the location where the EXEC was issued.
TYPE	Specifies whether the EXEC is scheduled to run on a Normal priority thread or a High priority thread.
STATE	Specifies if the EXEC is RUNNING or is ENQUEUED. You can cancel the EXEC only when it is RUNNING.
CPU	Is the amount of CPU time used by the EXEC.
CPU%	Is the percentage of CPU used by this EXEC during a 15 second interval.
CPUL	Is the maximum allotted time allowed for running this EXEC.
CPUL%	Is the maximum percentage of CPU allocated to this EXEC during a 15 second reporting interval.

Documenting Your EXECs

Use the EXEC information line to classify EXECs by GROUP, FUNCTION, CODE, DESCRIPTION, DISPLAY, or AUTHOR. Then you can structure solutions or EXECs to automate your data center. The EXEC information line is specified within each EXEC and has the following format:

```
/* DOC GROUP() FUNC() CODE() DESC() DISP() AUTHOR()*/
```

where:

GROUP Designates the first hierarchical level in 1- to 8 alphanumeric characters.

FUNC Designates the second hierarchical level in 1- to 8 alphanumeric characters.

DESC Provides a minimal description of the EXEC's purpose in 1- to 25 alphanumeric characters.

DISP Indicates whether or not the EXEC should appear in a formatted member list; Y or N.

CODE Is a two-character user assigned code.

AUTHOR Overrides the ISPF author statistic in 1- to 8-alphanumeric characters.

When an EXEC has a /*DOC statement, the EXEC Management application displays the values specified in the EXEC Description display. The information line can be located anywhere in the EXEC. It can be split over any number of physical lines.

Chapter 15. General Services

The BBI General Services are common to all BMC Software products that use BBI-2. They are listed on the Primary Option Menu as:

```
⋮

 General Services:
   C  Service Refresh Cycle Setup        K  Current PF Key Assignments
   L  Display Journal                    T  Tutorials
   M  Display Messages and Codes         X  Terminate
⋮
```

Figure 264. General Services Options on the Primary Option Menu

Each of these options is described separately in the following sections.

Creating Cyclic Displays: CYCLE SETUP

Use the CYCLE SETUP option to specify 10 or less cyclic displays.

The SERVICE REFRESH CYCLE application is shown in Figure 265.

```
BMC Software -- -  --------- SERVICE REFRESH CYCLE ------ GENERAL SERVICES
COMMAND ===>                                              TGT ===> CICSPROD
                                                          PAGE     1  OF  1
                       TARGET   TYPE    DTIME  LOG   DESCRIPTION
SERVICE ===> MFSUT     IMSPROD  IMS       3          MFS POOL UTILIZATION
  PARMS ===>
SERVICE ===> DUSER     DB2A     DB2       3          DETAIL USER STATUS
  PARMS ===>
SERVICE ===> DA        SYSA     MAO       3          DISPLAY ACTIVE
  PARMS ===>
SERVICE ===> FILE      CICSPROD CICS      5          FILE DISPLAY
  PARMS ===> * OPEN
SERVICE ===> STAT      CICSP    CICS      3          SYSTEM STATUS
  PARMS ===>
SERVICE ===>
  PARMS ===>
SERVICE ===>
  PARMS ===>
SERVICE ===>
  PARMS ===>
SERVICE ===>
  PARMS ===>
SERVICE ===>
  PARMS ===>
```

Figure 265. SERVICE REFRESH CYCLE Panel

The input fields are defined as follows.

Input Field	Description
SERVICE	Enter the service select code, such as ALERTS or EMA. **Note:** If other products are installed, services and target types (AO, CICS, CAO, IAO, IMS, DB2, or MAO) can be intermixed. The available service select codes are listed in Table 56 on page 471.
TARGET	A 1- to 8 character identifier of the target. If TARGET and TYPE are not specified, the target displayed in the TGT field is used. If TARGET is not specified but TYPE is, the current target for the specified product line is used.
TYPE	The type of product line to process the requested service. A product line type does not need to be entered when only one product line is installed. If a product line is not specified, the product line where CYCLE was invoked is used. When multiple product lines are installed, a product line type must be specified if the service to be requested does not belong to the active product line.
DTIME	The number of seconds the specified service display is to be shown before the next display. If a time is not entered, the default specified by the INTERVAL parameter in the BBPROF data set member BBITSP00 is used and displayed in the DTIME field. The BMC Software distributed value is 3 seconds. If the maximum of 30 services is specified, each with a display time (DTIME) of 3 seconds, the first service in the refresh cycle is displayed approximately every 90 seconds.
LOG	(For IMF and MAINVIEW for DB2 services) Enter a Y (YES) or N (NO) to log screen images to the BBI-TS Image log for offline printing. The default, N, is displayed if no entry is made. A dash (—) displayed in this field indicates that Image logging is not supported for the requested service.

Specifying Application Names in CYCLE SETUP

You can display the following applications in cycle mode:

Table 56. Service Select Codes for Refresh Cycle SERVICE Field

Service Select Code	Application Description	Product Line (Type)
AutoOPERATOR Base Applications		
ALE|ALERTS	ALERTS Overview	AO
EMA	EXEC Management	AO
EXEC	EXEC Management	AO
RUL|RULES	Automation Control	AO
XAL|XALRTS	Alert Detail	AO
BBI (General) Applications		
JOU (to see the BBI Log Display)	LOG Display	
LOG	LOG Display	
CICS Operator Workstation (AutoOPERATOR) Applications		
STATUS CAO	CICS System Status	CAO
IMS Operator Workstation (AutoOPERATOR) Applications		
EX	Status/Exception	IAO
REGION	IMS Regions	IAO
STATUS	Status/Exception	IAO
MVS Operator Workstation (AutoOPERATOR) Applications		
DA	Address Spaces	MAO
DISP|DISPLAY	Address Spaces	MAO
ENQ|ENQUEUES	Enqueue/Reserve	MAO
OPER|OPERATOR	Operator Requests	MAO
OR	Operator Requests	MAO
REQ|REQUESTS	Operator Requests	MAO
RES|RESERVES	Enqueue/Reserve	MAO
STATUS	System Status	MAO
MAINVIEW for CICS Applications		
AB|ABEND	ABEND Display	CICS
AI|AID	AID Display	CICS

Table 56. Service Select Codes for Refresh Cycle SERVICE Field (Continued)

Service Select Code	Application Description	Product Line (Type)
AL|ALIAS	ALIAS Display	CICS
CLA|CLASSES	CLASSES Display	CICS
C|CONNECT	CONNECT Display	CICS
CONNX|CONNXPND	CONNXPND Display	CICS
CONNXPN2	CONNXPN2 Display	CICS
CONS|CONSOLES	CONSOLES Display	CICS
CST|CSTATUS	CSTATUS Display	CICS
DA|DATATABL	DATABL Display	CICS
DB2S|DB2SYSP	DB2SYSP Display	CICS
DB2T|DB2TASK	DB2TASK Display	CICS
DBC|DBCTL	DBCTL Display	CICS
DBCTA|DBCTASK	DBCTASK Display	CICS
DBCTT|DBCTTASK	DBCTTASK Display	CICS
DD|DDIR	DDIR Display	CICS
DDIRXPND	DDIRXPND Display	CICS
DDIRXPN2	DDIRXPN2 Display	CICS
DE|DEST	DEST Display	CICS
DL|DL/I	DL/I Display	CICS
DSA|DSAS	DSAS Display	CICS
DS|DSNAMES	DSNAMES Display	CICS
EN|ENQUEUE	ENQUEUE Display	CICS
EXI|EXITS	EXITS Display	CICS
F|FILE	FILE Display	CICS
FILEX|FILEXPND	FILEXPND Display	CICS
FILEXPN2	FILEXPN2 Display	CICS
G|GRAPH	GRAPH Display	CICS
I|ICE	ICE Display	CICS
J|JOURNAL	JOURNAL Display	CICS
LP|LPAS	LPAS Display	CICS
M|MONITOR	MONITOR Display	CICS
NUC|NUCLEUS	NUCLEUS Display	CICS

Table 56. Service Select Codes for Refresh Cycle SERVICE Field (Continued)

Service Select Code	Application Description	Product Line (Type)
PL|PLAN	PLAN Display	CICS
PLANX|PLANXPND	PLANXPND Display	CICS
PPST	PPST Display	CICS
P|PROBLEM	PROBLEM Display	CICS
PR|PROGRAM	PROGRAM Display	CICS
PSB|PSBNAME	PSBNAME Display	CICS
REG|REGIONS	REGIONS Display	CICS
REM|REMOTES	REMOTES Display	CICS
REV|REVIEW	REVIEW Display	CICS
SE|SESSIONS	SESSIONS Display	CICS
SH|SHARE	SHARE Display	CICS
ST|STATUS	STATUS Display	CICS
STE|STEPLIB	STEPLIB Display	CICS
S|SUBPOOL	SUBPOOL Display	CICS
SUF|SUFFIX	SUFFIX Display	CICS
SUM|SUMMARY	SUMMARY Display	CICS
T|TASK	TASK Display	CICS
TC|TCBS	TCBS Display	CICS
TEMP|TEMPSTRG	TEMPSTRG Display	CICS
TEMPX|TEMPXPND	TEMPXPND Display	CICS
TE|TERMINAL	TERMINAL Display	CICS
TERMX|TERMXPND	TERMXPND Display	CICS
TI|TIOT	TIOT Display	CICS
TR|TRAN	TRAN Display	CICS
TRANX|TRANXPND	TRANXPND Display	CICS
TS|TSUT	TSUT Display	CICS
VS|VSAM	VSAM Display	CICS
VT|VTAM	VTAM Display	CICS
MAINVIEW for DB2 Applications		
BFRPL	Buffer Pool Status	DB2
CICSC	CICS DB2 Connections	DB2

Table 56. Service Select Codes for Refresh Cycle SERVICE Field (Continued)

Service Select Code	Application Description	Product Line (Type)
CICSE	CICS DB2 RCT Entry	DB2
CICSR	CICS DB2 RCT Summary	DB2
DBTS	DB/TS Status	DB2
DB2EX	DB2 Exceptions	DB2
DB2ST	DB2 System Status	DB2
DDFCV	DDF Conversations	DB2
DDFDT	DDF Statistics Detail	DB2
DDFSM	DDF Statistics Summary	DB2
DDFVT	DDF VTAM Status	DB2
DLOGS	DB2 LOG Status	DB2
DMON	Monitor Summary	DB2
DTRAC	Detail Trace Entry	DB2
DUSER	Detail User Status	DB2
DWARN	Warning Summary	DB2
EDMPL	EDM Pool Status	DB2
LOCKD	Lock Contention by DB/TS	DB2
LOCKE	Lock Contention, User Detail	DB2
LOCKU	Lock Contention by User	DB2
LTRAC	DB2 Trace Entries	DB2
PLOT	Monitor History	DB2
RIDPL	RID Pool Status	DB2
STRAC	Summary Trace Entry	DB2
TSTAT	Trace Statistics	DB2
TSUMA	Trace Summary by AUTHID	DB2
TSUMC	Trace Summary by CONNECT	DB2
TSUML	Trace Summary by LOCATION	DB2
TSUMP	Trace Summary by PLAN	DB2
TSUMT	Trace Summary by TIME	DB2
USERS	User Summary	DB2
UTRAC	User Detail Trace	DB2
ZPARM	DB2 System Parameters	DB2

Table 56. Service Select Codes for Refresh Cycle SERVICE Field (Continued)

Service Select Code	Application Description	Product Line (Type)
IMF/MAINVIEW for DBCTL Applications		
Note: Unless indicated otherwise, the following applications are provided by both IMF and MAINVIEW for for DBCTL.		
APPC	APPC Activity Summary (IMF)	IMS
APPCL	APPC LU Status (IMF)	IMS
BALGQ	BALG Queuing (IMF)	IMS
CLASQ	Class Queuing (IMF)	IMS
DAPPC	Inbound|Outbound Allocation (IMF)	IMS
DBST	ISAM/OSAM Pools	IMS
DLIST	DL/I Call Status	IMS
DLTCH	Latch Detail	IMS
DMBUT	DMB Pool Utilization	IMS
DMON	Monitor Summary	IMS
DPOOL	Detail Pool	IMS
DREGN	Region Detail	IMS
DSPST	Dispatcher Statistics	IMS
DTRAC	Display Workload Trace	IMS
DWAIT	Display Workload Wait	IMS
DWARN	Warning Summary	IMS
FPBST	Fast Path Buffer Pool	IMS
IRLM	IRMLM IMS Status	IMS
IRLMG	IRMLM Global Status	IMS
ISTAT	Terminal Input Status (IMF)	IMS
LATCH	Latch Summary	IMS
LCRES	IRLM Lock Contention by Resource	IMS
LCUSR	IRLM Lock Contention by User	IMS
LHRES	IRLM Locks Held by Resources	IMS
LHUSR	IRLM Locks Held by User	IMS
LOGST	Log Statistics	IMS
LUSRD	IRLM Lock User Detail	IMS

Table 56. Service Select Codes for Refresh Cycle SERVICE Field (Continued)

Service Select Code	Application Description	Product Line (Type)
LTRAC	List of Trace Entries	IMS
MFSST	MFS Statistics (IMF)	IMS
MFSUT	MFS Pool Utilization (IMF)	IMS
OSTAT	Terminal Output Status (IMF)	IMS
QUEST	Queue Statistics (IMF)	IMS
PI	Program Isolation	IMS
PLOT	Monitor History	IMS
POOLC	Pool Summary (CBT)	IMS
POOLS	Pool Summary (non-CBT)	IMS
PSBUT	PSB Pool Utilization	IMS
REGNS	IMS Regions	IMS
REGND	Region Detail	IMS
RS	ESA Real Storage	IMS
SCHED	Scheduling Statistics (IMF)	IMS
STAT/ STATR	System Status	IMS
STRAC	Summary Trace Entry	IMS
TRANQ	Transaction Queue Status (IMF)	IMS
USER	User Status Summary (IMF)	IMS
VSST	VSAM GLOBAL or Subpool Statistics	IMS

You can define up to 30 services. Use forward and backward scrolling (PF7/19 and PF8/20) to define and display additional services.

To validate the services you specified, press ENTER. The values are shown in the input fields of the SERVICE REFRESH CYCLE application. You can change any of the input field values.

To start the cycle, enter the GO command (PF6/18), as described in “Starting and Stopping the SERVICE REFRESH CYCLE” on page 479. This definition can be saved for later reuse, as described in “SAVE Command” on page 479.

Saving a Refresh Cycle

If you have a refresh cycle that you use often, you can predefine it in a member of the BBPROF data set, then use the member name to invoke it from the COMMAND line. The naming convention for the members is:

Member Name	Description
MTODMIN or OPERATOR	Examples of a meaningful 1- to 8 character member name
CAOCYCnn	Where nn is any two alphanumeric characters (MAINVIEW AutoOPERATOR for CICS option)
CMRCYCnn	Where nn is any two alphanumeric characters (MAINVIEW for CICS product line)
IAOCYCnn	Where nn is any two alphanumeric characters (MAINVIEW AutoOPERATOR for IMS option)
CYCxx	Where xx is any two alphanumeric characters (MAINVIEW for DB2 or IMF product lines)
MAOCYCnn	Where nn is any two alphanumeric characters (MAINVIEW AutoOPERATOR for OS/390 option)
AOCYCnn	Where nn is any two alphanumeric characters (AutoOPERATOR Base)

Use the following keywords to define a refresh cycle in a BBPROF member.

Keyword	Description
SERV=	The service select code; for example, SERV=LOG. See the SERVICE parameter values in “Creating Cyclic Displays: CYCLE SETUP” on page 469.
OPT=	Specifies the service parameter(s). A valid statement: – Can be written within single quotes Multiple parameters (maximum of 60 characters) can be specified in a statement, separated by blanks, enclosed in single quotes. A comma or end-of-card terminates the operand. – Can be written without single quotes If single quotes are not used, a statement terminates with the last parameter or a comma. Multiple parameters (maximum of 60 characters) can be specified in a statement, separated by blanks.

TARGET= A 1- to 8 character identifier of the target.

If TARGET and TYPE are not specified, the target displayed in the TGT field when the member is selected is used.

If TARGET is not specified but TYPE is, the current target for the specified product line when the member is selected is used.

TYPE= The type of product line to process the requested service. A product line type does not need to be entered when only one product line is installed.

If a product line is not specified, the product line where CYCLE was invoked is used.

When multiple product lines are installed, a product line type must be specified if the service to be requested does not belong to the active product line.

INTVL= Specifies the time in seconds (1 to 99) the service is to be displayed before the next one is shown.

LOG= Specifies either Y (YES) or N (NO) to log the display to the BBI-TS Image log. This parameter is applicable only to the IMF and MAINVIEW for DB2 product lines.

For example, the services defined for a refresh cycle in “Creating Cyclic Displays: CYCLE SETUP” could be defined in a BBPROF member as:

```
*  SAMPLE CYCLE SETUP MEMBER
*
*  IMS DISPLAY MFS UTILIZATION
*
SERV=MFSUT,TYPE=IMS,INTVL=3
*
*  DB2 USER STATUS
*
SERV=DUSER,TYPE=DB2,INTVL=3
*
*  MVS ACTIVE DISPLAY
*
SERV=DA,TYPE=MAO,INTVL=5
*
*  CICS OPEN FILE DISPLAY
*
SERV=FILE,OPT=* OPEN,TYPE=CICS,INTVL=3
*
```

Figure 266. Sample Cycle Setup Member

Include comments by putting an asterisk in column 1.

Select the BBPROF member by using the SELECT|S primary command with the member name on the COMMAND line of the SERVICE REFRESH CYCLE application. The SELECT command can select a 1- to 8 character BBPROF member name across product lines. The SELECT command also allows meaningful names to be defined; it is the recommended method for invoking an BBPROF member. For example:

```
COMMAND ===> SELECT MAOCYC00
```

Note: AOCYC00, CAOCYC00, IAOCYC00, and MAOCYC00 members are distributed in the BBPROF data set with AutoOPERATOR.

SAVE Command

Once a group of services is defined using the CYCLE SETUP application, the definition can be saved in the BBPROF data set by issuing:

```
COMMAND ===> SAVE nn
```

where `nn` is an alphabetic suffix for IMFCYC/CMRCYC or MVSCYC or

```
COMMAND ===> SAVE member-name
```

where `member-name` is any 3- to 8 character alphanumeric member name.

Starting and Stopping the SERVICE REFRESH CYCLE

Enter the GO command (PF6/18) to start the cycle. Press the attention interrupt key (ATTN for SNA terminals and PA1 for non-SNA terminals) to stop the cycle.

When the cycle stops, the last display shown is reissued and returned to the screen in INPUT mode.

You can analyze the status and make additional requests for other displays as usual. When you press PF3, the display returns to the SERVICE REFRESH CYCLE application and shows the service names, parameters, and any short messages. Press PF6 or type GO to restart the cycle.

Each refreshed service display is shown in the sequence and time (DTIME) specified. For the example shown in “Creating Cyclic Displays: CYCLE SETUP” on page 469, the IMF MFSUT application will be displayed first, followed by the MAINVIEW for DB2 DUSER application, the DA application of the AutoOPERATOR MVS option, and the MAINVIEW for CICS FILE application. The cycle repeats again, starting with IMF MFSUT.

Viewing Current Operational Events: Log Display

The *MAINVIEW Common Customization Guide* describes Journal logging. AutoOPERATOR uses the BBI Journal log to record:

- MVS, IMS, CICS, and NetView commands issued from AutoOPERATOR
- Responses to MVS, IMS, CICS, and NetView commands issued from AutoOPERATOR
- All messages handled by a Rule that specifies JRNL=Y
- BBI operation messages
- All BBI-TS user commands

When AutoOPERATOR messages are written to the MVS console, the BBI-SS PAS ID is appended whenever possible. The BBI-SS PAS Address Space ID is not recorded in the Journal log.

Log Display provides a realtime picture of all current operational events. In the BBI Log Display, you can split the screen to view the BBI-SS PAS Journal log on one side and the SYSTEM STATUS application on the other so you can see current exceptions and actual commands and responses at the same time.

Option L, Log Display, displays the Journal log belonging to:

- The BBI-SS PAS monitoring the target identified in the TGT field
- The BBI-SS PAS identified by BBI-SS PASS ID in the TGT field
- The BBI-TS, if LOCAL is specified in the TGT field

The identifier in the TGT field can be changed to point to any valid MVS, IMS, or CICS jobname or stepname, any valid BBI-SS PAS ID, or to LOCAL. Ask your AutoOPERATOR administrator for a list of valid BBI-SS PAS IDs.

In addition to the standard primary commands, the primary commands listed below are valid for Log Display.

Command	Description
L	Locate a specific line in the log. For example: L 5 locates the fifth line of the log, L 10 locates the tenth line of the log, and so on. The specified line is moved to the top of the display.
F	Find a specific string in the log. Strings with blanks must be enclosed in single quotes. The line with the specified string is moved to the top of the display.
T	Locate a specific time in the log. The line with the specified time is moved to the top of the display. Time can be entered as hh:mm:ss, hh:mm, hhmm, hh, or hh.mm.

PROFILE Use the PROFILE command to access the Enhanced Journal Facility, which provides user-customizable views of the Log Display. It allows you to view subsets of the Journal log defined by one or more message origins.

Using the PROFILE command: This command displays an input panel where you can define which messages you want to see. You can specify up to six patterns for message origins to be included and six patterns for message origins to be excluded in the display. Generic qualifiers can be used to define these patterns. For example, you can include all messages from CICS* and exclude all messages from CICSTEST, as shown in Figure 267:

```
BMC Software ------------------- Log Display -------------- General services
COMMAND ===>
                                                              Date --- 01/03/08
 Included Origins            Excluded Origins                 Time --- 10:00:11

  CICS*___                      CICSTEST
  ________                      ________
  ________                      ________
  ________                      ________
  ________                      ________
  ________                      ________
```

Figure 267. Define a Subset of Messages with the PROFILE Command

The PROFILE specifications are saved in userid.BBPROF member LDPARM00. This enables each user to have an individual application profile.

You also can include messages from the current target specified in the TGT field by specifying the variable, &target, in the Included Origins column, as shown in Figure 268:

```
BMC Software ------------------- Log Display -------------- General services
COMMAND ===>
                                                              Date --- 01/0308
 Included Origins            Excluded Origins                 Time --- 10:05:18

  &TARGET_                      ________
  ________                      ________
  ________                      ________
  ________                      ________
  ________                      ________
  ________                      ________
```

Figure 268. Include Messages from the Current Target

Figure 269 shows a sample of the Enhanced Journal Facility output produced from the profile specified in Figure 268. It displays only those messages from target, DB2F.

```
BMC Software ------------------- Log Display -------------- General services
COMMAND ===>                                                TGT ===> DB2F
Line        3,913  Log  #1   Status  INPUT    Time  10:33:25 INTV===> 3
15:57:01  DW0120W  (02) 15:57:00 AVG ELAPSED TIME(TRANS) = 99.882  (>5.000) ***
15:58:00  DW0121I   15:58:00 AVG ELAPSED TIME(TRANS) NO LONGER > 5.000
15:58:00  DS0541I   15:58:00 CSA PAGING(TOTAL) NO LONGER > 1800
15:59:00  DW0120W  (01) 15:59:00 AVG ELAPSED TIME(TRANS) = 99.690  (>5.000) ***
16:00:08  DW0121I   16:00:00 AVG ELAPSED TIME(TRANS) NO LONGER > 5.000
17:01:02  DS0540W  (01) 17:01:00 CSA PAGING(TOTAL) = 2599 IN 01:00 MIN (>1800)
17:02:00  DS0540W  (02) 17:02:00 CSA PAGING(TOTAL) = 2530 IN 01:00 MIN (>1800)
17:03:00  DS0540W  (03) 17:03:00 CSA PAGING(TOTAL) = 2341 IN 01:00 MIN (>1800)
17:04:00  DS0541I   17:04:00 CSA PAGING(TOTAL) NO LONGER > 1800
18:19:00  DS0540W  (01) 18:19:00 CSA PAGING(TOTAL) = 2185 IN 01:00 MIN (>1800)
18:20:00  DS0541I   18:20:00 CSA PAGING(TOTAL) NO LONGER > 1800
19:18:00  DS0540W  (01) 19:18:00 CSA PAGING(TOTAL) = 2251 IN 01:00 MIN (>1800)
19:19:00  DS0541I   19:19:00 CSA PAGING(TOTAL) NO LONGER > 1800
19:31:00  DW0120W  (01) 19:31:00 AVG ELAPSED TIME(TRANS) = 101.138  (>5.000) **
19:32:00  DW0120W  (02) 19:32:00 AVG ELAPSED TIME(TRANS) = 99.313  (>5.000) ***
19:33:00  DW0121I   19:33:00 AVG ELAPSED TIME(TRANS) NO LONGER > 5.000
19:34:00  DW0120W  (01) 19:34:00 AVG ELAPSED TIME(TRANS) = 99.200  (>5.000) ***
19:35:00  DW0121I   19:35:00 AVG ELAPSED TIME(TRANS) NO LONGER > 5.000
19:36:00  DW0120W  (01) 19:36:00 AVG ELAPSED TIME(TRANS) = 99.200  (>5.000) ***
19:37:00  DW0120W  (02) 19:37:00 AVG ELAPSED TIME(TRANS) = 99.170  (>5.000) ***
19:38:00  DW0121I   19:38:00 AVG ELAPSED TIME(TRANS) NO LONGER > 5.000
```

Figure 269. View Messages from the Current Target

To view the origin of the messages in the Log Display, scroll to the left.

Note: You must specify JOURNAL=ENHANCED in BBPARM member BBISSP00 for the target BBI-SS PAS for the PROFILE specifications to take effect.

The PROFILE specifications are not active for a target of LOCAL.

Displaying System Messages: MESSAGES

Option M, MESSAGES, displays a scrollable list of AutoOPERATOR short messages, error messages, and abend codes. Any message or code can be selected with a line command (S) to produce a complete description that includes:

- The reason the message was issued
- What system action will be taken
- What user action should be taken
- The name of the module of origin

Use the LOCATE|L command to locate a specific message. For example, `L IM` locates the first message beginning with the characters IM and displays the message as the first line on the screen.

Messages appearing in the short messages field that have more detailed information associated with them are also contained in this list. The short messages precede the messages starting with message numbers and are stored in message-text collating sequence.

The messages are obtained from the BBMLIB data set. This data set may be browsed for messages if a BBI-TS is not available. If user messages are added to the BBMLIB data set, they must be prefixed with an at (@) sign.

Using PF Keys: KEYS

Option K, KEYS, displays the characteristics of the AutoOPERATOR program function PF key assignments.

See “Using Program Function (PF) Key Definitions in AutoOPERATOR” on page 26 to change the PF key assignments.

Requesting Time-Initiated EXECs: TIMEXEC Application

The TIMEXEC application manages time-initiated EXEC requests by using the CALLX service to invoke your specified EXEC based on time. It extracts and formats system, job, and time data from a target and BBI. It then passes this information to the EXEC as variables in 10 standard parameters that can be used within the EXEC logic.

Refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs* for more information regarding EXECs.

The input panel for the TIMEXEC application is shown in Figure 270.

```
BMC Software --------- TIME-INITIATED EXEC REQUESTS ------------ REQUEST ISSUED
COMMAND ===>                                                     TGT ===> CICSA
                                                                 DATE - 01/15/97
Enter single CALLX request:                                      TIME - 14:16:44
 EXEC ==>
    INTVL ==>          START ==>          STOP ==>          QIS ==>
  STOPCNT ==>

Enter BLK request:
  BLK ==>
                         --- CURRENT CALLX REQUESTS ---
 LC CMDS : P(PURGE)
 LC  EXEC     STATUS INTVL    START     STOP    QIS USER
     DRIVER   INIT   24:00:00 07:00:00          NO  CWB1
     SMFDUMP  INIT   24:00:00 18:00:00          NO  CWB1
     DISA     INIT   00:30:00 08:00:00          NO  CWB1
     *************** END OF REQUESTS *****************
```

Figure 270. TIME-INITIATED EXEC REQUESTS Panel

The TIME-INITIATED EXEC REQUESTS panel fields are defined as follows.

Field	Description
LC	The line command field. Enter the following 1-character command on the line of the CALLX request you want to purge: P
EXEC	The name of the EXEC for which a CALLX request was made.
STATUS	The current status of the CALLX request for an EXEC, which can be:
ACTV	The request is active.
INIT	The request is waiting to be scheduled. A start time was specified and has not been reached yet.
HELD	HOLD=YES was specified by the SET command. The request is being held until another SET command is issued for the CALLX request for this EXEC.
COMP	The request stopped. The stop time or count specified by the TIMEXEC application STOP parameter was reached.
INV	Request terminated because of an invalid parameter.

LOCK	The request terminated because: – A service routine ABEND locked the CALLX service. – A locked request cannot be reactivated. It must be purged.
RST	The request will restart at the next interval time.
INTVL	The period of time between EXEC invocations.
START	The start time of the request.
STOP	The stop time of the request. Blank means a stop time has not been specified.
USER	The identifier of the AutoOPERATOR user who made the request (USERID).

Describing the TIMEXEC Application

A CALLX request can be a single EXEC scheduled and invoked from the TIMEXEC panel. Or a CALLX request can be a block of multiple EXECs and timer-driven service requests. Blocks of service requests can be invoked:

- From the TIMEXEC panel
- Automatically at BBI-SS PAS startup (BLK keyword parameter in BBPARM member BBIISP00)

Instructions for scheduling and invoking CALLX requests are given in “Setting Up a CALLX Request” on page 486.

Use the TIMEXEC application to:

- Schedule and invoke one EXEC over a specified time at defined intervals
- Activate a group of predefined CALLX requests, each of which can schedule an EXEC at different times and intervals
- View a scrollable list of active CALLX EXEC scheduling requests
- Purge an active CALLX request shown in the scrollable list

Setting Up a CALLX Request

Use TIMEXEC to make single or multiple timer-driven CALLX requests.

A single request specifies a unique EXEC name. If timer-driven, the CALLX request invokes the EXEC at a specific time and specifies a timed interval when the EXEC is to re-execute. A single request is set up by:

- Specifying a name for the EXEC parameter

and optionally:

- Setting times with the INTVL, START, or STOP parameters
- Specifying QIS ==> YES if the EXEC should quiesce while the target is inactive *(IMS targets only)*

A request for multiple CALLX services is set up by:

- Entering a BBPARM member name for the BLK parameter

The details and valid values for these TIMEXEC parameters are described in the next two sections. If you make an error when you enter these values, a brief message is displayed in the upper right-hand corner of the panel. If an ERROR IN BLOCK REQUEST, ERROR IN REQUEST, or ERROR IN PURGE message is displayed, an explanatory message is written to the Journal log.

Single CALLX Request

The TIMEXEC parameter values for invoking and scheduling a single EXEC are described in the next table.

Table 57. TIMEXEC Parameters

Parameter	Value
EXEC ==>	Enter a 1- to 8 character member name of an EXEC in the BBIPROC data set. You cannot use the EXEC parameter and the BLK parameter, described in “Block of CALLX Requests” on page 488, at the same time. If you do, AutoOPERATOR generates an INVALID COMBINATION error message.
INTVL ==>	This optional parameter specifies the time interval between each EXEC invocation. Enter hours, minutes, and seconds in hh:mm:ss format. The minimum value is one second, 00:00:01, and the maximum value is 24 hours, 24:00:00. The INTVL value specifies how often the EXEC is to be scheduled. For example, you create an EXEC named BACKUP that issues an CICS command to start a backup procedure. Using the TIMEXEC application parameters, EXEC, START, and INTVL, you identify the BACKUP EXEC and specify a START of 18:00:00, with an INTVL of 24:00:00. This means a backup procedure, issued as an CICS command by the BACKUP EXEC, will be initiated every day at 6:00 pm. Refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs* for more information regarding EXECs. If you do not specify an interval time, the default is specified by the SET keyword, INTERVAL, in the BBPARM member BBIISP00. INTERVAL in the distributed BBIISP00 member is set to one minute.
START ==>	This optional parameter specifies the time the EXEC is to start. Enter hours, minutes, and seconds in hh:mm:ss format. Values for the time parameters are specified as hh:mm:ss, where hh is the number of hours, mm is the number of minutes, and ss is the number of seconds. All time fields for this format must contain numeric data. Leading zeroes must be used to fill a field as needed, such as 00:05:25 or 05:15:00. If you specify a time 10 minutes or more *before* the current time, AutoOPERATOR adds 24 hours to the START time. For example, if at 15:50:00 you request a START time of 15:35:00, AutoOPERATOR starts the EXEC the next day at 15:35:00. If no START time is specified, the default is to start at the next full minute from the current time.
STOP ==>	This optional parameter specifies when the EXEC is to stop. Enter hours, minutes, and seconds in hh:mm:ss format.

Table 57. TIMEXEC Parameters (Continued)

Parameter	Value
STOPCNT ==>	You can enter a stop count as any number from 1 to 99999999. A stop count specifies the number of times the EXEC is to execute before it stops. For example, if you enter the number 5, the EXEC is executed five times and then stops. If you want an EXEC to be scheduled only once, enter a 1. The STOPCNT is entered in the STOP field. If you do not specify a stop time, the EXEC remains active until purged (see "Purging an Active CALLX Request" on page 489) or the BBI-SS terminates.
QIS ==> *(IMS targets only)*	This optional parameter specifies whether or not this CALLX request should be quiesced (YES) when IMS is inactive. Enter YES or NO. YES tells the TIMER facility not to schedule the EXEC when IMS is inactive. If NO is entered, the EXEC is scheduled as specified by the START, STOP, or INTVL parameters. The default is NO. The AutoOPERATOR CALLX service passes the target status to the time-initiated EXEC *(IMS targets only)*. The EXEC can test this status before issuing any commands.

Block of CALLX Requests

Timer-driven requests can be grouped and named as a BBPARM member. This includes CALLX service requests. Timer-driven requests can then be issued by:

- Entering the name of the BBPARM member in the BLK parameter of the TIMEXEC application:

Parameter	Value
BLK	Enter a 1- to 8 character name of a BBPARM member containing a group of CALLX requests. AutoOPERATOR generates an INVALID COMBINATION error message if you request a block of CALLX services and a single CALLX service at the same time.

- Specifying the BBPARM member name in the BLK parameter of the SET command.
- Specifying the BBPARM member name in the BLK parameter of BBPARM member BBIISP00. The CALLX timer-driven requests are automatically started when the BBI-SS PAS starts.

  The example steps are:

  1. Create a member in BBPARM and name it STARTX.

     STARTX requests three timer-driven EXECs called DRIVER, SMFDUMP, and DISA. These EXECs do the following:

EXEC	Description
DRIVER	Starts VTAM system at 7 am and, when initialized, starts TSO.
SMFDUMP	Switches and Dumps SMF data every day at 6:00 pm except weekends.
DISA	Issues a #D A,L command every 30 minutes.

2. Include the timer-driven CALLX requests and any other timer-driven service requests you want in the BBPARM STARTX member:

   ```
   REQ=CALLX DRIVER
   REQ=CALLX SMFDUMP START=18:00:00
   REQ=CALLX DISA I=00:30:00 START=14:17:00
   ```

3. Specify either:

   – STARTX for the BLK parameter value (`BLK ==> STARTX`) in the TIMEXEC application.

   – BLK=STARTX in the BBIISP00 member of BBPARM, which automatically initiates the CALLX requests in the BBPARM STARTX member whenever the BBI-SS PAS starts.

Purging an Active CALLX Request

When you select the TIMEXEC application, a scrollable list of active EXEC schedule requests is displayed as CURRENT CALLX REQUESTS. On the left is the line command (LC) column. The only line command is P, to purge an active CALLX request. Move the cursor to the LC column beside the name of an EXEC and enter a P to purge the CALLX request for that EXEC.

Note: To purge a time-initiated request that was created by an EXEC, the TGT field on the TIME-INITIATED EXEC REQUESTS panel must specify the same name as that used on the TARGET=xxxx parameters on the IMFEXEC IMFC SET REQ=CALLX statement.

For example, the TGT field on the panel must display SYSB if the following were coded in an EXEC:

```
IMFEXEC IMFC SET REQ=CALLX                                  +
    @DAILY, START=10:45:00, STOP=10:55:00,                  +
    I=00:10:00, TARGET=SYSB, USRID=JDB1
```

Part 5. Using AutoOPERATOR Sample Solutions

This part contains chapters that describe how you can use the AutoOPERATOR Sample Solutions to accomplish many of the initial automation tasks quickly and efficiently after you install the AutoOPERATOR product.

Chapter 16. Introduction to Using Sample Solutions 493
Distributed Sample Solutions 493
Naming Conventions for EXECs 494
Standards for Sample Solutions 497

Chapter 17. Sample VTAM/NCP Solution 499
Sample VTAM Commands Solution 499
Setting Variables for the Sample VTAM Commands Solution 499
Invoking the Sample VTAM Commands Solution 499
Processing Flow for the Sample VTAM Commands Solution 500
EXECs for the Sample VTAM Commands Solution 500
Rules for the Sample VTAM Commands Solution 500

Chapter 18. Sample Monitoring Solution 501
Sample Operations Monitors Solution 501
Setting Variables for the Sample Operations Monitors Solution 501
Invoking the Sample Operations Monitors Solution 501
Processing Flow for the Sample Operations Monitors Solution 501
EXECs for the Sample Operations Monitors Solution 503
Rules for the Sample Operations Monitors Solution 503

Chapter 19. Sample CICSPlex Solution 505
Invoking the Sample CICSPlex Solution 505
EXECs 506
Rules 506

Chapter 20. Sample Info/System Management Solution 507
Invoking the Sample Info/System Management Solution 507
EXECs for the Sample Info/System Management Solution 507
Rules for the Sample Info/System Management Solution 508

Chapter 21. Activating the MVALARM Solution 509
Enabling the MVALARM Solution 509
Stopping the MVALARM Solution 510

Chapter 16. Introduction to Using Sample Solutions

AutoOPERATOR 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.

Caution
With the Sample Solutions, BMC Software delivers sample source code that is to be us 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.

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:

```
gffnnnl
```

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

Code	Function
AM	ACCESS METHODS[1]
ER	ERROR RECOVERY[1]
IN	INFO SYS/MANAGEMENT
J2	JES2
J3	JES3[1]
MN	MONITORS
PE	PERFORMANCE[1]
ST	STORAGE SUBSYSTEMS[1]
UT	UTILITIES
VT	VTAM/NCP

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 documentation box.

[1] The EXECs for these functional areas are available as part of the MV Solutions for AutoOPERATOR product.

Figure 271 shows an example of the 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                          */
```

Figure 271. Example of a 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" on page 457 for detailed explanation). The rest of the information in the documentation box is for self-documentation and easy reference.

Following is a brief description of the title fields shown in Figure 271 on page 495:

GROUP
Is the group to which this EXEC belongs (for example, MVS, CICS, DB2, or IMS)

FUNC
Is the functional category within the group

CODE
Is the 2-character function code

DISP
Shows whether or not information about this EXEC should be displayed under the AutoOPERATOR EXEC Management Application (refer to "Managing EXECs Using the EXEC Management Application" on page 457)

AUTHOR
Is the author of the EXEC

DESC
Is a 25 character (or less) description of what this EXEC does

NAME
Is the name of this EXEC

DESCRIPTION
Is a detailed description of what this EXEC does

SEE ALSO
Is a cross-reference listing

INVOKED BY
Is a list of other EXECs that call this EXEC

INPUT PARAMETERS
If any input parameters are required, this indicates the parameter name, description, and default value

OUTPUT
Is the information or names of variables which is passed back

OUTBOARD CALLS
Indicates if any pager calls are placed

EXTERNAL ROUTINES CALLED
Are other routines or services called along with any special considerations for those calls

ALERTS ISSUED
If this is a long EXEC and ALERTS are issued, this gives a brief text; otherwise, a YES or NO

VARIABLES USED
If any variables are used, this indicates the variable name, description, and default value

TEST SCRIPT INFO
Indicates the necessary environment and steps needed to test this EXEC if changes are required

RETURN CODES
Shows what return codes (if any) are passed back

CHANGE LOG
Is a 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.

Chapter 17. Sample 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

Caution
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 VTAM Commands Solution

The following sections describe the Sample VTAM Commands Solution.

Setting Variables for the Sample VTAM Commands Solution

Enabling and disabling 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:

P1 Resource name

P2 Scope (optional)

Inactivating a Resource

To inactivate a resource, enter:

```
%NETINACT P1 P2
```

on the COMMAND line where:

P1	Resource name
P2	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.

NETACT	Activates resources
NETBOUNC	Bounces resources
NETINACT	Inactivates resources

Rules for the Sample VTAM Commands Solution

There are no Rules for the Sample VTAM Commands Solution.

Chapter 18. 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

Caution
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 five volumes with the highest I/O rates are reported using the ALERT facility. 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.

$CSAABV	Determines top five users of extended CSA
$CSABLO	Determines top five users of CSA
$JOBCPU	Determines top batch job user of CPU
$STCCPU	Determines top started task user of CPU
$TSOCPU	Determines top TSO user of CPU
$TOTCPU	Determines top user of CPU
$TOTSIO	Determines top five busiest DASD devices
$TOTSIO2	Determines which address spaces are accessing a particular device
$WATCH	Provides job statistics
MMN002C	Issues message asking if the address space should be cancelled
MMN003C	Cancels address space

Rules for the Sample Operations Monitors Solution

There are no Rules for the Sample Operations Monitors Solution.

Chapter 19. 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 will process first for address spaces other then 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 BB* Rules needed to include the appropriate logic to determine the jobname 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, jobname, 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, jobname, message text, and an ALERT key of `SEVERE ERROR`.
BBMZA800W	CAS(ssid) connect interface disabled.	A red critical alert is created with time, date, jobname, and message text.
BBC*W	All warning messages related to the Communications Manager.	Something might be wrong, but CICSPlex processing continues. A yellow major ALERT is created with time, date, jobname, message text, and an ALERT key of `COMMUNICATIONS ERROR: INVESTIGATE`.
BB*W	All warning messages.	Something may be wrong, but CICSPlex processing continues. A yellow major ALERT is created with time, date, jobname, message text, and an ALERT key of `WARNING ERROR: INVESTIGATE`.

Message ID Types	Description	Notes.
BBMZA995A	CAS(SSID) reply “U” to wait or “C” to continue termination.	WTOR for termination. A pink major ALERT is created with time, date, jobname, 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, jobname, 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, jobname, 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, jobname, 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, jobname, 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, jobname, 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, jobname, message text, and an ALERT key of `WTOR OUTSTANDING`.

EXECs

There are no EXECs for this solution.

Rules

There is one message Rule for each message or message category listed in “Invoking the Sample CICSPlex Solution” on page 505. The Rules are located in BBPARM member RULBCPSM.

Chapter 20. Sample Info/System Management Solution

The Sample Info/System Management Solution helps data center personnel add problem records to the INFO database.

Caution
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 OSPINFO**1**. OSPINFO**0** was created using the OSPI application.

Refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs* for more information about the OSPI application.

The member OSPINFO**1** was created from the base OSPINFO**0**. 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.

IEF450I Call OSPINFO1 based on job name

OSPINFO1 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.

IEF450I Text-ID = IEF450I, distributed DISABLED

Chapter 21. Activating the MVALARM Solution

The MVALARM solution allows you to convert alarms sent from MAINVIEW Alarm Manager to AutoOPERATOR ALERTs. The AutoOPERATOR ALERTs can be viewed with the ALERT Management Facility. This allows you to be 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.

The MVALARM Solution is comprised of two Rules in the AAORUL00 Rule Set: ALRMSTRT and ALRMSTOP.

Enabling the MVALARM Solution

The MVALARM Solution is comprised of two Rules in the AAORUL00 Rule Set: ALRMSTRT and ALRMSTOP.

To enable the Rules:

1. Invoke the Rule Processor application by typing RULES from any BBI terminal session screen.
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.
3. Select the AAORUL00 Rule Set.
4. Enable the ALRMSTRT and ALRMSTOP Rules. Do this 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 will be upgraded in priority every 10 minutes.

After the ALERT reaches the CRITICAL priority, it will be deleted. If the situation that caused the alarm is resolved before this, 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
```

Part 6. Appendixes

This part contains the following appendixes:

Appendix A. Diagnosing CSM Errors 513
Creating a Cross-Reference Report 513
Gathering Environmental Information 514
Generating CSM Exec Diagnostics for Support 514
Tracing Starts and Stops of CSM EXECs 516
Tracing All Activity in a CSM EXEC 516
Stopping a Trace 517
Creating CSM Start and Stop Rules 518
Creating CSM Start and Stop Rules for the APPC Object 518
Creating CSM Start and Stop Rules for the TCPIP Object 518
Common Questions and Answers 520
Starting/Stopping Errors 520
Restart-Only Objects 521
Defining Special Objects 521
CSMACT Usage 524
BBI Terminal Sessions (BBI-TS) 525
TSO/E or Environmental Problems 525
CSM Installation 526
Effective Implementation 526
Other Tips and Questions 527

Appendix B. MVS Sample Solutions 531

Appendix C. DBCTL Commands 533
Issuing Commands in a DBCTL Environment 533
Issuing DBCTL Commands from a Terminal Session 533
Issuing DBCTL Commands from an EXEC 534

Appendix D. Describing Rule Selection Criteria and Action Specifications 535
Selecting Criteria for ALRM-Initiated Rules 536
Specifying Actions for ALRM-Initiated Rules 538
Selecting Criteria for ALRT-Initiated Rules 541
Specifying Actions for ALRT-Initiated Rules 543
Selecting Criteria for CICS-Initiated Rules 546
Specifying Actions for CICS-Initiated Rules 548
Selecting Criteria for CMD-Initiated Rules 551
Specifying Actions for CMD-Initiated Rules 553
Selecting Criteria for DB2-Initiated Rules 557
Specifying Actions for DB2-Initiated Rules 559
Selecting Criteria for EXT-Initiated Rules 562
Specifying Actions for EXT-Initiated Rules 564
Selecting Criteria for IMS-Initiated Rules 567
Specifying Actions for IMS-Initiated Rules 569
Selecting Criteria for JES3-Initiated Rules 572
Specifying Actions for JES3-Initiated Rules 575
Selecting Criteria for JRNL-Initiated Rules 578
Specifying Actions for JRNL-Initiated Rules 580
Selecting Criteria for MSG-Initiated Rules 583
Specifying Actions for MSG-Initiated Rules 586

Selecting Criteria for Time-Initiated Rules 590
Examples 591
Specifying Actions for Time-Initiated Rules. 593
Selecting Criteria for VAR-Initiated Rules 596
Specifying Actions for VAR-Initiated Rules 597

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.

- 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’s 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:

1. Display the CSM panel (Figure 272).

```
BMC Software ------------------ Global Overview ------------------ 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)elete
                 (CAN)cel  (M)ove    (R)epeat  (B)rowse      s(Y)nc

LC  Object   Group    System   Actual       Desired      Ind   Type
    ________ ________ ________ ____________ ____________ _____ ________
___ MVS      KMZ1     SJSD     UP           UP                 NORM

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

Figure 272. `Global Overview` (CSM Administration Panel): Generating a Cross-Reference Report

2. Enter the `Xref` primary command.

Figure 273 is displayed.

```
BMC Software ------------ CSM Diagnostic Cross-Reference --------- AutoOPERATOR
COMMAND ===> _                                                     TGT --- SYSB

CSM will generate a cross-reference report of the object definitions for
the subsystem contained in: 'SYSB.AO31.SYSB.UBBPARM'

The report will be written to:

Data Set Name ===> 'SYSB.CSMXREF.KMZ1'__________________________

Browse report after generation (Y/N) ? Y

SYSOUT class for printed report?  _


Enter END to generate the report and return.
Enter CANCEL to return without generating report.
```

Figure 273. CSM Diagnostic Cross-Reference Panel

3. If you would rather use a different data set name, overtype the `Data Set Name` field. Remember to enclose the data set name in quotes.

   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.

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.

5. When you are finished browsing the report data set, press PF3/END.

6. Write the report data set to tape and send it and/or the printed report 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 invoke 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. **Note:** Specifying `L(GLOBAL)` may produce a 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 BBILOG) 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 hundreds 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 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 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.DEBUGThis 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:

1. On the `Add NEW Object` panel for APPC, specify NO in the `Generate the Rules` field.
2. Build a Start Rule:
   a. On the Selection Criteria panel, specify ATB007I on the `Text Id` field
   b. On the Variable Dependencies panel, specify `!CSM.ACTIVEFLAG EQ YES`
   c. On the Automation Specification panel, specify YES for `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.

3. Build a Stop Rule:
   a. On the Selection Criteria panel, specify ATB002I on the `Text Id` field
   b. On the Variable Dependencies panel, specify `!CSM.ACTIVEFLAG EQ YES`
   c. On the Automation Specification panel, specify YES for `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:

1. On the `Add NEW Object` panel for APPC, specify NO in the `Generate the Rules` field.
2. Build a Start Rule:
   a. On the Selection Criteria panel, specify EZB6473I on the `Text Id` field
   b. On the Variable Dependencies panel, specify !CSM.ACTIVEFLAG EQ YES
   c. On the Automation Specification panel, specify YES for `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.

3. Build a Stop Rule:
   a. On the Selection Criteria panel, specify IEF404I on the `Text Id` field
   b. On the Variable Dependencies panel, specify !CSM.ACTIVEFLAG EQ YES
   c. On the Automation Specification panel, specify YES for `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 the 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 time out. 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, invoke 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)
```

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/or VTAM do not supply a jobname for message IST020I. Since jobname 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. This makes sure 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 provides database services for either CICS or IMS, DB2MSTR would 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.

It's advisable also to define a Rule in a Rule Set other than AAORULBC or AAORULCM that selects on the IEF450I message being issued by DB2MSTR, DB2DBM1, or DB2IRLM. With this Rule in place, all three components of DB2 are recycled properly (and CSM's variables are updated properly) in the event that any DB2 component fails.

Question: Why doesn't CSM recognize the start and stop of the APPC object?

Answer: This can happen because the Start and Stop Rules have an IMFOJOB variable value equal to APPC as selection criteria, and since 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 223.

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 ***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. This is because of the difference in EXEC scheduling between the Rules Processor and other EXEC submission vehicles. See the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* and 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 or 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.CMDLIB to SYS1.LPALIB, this error occurs. To correct the problem, either move the ALLOCATE module back to SYS1.CMDLIB 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 CLIST EXECs* or 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. When it completes, 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. This 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's 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. DB@ 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. MVS Sample Solutions

Table 58 is the index to the AutoOPERATOR EXECs for MVS in the $MAOINDX data set. The EXECs are used to provide solutions to problems relating to the MVS Group.

Table 58. MVS Sample Solutions Index

Member	Description
Sample VTAM/NCP Solution: Sample VTAM Commands Solution	
NETACT (OPER)	Command interface to activate Network resources
NETINACT (OPER)	Command interface to inactivate Network resources
NETBOUNC (OPER)	Command interface to inactivate/activate Network resources
Sample Monitoring Solution: Sample Operations Monitors Solution	
$CSAABV (OPER)	Determine CSA usage above line
$CSABLO (OPER)	Determine CSA usage below line
$JOBCPU (OPER)	Check for looping JOB
$STCCPU (OPER)	Check for looping STC
$TOTCPU (OPER)	Check for looping task prompt operator for cancel
$TSOCPU (OPER)	Check for looping TSU prompt operator for cancel
$TOTSIO (OPER)	Monitor I/O by volume issue alert for high usage
$TOTSIO2	Determine which address spaces are doing I/O to a particular volume and display alerts in order of highest user of device
$WATCH (OPER)	Give back statistics on a job/task

Appendix C. DBCTL Commands

This appendix describes how to issue commands to a DBCTL target address space.

Issuing Commands in a DBCTL Environment

DBCTL commands can be issued from a terminal session or from an EXEC.

Issuing DBCTL Commands from a Terminal Session

When issued from a TS, the TARGET field must point to a DBCTL address space which is defined in BBPARM member BBIJNT00 with TYPE=IMS. Both the command and the command response are recorded in the local BBI Journal. Only the command is logged to the BBI Journal.

To have the command response also logged to the BBI Journal, you must have AutoOPERATOR for MVS installed. Then, create Rules to capture the messages from the MVS console. Commands can be entered in one of these two ways.

If you have authority to issue MVS commands:

- Enter a pound sign (#), which is the BBI MVS command prefix, followed by the MVS command recognition character, and then followed by the DBCTL command.

  To have the command response also logged to the local BBI Journal, you must have AutoOPERATOR for MVS installed.

  For example, issuing the

  ```
  #!STO DATABASE ABC
  ```

  command stops the database named `ABC`.

  Note: For this example, the ! (exclamation point) is the MVS command recognition character.

If you have authority to issue IMS commands:

- Enter the command syntax exactly as you would enter an IMS command: (`/ command`)

  For example, issuing the:

  ```
  /STO DATABASE ABC
  ```

  command stops the database named `ABC`.

Issuing DBCTL Commands from an EXEC

You can issue DBCTL commands from an EXEC by using the IMS or the MVS command format of the IMFEXEC CMD.

When you use the IMS command format of the IMFEXEC CMD command, add the additional keyword `DBCTL(target)` to specify the jobname of a DBCTL address space. For example:

```
IMFEXEC CMD '/STO DATABASE ABC'
  COUNT(1O) TYPE(IMS) DBCTL(IMSDBC)
```

Refer to the section “CMD (Issue IMS Command with Response)” in the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs* for information about how to use this command.

To use the MVS command format of the IMFEXEC CMD command, you do not need to add the keyword `DBCTL(target)`. For example:

```
IMFEXEC CMD '#!STO DATABASE ABC'
  COUNT(1O)
```

Refer to the section “CMD (Issue MVS/JES Command with Response)” in the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs* for information on how to use this command.

Appendix D. Describing Rule Selection Criteria and Action Specifications

The following sections describe the customized Selection Criteria and Action Specification panels for all event types. The different kinds of Rules are:

- ALRM-initiated Rules (for ALRM event types), page 536
- ALERT-initiated Rules (for ALRT event types), page 541
- CICS-initiated Rules (for CICS event types), page 546
- CMD-initiated Rules (for CMD event types), page 551
- DB2-initiated Rules (for DB2 event types), page 556
- EXT-initiated Rules (for EXT event types), page 562
- IMS-initiated Rules (for IMS event types), page 567
- JES3-initiated Rules (for JES event types), page 572
- BBI Journal-initiated Rules (for JRNL event types), page 578
- MSG-initiated Rules (for MSG event types), page 583
- TIME-initiated Rules (for firing a Rule based on time), page 590
- VAR-initiated Rules (for firing a Rule based on variable value) page 596

Selecting Criteria for ALRM-Initiated Rules

When you are creating an ALRM-Initiated Rule, the following Selection Criteria panel is displayed when you press ENTER on the Rule Detail Processor Control panel.

```
BMC Software --------------- Selection Criteria - ALRM ------------ AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === DUMMY001

Text Description:
Alarm ID    ===> $HASP*                              Alarm Identifier
Alarm Text (Enter Below):


Context   ===>                                       Alarm Context
Scope     ===>                                       Alarm Scope
Userid    ===>                                       Monitor Identifier
Queue     ===>                                       Alarm Queue
Priority  ===>                                       Alarm Priority
Type      ===>                                       Alarm Start/Stop Event









Press END to return to Detail Control, ENTER to continue
```

Figure 274. Selection Criteria Panel for ALRM-Initiated Rules

Use the `Selection Criteria - ALRM` panel to specify the attributes of a MV AlarmManager Alarm that you want to create a Rule for. Table 59 describes the Selection Criteria fields for an ALRM-initiated Rule.

Table 59. Selection Criteria for ALRM-Initiated Rules

Field	Description
Alarm ID	Enter the Alarm ID associated with the Alarm. The length of the Alarm ID is determined by the first blank character in the text. To choose more information to select for, use the Alarm Text field to enter Alarm text. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Alarm Text	Specify Alarm text to be matched in addition to the Alarm ID. In this field, you must enter the Alarm ID and additional information from the message text for the Rule to match before it fires. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Context	Specify the context of the Alarm. SHARED variables, EVENT variables, and pattern matching can be used in this field.

Table 59. Selection Criteria for ALRM-Initiated Rules (Continued)

Field	Description
Scope	Specify the Scope of the Alarm SHARED variables, EVENT variables, and pattern matching can be used in this field.
Userid	Specify the User ID of the Alarm. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Queue	Specify the Queue name of the Alarm. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Priority	Specify the priority of the Alarm. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Type	Specify the Type of the Alarm. SHARED variables, EVENT variables, and pattern matching can be used in this field.

"Specifying Actions for ALRM-Initiated Rules" describes the actions you can specify for an ALRM-initiated Rule.

Specifying Actions for ALRM-Initiated Rules

Figure 275 shows an example of the Action Specification panel for ALRM-initiated Rules.

```
BMC Software --------------- Action Specification - ALRM ---------- AutoOPERATOR
COMMAND ===>

       Rule-set === AAORUL00            Rule-id  === ICH70001

Automation Actions:
EXEC Name/Parms  ===>
Send (TSO IDs)   ===>
Cmd (Type   MVS) ===>

Set Variable     ===>                                    ===>
Notify           ===>                                   Outboard Pager ID
Info             ===>

DOM Id           ===>                                   Delete Operator Message
Issue WTO Msg    ===>





Press END to return to Detail Control, ENTER to continue
```

Figure 275. Action Specification Panel for ALRM-Initiated Rules

Use the `Action Specification - ALRM` panel to specify what actions an ALRM-initiated Rule will take.

Table 60 describes the Action Specification fields for an ALRM-initiated Rule.

Table 60. Action Specifications for ALRM-Initiated Rules

Field	Description
EXEC name|Parms	Allows you to specify the name of an EXEC the Rule can invoke and any parameters that you want to send to the EXEC. This field is 56 characters long. Separate the EXEC name and any parameters by a blank. Note that the first parameter specified in this field will be the first parameter passed to the EXEC. This also means the message ID and any message text will not be passed to the EXEC. To have the message ID and message text passed to the EXECs, use the variable IMFTEXT in this field. EVENT and SHARED variables can be used in this field.
Send (TSO)	Allows you to send a copy of the Alarm using the MVS SEND command to 1-5 TSO user IDs. Separate each TSO user ID by a blank. This field is 40 characters long. EVENT and SHARED variables are valid in this field.

Table 60. Action Specifications for ALRM-Initiated Rules (Continued)

Field	Description
Cmd (Type xxx)	Enter one or more commands to be issued, where xxx represents a user-specified command type. Possible values are: **MVS** Allows you to specify MVS console commands in this field; the address space that originated the event will issue the commands **IMS** Allows you to specify IMS commands **BBI** Allows you to specify BBI commands **SS** Allows you to specify MVS commands that originate from the BBI-SS PAS **CIC** Allows you to specify CICS commands. The CICS command must contain a target; for example: `TARGET:COMMANDTEXT` **MQ** Allows you to specify MQSeries commands In addition: • This field is 126 characters long • EVENT and SHARED variables can be used in this field • You can use multiple commands by separating them with double colon marks: :: • MQSeries commands must begin with the name of the queue manager that will process the command specified. For example: `queuemgrname:command text` For documentation about how to use the Cmd(MQ) field, refer to the AutoOPERATOR *for MQSeries Installation and Reference Guide*.
Set variable	Allows you to set, increment, or decrement the value of a SHARED variable. A variable's numerical value may be incremented or decremented by specifying `+n` or `-n`, where `n` is a number. The first field is 33 characters long and SHARED variables are valid. Use the first field to specify the variable to be modified. The second field is 17 characters long and SHARED variables and constants are valid. Use the second field to assign a new value (with a variable or an unsigned constant) or modify a numeric value (with a signed constant). **Note:** The variables set by using this field are set asynchronously to the firing of the Rule. In other words, there will be a slight delay before the value of the variables are set.

Table 60. Action Specifications for ALRM-Initiated Rules (Continued)

Field	Description
Notify	Enter the telephone number of the AutoOPERATOR Elan operator ID to be notified using an outboard pager. This field is 33 characters long. EVENT and SHARED variables are valid. Specify the telephone number exactly as it must be dialed. The Elan operator ID must exist in the Elan Operator Profile. Use this field in conjunction with the `Info` field.
Info	Specify a 1-80 alphanumeric character string of information to be displayed in the area of an outboard pager managed by AutoOPERATOR Elan. Use this field in conjunction with the `Notify` field.
DOM ID	Specify the name of a SHARED variable that contains the delete-operator-message ID of a previously issued WTO.
Issue WTO Msg	Specify a 1-125 character message to be sent to an MVS console. The message is issued without routing or descriptor codes unless specified with the R (route) or D (descriptor code) keywords (or both), and followed by a colon (:). For example, to issue a message with Descriptor code 2 and Route code 11: `Issue WTO Msg ===> D(2) R(11): EMPMSG88 &WORD1 ISSUED` The message may be selected by another Rule. The default is blank, which means no message is issued. EVENT and SHARED variables are valid.

Selecting Criteria for ALRT-Initiated Rules

When you are creating an ALRT-initiated Rule, the following Selection Criteria panel is displayed when you press ENTER on the Rule Detail Processor Control panel.

```
BMC Software --------------- Selection Criteria - ALRT ------------ AutoOPERATOR
COMMAND ===>

       Rule-set === AAORUL00             Rule-id  === ICH70001

Text Description:
Text ID     ===> ICH70001                       First Word of ALert
Text String (Enter Below):


Key      ===>
Queue    ===>                                   Alert Queue





Press END to return to Detail Control, ENTER to continue
```

Figure 276. Selection Criteria Panel for ALRT-Initiated Rules

Use the `Selection Criteria - ALRT` panel to specify the attributes of an AutoOPERATOR ALERT that you want to create a Rule for. The two ALERTs attributes you can specify are the Key and the Queue of an ALERT.

Table 61 describes the Selection Criteria fields for an ALRT-initiated Rule.

Table 61. Selection Criteria for ALRT-Initiated Rules

Field	Description
Text ID	Enter the 1-16 byte Text ID associated with the message. The length of the Text ID is determined by the first blank character in the text. The Text ID for the event ALRT is the first word, delimited by a blank, of the ALERT or the first 16 characters if the first word is longer than 16 characters. To select on more information, use the `Text String` field to enter message text. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Text String	Specify message text to be matched in addition to the Text ID. In this field, you must enter the Text ID and additional information from the message text for the Rule to match before it fires. This field supports up to 75 bytes. SHARED variables, EVENT variables, and pattern matching can be used in this field.

Table 61. Selection Criteria for ALRT-Initiated Rules (Continued)

Field	Description
Key	Specify the key that uniquely identifies an ALERT within a queue. Each ALERT has its own unique key and use this field to create Rules for ALERTs written with specific keys. SHARED variables, EVENT variables, and pattern matching can be used in this field. For more information about ALERT keys, refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.
Queue	Specify the queue name to which the ALERT is targeted. Every ALERT belongs to a queue and use this field to create Rules for ALERTs destined for specific queues. SHARED variables, EVENT variables, and pattern matching can be used in this field. For more information about ALERT queues, refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.

“Specifying Actions for ALRT-Initiated Rules” describes the actions you can specify for an ALRT-initiated Rule.

Specifying Actions for ALRT-Initiated Rules

Figure 277 shows an example of the Action Specification panel for ALRT-initiated Rules.

```
BMC Software --------------- Action Specification - ALRT ---------- AutoOPERATOR
COMMAND ===>

       Rule-set === AAORUL00             Rule-id  === ICH70001

Automation Actions:
Display at dest. ===> YES       Journal ===>
EXEC Name/Parms  ===>
Send (TSO)       ===>
Cmd (Type    MVS)===>

Set Variable     ===>                                    ===>
Reword Alert     ===>

Notify           ===>                                  Outboard Pager ID
Info             ===>

DOM Id           ===>                                  Delete Operator Message
Issue WTO Msg    ===>



Press END to return to Detail Control, ENTER to continue
```

Figure 277. Action Specification Panel for ALRT-Initiated Rules

Use the `Action Specification - ALRT` panel to specify what actions an ALRT-initiated Rule will take. You can specify actions such as invoking an EXEC, setting a variable, rewording the ALERT, or notifying an outboard operator.

Table 62 describes the Action Specification fields for an ALRT-initiated Rule.

Table 62. Action Specifications for ALRT-Initiated Rules

Field	Description
Display at dest.	Specifies whether the ALERT will be displayed at the destination it was originally sent to. Valid values are YES and NO. YES is the default. If you specify NO, the ALERT is suppressed. Variables cannot be used in this field.
Journal	Specifies whether the ALERT will be recorded in the BBI Journal. Valid values are YES and NO. The default is NO.

Table 62. Action Specifications for ALRT-Initiated Rules (Continued)

Field	Description
EXEC name|Parms	Allows you to specify the name of an EXEC the Rule can invoke and any parameters that you want to send to the EXEC. This field is 56 characters long. Separate the EXEC name and any parameters by a blank. Note that the first parameter specified in this field will be the first parameter passed to the EXEC. This also means the message ID and any message text will not be passed to the EXEC. To have the message ID and message text passed to the EXECs, use the variable IMFTEXT in this field. EVENT and SHARED variables can be used in this field.
Send (TSO)	Allows you to send a copy of the ALERT using the MVS SEND command to 1-5 TSO user IDs. Separate each TSO user ID by a blank. This field is 40 characters long. EVENT and SHARED variables are valid in this field.
Cmd (Type xxx)	Enter one or more commands to be issued, where xxx represents a user-specified command type. Possible values are: **MVS** Allows you to specify MVS console commands in this field; the address space that originated the event will issue the commands **IMS** Allows you to specify IMS commands **BBI** Allows you to specify BBI commands **SS** Allows you to specify MVS commands that originate from the BBI-SS PAS **CIC** Allows you to specify CICS commands. The CICS command must contain a target; for example: `TARGET:COMMANDTEXT` **MQ** Allows you to specify MQSeries commands In addition: • This field is 126 characters long • EVENT and SHARED variables can be used in this field • You can use multiple commands by separating them with double colon marks: :: • MQSeries commands must begin with the name of the queue manager that will process the command specified. For example: `queuemgrname:command text` For documentation about how to use the Cmd(MQ) field, refer to the AutoOPERATOR *for MQSeries Installation and Reference Guide*.

Table 62. Action Specifications for ALRT-Initiated Rules (Continued)

Field	Description
Set variable	Allows you to set, increment, or decrement the value of a SHARED variable. A variable's numerical value may be incremented or decremented by specifying `+n` or `-n`, where `n` is a number. The first field is 33 characters long and SHARED variables are valid. Use the first field to specify the variable to be modified. The second field is 17 characters long and SHARED variables and constants are valid. Use the second field to assign a new value (with a variable or an unsigned constant) or modify a numeric value (with a signed constant). **Note:** The variables set by using this field are set asynchronously to the firing of the Rule. In other words, there will be a slight delay before the value of the variables are set.
Reword Alert	Use this field to change the ALERT text to be sent to its destination. Note that the reworded text is used for all subsequent Rule matching comparisons if you use an Automation strategy of ALL. The length of the reworded text cannot exceed 255 characters. EVENT and SHARED variables are valid in this field.
Notify	Enter the telephone number of the AutoOPERATOR Elan operator ID to be notified using an outboard pager. This field is 33 characters long. EVENT and SHARED variables are valid. Specify the telephone number exactly as it must be dialed. The Elan operator ID must exist in the Elan Operator Profile. Use this field in conjunction with the `Info` field.
Info	Specify a 1-80 alphanumeric character string of information to be displayed in the area of an outboard pager managed by AutoOPERATOR Elan. Use this field in conjunction with the `Notify` field.
DOM ID	Specify the name of a SHARED variable that contains the delete-operator-message ID of a previously issued WTO.
Issue WTO Msg	Specify a 1-125 character message to be sent to an MVS console. The message is issued without routing or descriptor codes unless specified with the R (route) or D (descriptor code) keywords (or both), and followed by a colon (:). For example, to issue a message with Descriptor code 2 and Route code 11: `Issue WTO Msg ===> D(2) R(11): EMPMSG88 &WORD1 ISSUED` The message may be selected by another Rule. The default is blank, which means no message is issued. EVENT and SHARED variables are valid.

Selecting Criteria for CICS-Initiated Rules

When you are creating a CICS-initiated Rule, the following Selection Criteria panel is displayed when you press ENTER on the Rule Detail Processor Control panel.

```
BMC Software --------------- Selection Criteria - CICS ------------ AutoOPERATOR
COMMAND ===>

       Rule-set === AAORULZZ             Rule-id  === ICH70001

Text Description:
Text ID     ===> ICH70001                          First word of TDQ msg/cmd
Text String (Enter Below):



CICS TDQ    ===>                                   CICS Transient Data Queue









Press END to return to Detail Control, ENTER to continue
```

Figure 278. Selection Criteria Panel for CICS-Initiated Rules

Use the `Selection Criteria - CICS` panel to specify the attributes of a CICS Transient Data Queue message that you want to create a Rule for. The one CICS TDQ attribute you can specify is the TD Q itself.

Table 63 describes the Selection Criteria fields for an CICS-initiated Rule.

Table 63. Selection Criteria for CICS-Initiated Rules

Field	Description
Text ID	Enter the 1-16 byte Text ID associated with the message. The length of the Text ID is determined by the first blank character in the text. The Text ID for the message is the first word, delimited by a blank, or the first 16 characters if the first word is longer than 16 characters. To select on more information, use the `Text String` field to enter message text. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Text String	Specify message text to be matched in addition to the Text ID. In this field, you must enter the Text ID and additional information from the message text for the Rule to match before it fires. This field supports up to 75 bytes. SHARED variables, EVENT variables, and pattern matching can be used in this field.
CICS TDQ	Is the 1- to 4-character name of the CICS Transient Data Queue where the event originated. SHARED variables, EVENT variables, and pattern matching can be used in this field.

“Specifying Actions for CICS-Initiated Rules” describes the actions you can specify for an CICS-initiated Rule.

Specifying Actions for CICS-Initiated Rules

Figure 279 shows an example of the Action Specification panel for CICS-initiated Rules.

```
BMC Software --------------- Action Specification - CICS ---------- AutoOPERATOR
COMMAND ===>

       Rule-set === AAORULZZ             Rule-id  === ICH70001

Automation Actions:
Display at dest. ===> YES       Journal ===>
EXEC Name/Parms  ===>
Send (TSO)       ===>
Cmd (Type   MVS) ===>

Set Variable    ===>                                      ===>
Reword Msg       ===>


Notify           ===>                                      Outboard Pager ID
Info             ===>

DOM Id           ===>                                      Delete Operator Message
Issue WTO Msg    ===>


Press END to return to Detail Control, ENTER to continue
```

Figure 279. Action Specification Panel for CICS-Initiated Rules

Table 64 describes the Action Specification fields for an CICS-initiated Rule.

Table 64. Action Specifications for CICS-Initiated Rules

Field	Description
Display at dest.	Specifies whether the CICS TDQ message will be displayed at the destination it was originally sent to. Valid values are YES and NO. YES is the default. If you specify NO, the CICS TDQ message is suppressed. Variables cannot be used in this field. **For CICS events**, When any line of a multiline message is suppressed in the TDQ (XTDOUT) exit, then all subsequent lines of the multiline message are discarded (not presented to the XTDOUT exit). This means that if a Rule fires that handles a CICS message and has NO specified in this field, then all remaining lines of the multiline message will be discarded (not presented to the Rules Processor). This restriction is due to the design of the CICS exit. Given this CICS behavior, you cannot suppress a part of any multiline CICS message and be able to receive the rest of the multiline message in a Rule.
Journal	Specifies whether the CICS TDQ message will be recorded in the BBI Journal. Valid values are YES and NO. The default is NO.

Table 64. Action Specifications for CICS-Initiated Rules (Continued)

Field	Description
EXEC name|Parms	Allows you to specify the name of an EXEC the Rule can invoke and any parameters that you want to send to the EXEC. This field is 56 characters long. Separate the EXEC name and any parameters by a blank. Note that the first parameter specified in this field will be the first parameter passed to the EXEC. This also means the message ID and any message text will not be passed to the EXEC. To have the message ID and message text passed to the EXECs, use the variable IMFTEXT in this field. EVENT and SHARED variables can be used in this field.
Send (TSO)	Allows you to send a copy of the message using the MVS SEND command to 1-5 TSO user IDs. Separate each TSO user ID by a blank. This field is 40 characters long. EVENT and SHARED variables are valid in this field.
Cmd (Type xxx)	Enter one or more commands to be issued, where xxx represents a user-specified command type. Possible values are: **MVS** Allows you to specify MVS console commands in this field; the address space that originated the event will issue the commands **IMS** Allows you to specify IMS commands **BBI** Allows you to specify BBI commands **SS** Allows you to specify MVS commands that originate from the BBI-SS PAS **CIC** Allows you to specify CICS commands. The CICS command must contain a target; for example: `TARGET:COMMANDTEXT` **MQ** Allows you to specify MQSeries commands In addition: • This field is 126 characters long • EVENT and SHARED variables can be used in this field • You can use multiple commands by separating them with double colon marks: :: • MQSeries commands must begin with the name of the queue manager that will process the command specified. For example: `queuemgrname:command text` For documentation about how to use the Cmd(MQ) field, refer to the AutoOPERATOR *for MQSeries Installation and Reference Guide*.

Table 64. Action Specifications for CICS-Initiated Rules (Continued)

Field	Description
Set variable	Allows you to set, increment, or decrement the value of a SHARED variable. A variable's numerical value may be incremented or decremented by specifying `+n` or `-n`, where `n` is a number. The first field is 33 characters long and SHARED variables are valid. Use the first field to specify the variable to be modified. The second field is 17 characters long and SHARED variables and constants are valid. Use the second field to assign a new value (with a variable or an unsigned constant) or modify a numeric value (with a signed constant). **Note:** The variables set by using this field are set asynchronously to the firing of the Rule. In other words, there will be a slight delay before the value of the variables are set.
Reword Msg	Specifies that the CICS Transient Data queue messages should be reworded as specified in this field, prior to being sent to the Transient Data Queue.
Notify	Enter the telephone number of the AutoOPERATOR Elan operator ID to be notified using an outboard pager. This field is 33 characters long. EVENT and SHARED variables are valid. Specify the telephone number exactly as it must be dialed. The Elan operator ID must exist in the Elan Operator Profile. Use this field in conjunction with the `Info` field.
Info	Specify a 1-80 alphanumeric character string of information to be displayed in the area of an outboard pager managed by AutoOPERATOR Elan. Use this field in conjunction with the `Notify` field.
DOM ID	Specify the name of a SHARED variable that contains the delete-operator-message ID of a previously issued WTO.
Issue WTO Msg	Specify a 1-125 character message to be sent to an MVS console. The message is issued without routing or descriptor codes unless specified with the R (route) or D (descriptor code) keywords (or both), and followed by a colon (:). For example, to issue a message with Descriptor code 2 and Route code 11: `Issue WTO Msg ===> D(2) R(11): EMPMSG88 &WORD1 ISSUED` The message may be selected by another Rule. The default is blank, which means no message is issued. EVENT and SHARED variables are valid.

Selecting Criteria for CMD-Initiated Rules

When you are creating a CMD-initiated Rule, the following Selection Criteria panel is displayed when you press ENTER on the Rule Detail Processor Control panel.

```
BMC Software --------------- Selection Criteria - CMD  ------------ AutoOPERATOR
COMMAND ===>

       Rule-set === AAORUL00             Rule-id  === ICH70001

Command Description:
Command ID  ===> ICH70001                          First word of command
Command String (Enter Below):


Issuer Identification:
Job name    ===>                                   Name of issuer
Type        ===>                                   (JOB, STC, or TSO)
Jobclass    ===>                                   Job class of issuer
Acct Info   ===>                                   Job accounting information
RACF User   ===>                                   RACF Userid
RACF Group  ===>                                   RACF Group name
Console ID  ===>                                   Originating Console
Console Name ==>                                   Originating Console Name



Press END to return to Detail Control, ENTER to continue
```

Figure 280. Selection Criteria Panel for CMD-Initiated Rules

Use the `Selection Criteria - CMD` panel to specify the attributes of a command that you want to create a Rule for.

Table 65 describes the Selection Criteria fields for an CMD-initiated Rule.

Table 65. Selection Criteria for CMD-Initiated Rules

Field	Description
Command ID	Enter the 1-16 byte command ID associated with the command. The length of the command ID is determined by the first blank character in the text. The command ID is the first word, delimited by a blank, or the first 16 characters if the first word is longer than 16 characters. To select on more information, use the `Command String` field to enter more command text. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Command string	Specify command text to be matched in addition to the Command ID. In this field, you must enter the command ID and additional information from the command text for the Rule to match before it fires. This field supports up to 75 bytes. SHARED variables, EVENT variables, and pattern matching can be used in this field.

Table 65. Selection Criteria for CMD-Initiated Rules (Continued)

Field	Description
Job name	Specify a 1 to 8 byte name of the address space that issued the event. This field is usually used with the `Type` field which specifies the type of address space issuing the event. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Type	Specify a 1 to 3 byte name for the type of address space that issued the event. Valid values are JOB, STC, or TSO. This field provides additional information about the address space that issued the event. SHARED variables and EVENT variables can be used in this field.
Jobclass	Specify a one byte identifier for the job class of a batch job issuer. Valid values are 0-9 and A-Z. SHARED variables and EVENT variables can be used in this field.
Acct Info	Enter up to 27 bytes of information about the accounting information of the issuer of the event. SHARED variables, EVENT variables, and pattern matching can be used in this field.
RACF User	A 1 to 8 byte field where you can enter the RACF user ID for the address space that issued the message. The RACF user ID is taken from the USERID keyword of the job card. SHARED variables, EVENT variables, and pattern matching can be used in this field.
RACF Group	A 1 to 8 byte field where you can enter the RACF group ID for the address space that issued the message. The RACF Group ID is taken from the GROUP keyword of the job card. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Console ID	Specify the console ID of a specific MVS console associated with the message. Valid console ID numbers are 1-99. SHARED variables and EVENT variables can be used in this field.
Console Name	Specify the console name of a specific MVS console associated with the message. SHARED variables, EVENT variables, and pattern matching can be used in this field.

“Specifying Actions for CMD-Initiated Rules” describes the actions you can specify for an CMD-initiated Rule.

Specifying Actions for CMD-Initiated Rules

Figure 281 shows an example of the Action Specification panel for CMD-initiated Rules.

```
BMC Software  ------------- Action Specification - CMD  ---------- AutoOPERATOR
COMMAND ===>                                                 TGT  --- JPP1

       Rule-set === RULJON00             Rule-id  === SADKF

Automation Actions:
Reject Command   ===>           Journal ===>
EXEC Name/Parms  ===>
Send (TSO IDs)   ===>
Reword Cmd       ===>

Cmd (Type  MVS ) ===>

Set Variable     ===>                                      ===>

Notify           ===>                                     Outboard Pager ID
Info             ===>

DOM Id           ===>                                     Delete Operator Message
Issue WTO Msg    ===>



Press END to return to Detail Control, ENTER to continue
```

Figure 281. Action Specification Panel for CMD-Initiated Rules

Use the `Action Specification - CMD` panel to specify what actions a CMD-initiated Rule will take.

Table 66 describes the Action Specification fields for an CMD-Initiated Rule.

Table 66. Action Specifications for CMD-Initiated Rules

Field	Description
Reject Command	Specifies whether command will be issued or rejected.
Journal	Specifies whether the command will be recorded in the BBI Journal. Valid values are YES and NO. The default is NO.
EXEC name|Parms	Allows you to specify the name of an EXEC the Rule can invoke and any parameters that you want to send to the EXEC. This field is 56 characters long. Separate the EXEC name and any parameters by a blank. Note that the first parameter specified in this field will be the first parameter passed to the EXEC. This also means the message ID and any message text will not be passed to the EXEC. To have the message ID and message text passed to the EXECs, use the variable IMFTEXT in this field. EVENT and SHARED variables can be used in this field.

Table 66. Action Specifications for CMD-Initiated Rules (Continued)

Field	Description
Send (TSO)	Allows you to send a copy of the message using the MVS SEND command to 1-5 TSO user IDs. Separate each TSO user ID by a blank. This field is 40 characters long. EVENT and SHARED variables are valid in this field.
Cmd (Type xxx)	Enter one or more commands to be issued, where xxx represents a user-specified command type. Possible values are: **MVS** Allows you to specify MVS console commands in this field; the address space that originated the event will issue the commands **IMS** Allows you to specify IMS commands **BBI** Allows you to specify BBI commands **SS** Allows you to specify MVS commands that originate from the BBI-SS PAS **CIC** Allows you to specify CICS commands. The CICS command must contain a target; for example: `TARGET:COMMANDTEXT` **MQ** Allows you to specify MQSeries commands In addition: • This field is 126 characters long • EVENT and SHARED variables can be used in this field • You can use multiple commands by separating them with double colon marks: :: • MQSeries commands must begin with the name of the queue manager that will process the command specified. For example: `queuemgrname:command text` For documentation about how to use the Cmd(MQ) field, refer to the AutoOPERATOR *for MQSeries Installation and Reference Guide*. This field is 126 characters long. EVENT and SHARED variables can be used in this field. You can use multiple commands by separating them with double colon marks: ::

Table 66. Action Specifications for CMD-Initiated Rules (Continued)

Field	Description
Set variable	Enables you to set, increment, or decrement the value of a SHARED variable. A variable's numerical value may be incremented or decremented by specifying `+n` or `-n`, where `n` is a number. The first field is 33 characters long and SHARED variables are valid. Use the first field to specify the variable to be modified. The second field is 17 characters long and SHARED variables and constants are valid. Use the second field to assign a new value (with a variable or an unsigned constant) or modify a numeric value (with a signed constant). **Note:** The variables set by using this field are set asynchronously to the firing of the Rule. In other words, there will be a slight delay before the value of the variables are set.
Notify	Enter the telephone number of the AutoOPERATOR Elan operator ID to be notified using an outboard pager. This field is 33 characters long. EVENT and SHARED variables are valid. Specify the telephone number exactly as it must be dialed. The Elan operator ID must exist in the Elan Operator Profile. Use this field in conjunction with the `Info` field.
Info	Specify a 1-80 alphanumeric character string of information to be displayed in the area of an outboard pager managed by AutoOPERATOR Elan. Use this field in conjunction with the `Notify` field.
DOM ID	Specify the name of a SHARED variable that contains the delete-operator-message ID of a previously issued WTO.
Issue WTO Msg	Specify a 1-125 character message to be sent to an MVS console. The message is issued without routing or descriptor codes unless specified with the R (route) or D (descriptor code) keywords (or both), and followed by a colon (:). For example, to issue a message with Descriptor code 2 and Route code 11: `Issue WTO Msg ===> D(2) R(11): EMPMSG88 &WORD1 ISSUED` The message may be selected by another Rule. The default is blank, which means no message is issued. EVENT and SHARED variables are valid.
Reword CMD	Enter a new command that replaces the original command. All attributes associated with the command such as console ID are retained. The maximum length of a command replacement is 126 characters. This parameter should not be coded for Rules that are used in a release prior to AutoOPERATOR 6.2. Doing so causes this field to be displayed as a message on the console and in syslog.

Selecting Criteria for DB2-Initiated Rules

When you are creating a DB2-initiated Rule, the following Selection Criteria panel is displayed when you press ENTER on the Rule Detail Processor Control panel.

```
BMC Software --------------- Selection Criteria - DB2  ------------ AutoOPERATOR
COMMAND ===>

       Rule-set === AAORULZZ             Rule-id  === ICH70001

Text Description:
Text ID     ===> ICH70001                          First word of WTO/WTOR
Text String (Enter Below):


Issuer Identification:
Job name    ===>                                   Jobname of DB2 Subsystem









Press END to return to Detail Control, ENTER to continue
```

Figure 282. Selection Criteria Panel for DB2-Initiated Rules

Use the `Selection Criteria - DB2` panel to specify the attributes of a DB2 generated message that you want to create a Rule for.

Table 67 describes the Selection Criteria fields for an DB2-initiated Rule.

Table 67. Selection Criteria for DB2-Initiated Rules

Field	**Description**
Text ID	Enter the 1-16 byte Text ID associated with the message. The length of the Text ID is determined by the first blank character in the text. The Text ID for the message is the first word, delimited by a blank, or the first 16 characters if the first word is longer than 16 characters. To select on more information, use the `Text String` field to enter message text. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Text String	Specify message text to be matched in addition to the Text ID. In this field, you must enter the Text ID and additional information from the message text for the Rule to match before it fires. This field supports up to 75 bytes. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Job Name	Jobname of the DB2 system.

"Specifying Actions for DB2-Initiated Rules" describes the actions you can specify for an DB2-initiated Rule.

Specifying Actions for DB2-Initiated Rules

Figure 283 shows an example of the Action Specification panel for DB2-initiated Rules.

Use this panel to specify the actions for a DB2-initiated Rule.

```
BMC Software --------------- Action Specification - DB2  ---------- AutoOPERATOR
COMMAND ===>

       Rule-set === AAORULZZ             Rule-id  === ICH70001

Automation Actions:
Display at dest. ===> YES       Journal ===>
EXEC Name/Parms  ===>
Send (TSO)       ===>
Cmd (Type   MVS) ===>

Set Variable     ===>                                      ===>


Notify           ===>                                    Outboard Pager ID
Info             ===>

DOM Id           ===>                                    Delete Operator Message
Issue WTO Msg    ===>



Press END to return to Detail Control, ENTER to continue
```

Figure 283. Action Specification Panel for DB2-Initiated Rules

Use the `Action Specification - DB2` panel to specify what actions a DB2-initiated Rule will take.

Table 68 describes the Action Specification fields for an DB2-initiated Rule.

Table 68. Action Specifications for DB2-Initiated Rules

Field	Description
Display at dest.	Specifies whether the DB2 message will be displayed at the destination it was originally sent to. Valid values are YES and NO. YES is the default. If you specify NO, the DB2 message is suppressed. Variables cannot be used in this field.
Journal	Specifies whether the DB2 message will be recorded in the BBI Journal. Valid values are YES and NO. The default is NO.

Table 68. Action Specifications for DB2-Initiated Rules (Continued)

Field	Description
EXEC name|Parms	Allows you to specify the name of an EXEC the Rule can invoke and any parameters that you want to send to the EXEC. This field is 56 characters long. Separate the EXEC name and any parameters by a blank. Note that the first parameter specified in this field will be the first parameter passed to the EXEC. This also means the message ID and any message text will not be passed to the EXEC. To have the message ID and message text passed to the EXECs, use the variable IMFTEXT in this field. EVENT and SHARED variables can be used in this field.
Send (TSO)	Allows you to send a copy of the message using the MVS SEND command to 1-5 TSO user IDs. Separate each TSO user ID by a blank. This field is 40 characters long. EVENT and SHARED variables are valid in this field.
Cmd (Type xxx)	Enter one or more commands to be issued, where xxx represents a user-specified command type. Possible values are: **MVS** Allows you to specify MVS console commands in this field; the address space that originated the event will issue the commands **IMS** Allows you to specify IMS commands **BBI** Allows you to specify BBI commands **SS** Allows you to specify MVS commands that originate from the BBI-SS PAS **CIC** Allows you to specify CICS commands. The CICS command must contain a target; for example: `TARGET:COMMANDTEXT` **MQ** Allows you to specify MQSeries commands In addition: • This field is 126 characters long • EVENT and SHARED variables can be used in this field • You can use multiple commands by separating them with double colon marks: :: • MQSeries commands must begin with the name of the queue manager that will process the command specified. For example: `queuemgrname:command text` For documentation about how to use the Cmd(MQ) field, refer to the AutoOPERATOR *for MQSeries Installation and Reference Guide*.

Table 68. Action Specifications for DB2-Initiated Rules (Continued)

Field	Description
Set variable	Allows you to set, increment, or decrement the value of a SHARED variable. A variable's numerical value may be incremented or decremented by specifying `+n` or `-n`, where `n` is a number. The first field is 33 characters long and SHARED variables are valid. Use the first field to specify the variable to be modified. The second field is 17 characters long and SHARED variables and constants are valid. Use the second field to assign a new value (with a variable or an unsigned constant) or modify a numeric value (with a signed constant). **Note:** The variables set by using this field are set asynchronously to the firing of the Rule. In other words, there will be a slight delay before the value of the variables are set.
Notify	Enter the telephone number of the AutoOPERATOR Elan operator ID to be notified using an outboard pager. This field is 33 characters long. EVENT and SHARED variables are valid. Specify the telephone number exactly as it must be dialed. The Elan operator ID must exist in the Elan Operator Profile. Use this field in conjunction with the `Info` field.
Info	Specify a 1-80 alphanumeric character string of information to be displayed in the area of an outboard pager managed by AutoOPERATOR Elan. Use this field in conjunction with the `Notify` field.
DOM ID	Specify the name of a SHARED variable that contains the delete-operator-message ID of a previously issued WTO.
Issue WTO Msg	Specify a 1-125 character message to be sent to an MVS console. The message is issued without routing or descriptor codes unless specified with the R (route) or D (descriptor code) keywords (or both), and followed by a colon (:). For example, to issue a message with Descriptor code 2 and Route code 11: `Issue WTO Msg ===> D(2) R(11): EMPMSG88 &WORD1 ISSUED` The message may be selected by another Rule. The default is blank, which means no message is issued. EVENT and SHARED variables are valid.

Selecting Criteria for EXT-Initiated Rules

When you are creating an EXT-initiated Rule, the following Selection Criteria panel is displayed when you press ENTER on the Rule Detail Processor Control panel.

For more information about external events, refer to “Describing Special Event Types: DB2, EXT, TIME, and VAR” on page 53.

```
BMC Software --------------- Selection Criteria - EXT  ------------ AutoOPERATOR
COMMAND ===>

       Rule-set === AAORUL00             Rule-id  === ICH70001

Text Description:
Text ID     ===> ICH70001                          First word of Text
Text String (Enter Below):


Issuer Identification:
Job name    ===>                                   Name of issuer
Type        ===>                                   (JOB, STC, or TSO)
Jobclass    ===>                                   Job class of issuer
Acct Info   ===>                                   Job accounting information
RACF User   ===>                                   RACF Userid
RACF Group  ===>                                   RACF Group name





Press END to return to Detail Control, ENTER to continue
```

Figure 284. Selection Criteria Panel for EXT-Initiated Rules

Use the `Selection Criteria - EXT` panel to specify the attributes of an externally generated message (from another product) that you want to create a Rule for.

Table 69 on page 562 describes the Selection Criteria fields for an EXT-initiated Rule.

Table 69. Selection Criteria for EXT-Initiated Rules

Field	Description
Text ID	Enter the 1-16 byte Text ID associated with the message. The length of the Text ID is determined by the first blank character in the text. The Text ID for the message is the first word, delimited by a blank, or the first 16 characters if the first word is longer than 16 characters. To select on more information, use the `Text String` field to enter message text. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Text String	Specify message text to be matched in addition to the Text ID. In this field, you must enter the Text ID and additional information from the message text for the Rule to match before it fires. This field supports up to 75 bytes. SHARED variables, EVENT variables, and pattern matching can be used in this field.

Table 69. Selection Criteria for EXT-Initiated Rules (Continued)

Field	Description
Job name	Specify a 1 to 8 byte name of the address space that issued the event. This field is usually used with the `Type` field which specifies the type of address space issuing the event. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Type	Specify a 1 to 3 byte name for the type of address space that issued the event. Valid values are JOB, STC, or TSO. This field provides additional information about the address space that issued the event. SHARED variables and EVENT variables can be used in this field.
Jobclass	Specify a one byte identifier for the job class of a batch job issuer. Valid values are 0-9 and A-Z. SHARED variables and EVENT variables can be used in this field.
Acct Info	Enter up to 27 bytes of information about the accounting information of the issuer of the event. SHARED variables, EVENT variables, and pattern matching can be used in this field.
RACF User	A 1 to 8 byte field where you can enter the RACF user ID for the address space that issued the message. The RACF user ID is taken from the USERID keyword of the job card. SHARED variables, EVENT variables, and pattern matching can be used in this field.
RACF Group	A 1 to 8 byte field where you can enter the RACF group ID for the address space that issued the message. The RACF Group ID is taken from the GROUP keyword of the job card. SHARED variables, EVENT variables, and pattern matching can be used in this field.

"Specifying Actions for EXT-Initiated Rules" on page 564 describes the actions you can specify for an EXT-initiated Rule.

Specifying Actions for EXT-Initiated Rules

Figure 285 on page 564 shows an example of the Action Specification panel for EXT-initiated Rules.

```
BMC Software --------------- Action Specification - EXT  ---------- AutoOPERATOR
COMMAND ===>

       Rule-set === AAORUL00             Rule-id  === ICH70001

Automation Actions:
Journal          ===>
EXEC Name/Parms  ===>
Send (TSO)       ===>
Cmd (Type   MVS) ===>

Set Variable     ===>                                      ===>


Notify           ===>                                    Outboard Pager ID
Info             ===>

DOM Id           ===>                                    Delete Operator Message
Issue WTO Msg    ===>



Press END to return to Detail Control, ENTER to continue
```

Figure 285. Action Specification Panel for EXT-Initiated Rules

Use the `Action Specification - EXT` panel to specify what actions an EXT-initiated Rule will take.

Table 70 on page 564 describes the Action Specification fields for an EXT-initiated Rule.

Table 70. Action Specifications for EXT-Initiated Rules

Field	Description
Journal	Specifies whether the externally generated message will be recorded in the BBI Journal. Valid values are YES and NO. The default is NO.
EXEC name|Parms	Allows you to specify the name of an EXEC the Rule can invoke and any parameters that you want to send to the EXEC. This field is 56 characters long. Separate the EXEC name and any parameters by a blank. Note that the first parameter specified in this field will be the first parameter passed to the EXEC. This also means the message ID and any message text will not be passed to the EXEC. To have the message ID and message text passed to the EXECs, use the variable IMFTEXT in this field. EVENT and SHARED variables can be used in this field.

Table 70. Action Specifications for EXT-Initiated Rules (Continued)

Field	Description
Send (TSO)	Allows you to send a copy of the message using the MVS SEND command to 1-5 TSO user IDs. Separate each TSO user ID by a blank. This field is 40 characters long. EVENT and SHARED variables are valid in this field.
Cmd (Type xxx)	Enter one or more commands to be issued, where xxx represents a user-specified command type. Possible values are: **MVS** Allows you to specify MVS console commands in this field; the address space that originated the event will issue the commands **IMS** Allows you to specify IMS commands **BBI** Allows you to specify BBI commands **SS** Allows you to specify MVS commands that originate from the BBI-SS PAS **CIC** Allows you to specify CICS commands. The CICS command must contain a target; for example: `TARGET:COMMANDTEXT` **MQ** Allows you to specify MQSeries commands In addition: • This field is 126 characters long • EVENT and SHARED variables can be used in this field • You can use multiple commands by separating them with double colon marks: :: • MQSeries commands must begin with the name of the queue manager that will process the command specified. For example: `queuemgrname:command text` For documentation about how to use the Cmd(MQ) field, refer to the AutoOPERATOR *for MQSeries Installation and Reference Guide*.
Set variable	Allows you to set, increment, or decrement the value of a SHARED variable. A variable's numerical value may be incremented or decremented by specifying `+n` or `-n`, where `n` is a number. The first field is 33 characters long and SHARED variables are valid. Use the first field to specify the variable to be modified. The second field is 17 characters long and SHARED variables and constants are valid. Use the second field to assign a new value (with a variable or an unsigned constant) or modify a numeric value (with a signed constant). **Note:** The variables set by using this field are set asynchronously to the firing of the Rule. In other words, there will be a slight delay before the value of the variables are set.

Table 70. Action Specifications for EXT-Initiated Rules (Continued)

Field	**Description**
Notify	Enter the telephone number of the AutoOPERATOR Elan operator ID to be notified using an outboard pager. This field is 33 characters long. EVENT and SHARED variables are valid. Specify the telephone number exactly as it must be dialed. The Elan operator ID must exist in the Elan Operator Profile. Use this field in conjunction with the `Info` field.
Info	Specify a 1-80 alphanumeric character string of information to be displayed in the area of an outboard pager managed by AutoOPERATOR Elan. Use this field in conjunction with the `Notify` field.
DOM ID	Specify the name of a SHARED variable that contains the delete-operator-message ID of a previously issued WTO.
Issue WTO Msg	Specify a 1-125 character message to be sent to an MVS console. The message is issued without routing or descriptor codes unless specified with the R (route) or D (descriptor code) keywords (or both), and followed by a colon (:). For example, to issue a message with Descriptor code 2 and Route code 11: `Issue WTO Msg ===> D(2) R(11): EMPMSG88 &WORD1 ISSUED` The message may be selected by another Rule. The default is blank, which means no message is issued. EVENT and SHARED variables are valid.

Selecting Criteria for IMS-Initiated Rules

When you are creating an IMS-initiated Rule, the following Selection Criteria panel is displayed when you press ENTER on the Rule Detail Processor Control panel.

```
BMC Software --------------- Selection Criteria - IMS  ------------ AutoOPERATOR
COMMAND ===>

       Rule-set === AAORUL00             Rule-id  === ICH70001

Text Description:
Text ID     ===> ICH70001                          First word of message
Text String (Enter Below):


Issuer Identification:
Job name    ===>                                   Jobname of IMS
Type        ===>                                   (JOB, STC, or TSO)
Jobclass    ===>                                   Job class of issuer
Acct Info   ===>                                   Job accounting information
RACF User   ===>                                   RACF Userid
RACF Group  ===>                                   RACF Group name





Press END to return to Detail Control, ENTER to continue
```

Figure 286. Selection Criteria Panel for IMS-Initiated Rules

Use the `Selection Criteria - IMS` panel to specify the attributes of an IMS Master Terminal Operator-generated message that you want to create a Rule for.

Table 71 describes the Selection Criteria fields for an IMS-initiated Rule.

Table 71. Selection Criteria for IMS-Initiated Rules

Field	Description
Text ID	Enter the 1-16 byte Text ID associated with the message. The length of the Text ID is determined by the first blank character in the text. The Text ID for the message is the first word, delimited by a blank, or the first 16 characters if the first word is longer than 16 characters. To select on more information, use the `Text String` field to enter message text. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Text String	Specify message text to be matched in addition to the Text ID. In this field, you must enter the Text ID and additional information from the message text for the Rule to match before it fires. This field supports up to 75 bytes. SHARED variables, EVENT variables, and pattern matching can be used in this field.

Table 71. Selection Criteria for IMS-Initiated Rules (Continued)

Field	Description
Job name	Specify a 1 to 8 byte name of the address space that issued the event. This field is usually used with the `Type` field which specifies the type of address space issuing the event. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Type	Specify a 1 to 3 byte name for the type of address space that issued the event. Valid values are JOB, STC, or TSO. This field provides additional information about the address space that issued the event. SHARED variables and EVENT variables can be used in this field.
Jobclass	Specify a 1 byte identifier for the job class of a batch job issuer. Valid values are 0-9 and A-Z. SHARED variables and EVENT variables can be used in this field.
Acct Info	Enter up to 27 bytes of information about the accounting information of the issuer of the event. SHARED variables, EVENT variables, and pattern matching can be used in this field.
RACF User	A 1 to 8 byte field where you can enter the RACF user ID for the address space that issued the message. The RACF user ID is taken from the USERID keyword of the job card. SHARED variables, EVENT variables, and pattern matching can be used in this field.
RACF Group	A 1 to 8 byte field where you can enter the RACF group ID for the address space that issued the message. The RACF Group ID is taken from the GROUP keyword of the job card. SHARED variables, EVENT variables, and pattern matching can be used in this field.

"Specifying Actions for IMS-Initiated Rules" describes the actions you can specify for an IMS-initiated Rule.

Specifying Actions for IMS-Initiated Rules

Figure 287 shows an example of the Action Specification panel for IMS-initiated Rules.

```
BMC Software --------------- Action Specification - IMS  ---------- AutoOPERATOR
COMMAND ===>

       Rule-set === AAORUL00             Rule-id  === ICH70001

Automation Actions:
Display at dest. ===> YES       Journal ===>
EXEC Name/Parms  ===>
Send (TSO)       ===>
Cmd (Type   MVS) ===>

Set Variable     ===>                                    ===>
Reword Msg       ===>

Notify           ===>                                   Outboard Pager ID
Info             ===>

DOM Id           ===>                                   Delete Operator Message
Issue WTO Msg    ===>


Press END to return to Detail Control, ENTER to continue
```

Figure 287. Action Specification Panel for IMS-Initiated Rules

Use the `Action Specification - IMS` panel to specify what actions an IMS-initiated Rule will take.

Table 72 describes the Action Specification fields for an IMS-initiated Rule.

Table 72. Action Specifications for IMS-Initiated Rules

Field	Description
Display at dest.	Specifies whether the IMS message will be displayed at the destination it was originally sent to. Valid values are YES and NO. YES is the default. If you specify NO, the IMS message is suppressed. Variables cannot be used in this field.
Journal	Specifies whether the IMS message will be recorded in the BBI Journal. Valid values are YES, NO, and REW. The default is NO. A value of REW copies a reworded message to the BBI Journal. Use REW with the `Reword Msg` field.

Table 72. Action Specifications for IMS-Initiated Rules (Continued)

Field	Description
EXEC name|Parms	Allows you to specify the name of an EXEC the Rule can invoke and any parameters that you want to send to the EXEC. This field is 56 characters long. Separate the EXEC name and any parameters by a blank. Note that the first parameter specified in this field will be the first parameter passed to the EXEC. This also means the message ID and any message text will not be passed to the EXEC. To have the message ID and message text passed to the EXECs, use the variable IMFTEXT in this field. EVENT and SHARED variables can be used in this field.
Send (TSO)	Allows you to send a copy of the message using the MVS SEND command to 1-5 TSO user IDs. Separate each TSO user ID by a blank. This field is 40 characters long. EVENT and SHARED variables are valid in this field.
Cmd (Type xxx)	Enter one or more commands to be issued, where xxx represents a user-specified command type. Possible values are: **MVS** Allows you to specify MVS console commands in this field; the address space that originated the event will issue the commands **IMS** Allows you to specify IMS commands **BBI** Allows you to specify BBI commands **SS** Allows you to specify MVS commands that originate from the BBI-SS PAS **CIC** Allows you to specify CICS commands. The CICS command must contain a target; for example: `TARGET:COMMANDTEXT` **MQ** Allows you to specify MQSeries commands In addition: • This field is 126 characters long • EVENT and SHARED variables can be used in this field • You can use multiple commands by separating them with double colon marks: :: • MQSeries commands must begin with the name of the queue manager that will process the command specified. For example: `queuemgrname:command text` For documentation about how to use the Cmd(MQ) field, refer to the AutoOPERATOR *for MQSeries Installation and Reference Guide*.

Table 72. Action Specifications for IMS-Initiated Rules (Continued)

Field	Description
Set variable	Allows you to set, increment, or decrement the value of a SHARED variable. A variable's numerical value may be incremented or decremented by specifying `+n` or `-n`, where `n` is a number. The first field is 33 characters long and SHARED variables are valid. Use the first field to specify the variable to be modified. The second field is 17 characters long and SHARED variables and constants are valid. Use the second field to assign a new value (with a variable or an unsigned constant) or modify a numeric value (with a signed constant). **Note:** The variables set by using this field are set asynchronously to the firing of the Rule. In other words, there will be a slight delay before the value of the variables are set.
Reword msg	Use this field to change the message text to be sent to its destination. Note that the reworded text is used for all subsequent Rule matching comparisons if you use an Automation strategy of ALL. The length of the reworded text cannot exceed the length of the original message plus 20 characters. For example, if the length of the original text is 40 characters long, the reworded text cannot exceed 60 characters. EVENT and SHARED variables are valid in this field.
Notify	Enter the telephone number of the AutoOPERATOR Elan operator ID to be notified using an outboard pager. This field is 33 characters long. EVENT and SHARED variables are valid. Specify the telephone number exactly as it must be dialed. The Elan operator ID must exist in the Elan Operator Profile. Use this field in conjunction with the `Info` field.
Info	Specify a 1-80 alphanumeric character string of information to be displayed in the area of an outboard pager managed by AutoOPERATOR Elan. Use this field in conjunction with the `Notify` field.
DOM ID	Specify the name of a SHARED variable that contains the delete-operator-message ID of a previously issued WTO.
Issue WTO Msg	Specify a 1-125 character message to be sent to an MVS console. The message is issued without routing or descriptor codes unless specified with the R (route) or D (descriptor code) keywords (or both), and followed by a colon (:). For example, to issue a message with Descriptor code 2 and Route code 11: `Issue WTO Msg ===> D(2) R(11): EMPMSG88 &WORD1 ISSUED` The message may be selected by another Rule. The default is blank, which means no message is issued. EVENT and SHARED variables are valid.

Selecting Criteria for JES3-Initiated Rules

When you are creating a JES3-initiated Rule, the following Selection Criteria panel is displayed when you press ENTER on the Rule Detail Processor Control panel.

```
BMC Software --------------- Selection Criteria - JES3 ------------ AutoOPERATOR
COMMAND ===>

       Rule-set === AAORUL00             Rule-id  === ICH70001

Text Description:
Text ID     ===> ICH70001                          First word of Message/Cmd
Text String (Enter Below):


Issuer Identification:
Job name    ===>                                   Name of issuer
Type        ===>                                   (JOB, STC, or TSO)
Jobclass    ===>                                   Job class of issuer
Acct Info   ===>                                   Job accounting information
RACF User   ===>                                   RACF Userid
RACF Group  ===>                                   RACF Group name
Route Codes ===>
Desc Codes  ===>
Console ID  ===>                                   Origin/Destination Console
Console Name ==>                                   Origin/Dest Console Name

Press END to return to Detail Control, ENTER to continue
```

Figure 288. Selection Criteria Panel for JES3-Initiated Rules

Use the `Selection Criteria - JES3` panel to specify the attributes of a JES3-generated message that you want to create a Rule for.

Table 73 describes the Selection Criteria fields for an JES3-initiated Rule.

Table 73. Selection Criteria for JES3-Initiated Rules

Field	Description
Text ID	Enter the 1-16 byte Text ID associated with the message. The length of the Text ID is determined by the first blank character in the text. The Text ID for the message is the first word, delimited by a blank, or the first 16 characters if the first word is longer than 16 characters. To select on more information, use the `Text String` field to enter message text. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Text String	Specify message text to be matched in addition to the Text ID. In this field, you must enter the Text ID and additional information from the message text for the Rule to match before it fires. This field supports up to 75 bytes. SHARED variables, EVENT variables, and pattern matching can be used in this field.

Table 73. Selection Criteria for JES3-Initiated Rules (Continued)

Field	Description
Job name	Specify a 1 to 8 byte name of the address space that issued the event. This field is usually used with the `Type` field which specifies the type of address space issuing the event. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Type	Specify a 1 to 3 byte name for the type of address space that issued the event. Valid values are JOB, STC, or TSO. This field provides additional information about the address space that issued the event. SHARED variables and EVENT variables can be used in this field.
Jobclass	Specify a one byte identifier for the job class of a batch job issuer. Valid values are 0-9 and A-Z. SHARED variables and EVENT variables can be used in this field.
Acct Info	Enter up to 27 bytes of information about the accounting information of the issuer of the event. SHARED variables, EVENT variables, and pattern matching can be used in this field.
RACF User	A 1 to 8 byte field where you can enter the RACF user ID for the address space that issued the message. The RACF user ID is taken from the USERID keyword of the job card. SHARED variables, EVENT variables, and pattern matching can be used in this field.
RACF Group	A 1 to 8 byte field where you can enter the RACF group ID for the address space that issued the message. The RACF Group ID is taken from the GROUP keyword of the job card. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Route codes	Enter one or more MVS route codes associated with the message. Separate each route code by a blank. Valid codes are 1-128. Variables and pattern matching cannot be used in this field.
Desc codes	Enter one or more MVS descriptor codes associated with the message. Separate each descriptor code by a blank. Valid codes are 1-128. Variables and pattern matching cannot be used in this field.
Console ID	Specify the console ID of a specific MVS console associated with the message. Valid console ID numbers are 1-99. SHARED variables and EVENT variables can be used in this field.
Console Name	Specify the console name of a specific MVS console associated with the message. SHARED variables, EVENT variables, and pattern matching can be used in this field.

"Specifying Actions for JES3-Initiated Rules" describes the actions you can specify for an JES3-initiated Rule.

Specifying Actions for JES3-Initiated Rules

Figure 289 shows an example of the Action Specification panel for JES3-initiated Rules.

```
BMC Software --------------- Action Specification - JES3 ---------- AutoOPERATOR
COMMAND ===>

       Rule-set === AAORUL00             Rule-id  === ICH70001

Automation Actions:
Display at dest. ===> YES       Journal ===>
EXEC Name/Parms  ===>
Send (TSO)       ===>
Cmd (Type   MVS) ===>

Set Variable     ===>                                    ===>
Reword Msg       ===>

Notify           ===>                                  Outboard Pager ID
Info             ===>

DOM Id           ===>                                  Delete Operator Message
Issue WTO Msg    ===>



Press END to return to Detail Control, ENTER to continue
```

Figure 289. Action Specification Panel for JES3-Initiated Rules

Use the `Action Specification - JES3` panel to specify what actions a JES3-initiated Rule will take. Table 74 describes the Action Specification fields for an JES3-initiated Rule.

Table 74. Action Specifications for JES3-Initiated Rules

Field	Description
Display at dest.	Specifies whether the JES3 message will be displayed at the destination it was originally sent to. Valid values are YES and NO. YES is the default. If you specify NO, the JES3 message is suppressed. Variables cannot be used in this field.
Journal	Specifies whether the JES3 message will be recorded in the BBI Journal. Valid values are YES, NO, and REW. The default is NO. A value of REW copies a reworded message to the BBI Journal. Use REW with the `Reword Msg` field.

Table 74. Action Specifications for JES3-Initiated Rules (Continued)

Field	Description
EXEC name|Parms	Allows you to specify the name of an EXEC the Rule can invoke and any parameters that you want to send to the EXEC. This field is 56 characters long. Separate the EXEC name and any parameters by a blank. Note that the first parameter specified in this field will be the first parameter passed to the EXEC. This also means the message ID and any message text will not be passed to the EXEC. To have the message ID and message text passed to the EXECs, use the variable IMFTEXT in this field. EVENT and SHARED variables can be used in this field.
Send (TSO)	Allows you to send a copy of the message using the MVS SEND command to 1-5 TSO user IDs. Separate each TSO user ID by a blank. This field is 40 characters long. EVENT and SHARED variables are valid in this field.
Cmd (Type xxx)	Enter one or more commands to be issued, where xxx represents a user-specified command type. Possible values are: **MVS** Allows you to specify MVS console commands in this field; the address space that originated the event will issue the commands **IMS** Allows you to specify IMS commands **BBI** Allows you to specify MVS commands that originate from the BBI-SS PAS **SS** Allows you to specify commands originating from the BBI subsystem **CIC** Allows you to specify CICS commands. The CICS command must contain a target; for example: `TARGET:COMMANDTEXT` **MQ** Allows you to specify MQSeries commands In addition: • This field is 126 characters long • EVENT and SHARED variables can be used in this field • You can use multiple commands by separating them with double colon marks: :: • MQSeries commands must begin with the name of the queue manager that will process the command specified. For example: `queuemgrname:command text` For documentation about how to use the Cmd(MQ) field, refer to the AutoOPERATOR *for MQSeries Installation and Reference Guide*.

Table 74. Action Specifications for JES3-Initiated Rules (Continued)

Field	Description
Set variable	Allows you to set, increment, or decrement the value of a SHARED variable. A variable's numerical value may be incremented or decremented by specifying `+n` or `-n`, where `n` is a number. The first field is 33 characters long and SHARED variables are valid. Use the first field to specify the variable to be modified. The second field is 17 characters long and SHARED variables and constants are valid. Use the second field to assign a new value (with a variable or an unsigned constant) or modify a numeric value (with a signed constant). **Note:** The variables set by using this field are set asynchronously to the firing of the Rule. In other words, there will be a slight delay before the value of the variables are set.
Reword msg	Use this field to change the message text to be sent to its destination. Note that the reworded text is used for all subsequent Rule matching comparisons if you use an Automation strategy of ALL. The length of the reworded text cannot exceed the length of the original message plus 20 characters. For example, if the length of the original text is 40 characters long, the reworded text cannot exceed 60 characters. EVENT and SHARED variables are valid in this field.
Notify	Enter the telephone number of the AutoOPERATOR Elan operator ID to be notified using an outboard pager. This field is 33 characters long. EVENT and SHARED variables are valid. Specify the telephone number exactly as it must be dialed. The Elan operator ID must exist in the Elan Operator Profile. Use this field in conjunction with the `Info` field.
Info	Specify a 1-80 alphanumeric character string of information to be displayed in the area of an outboard pager managed by AutoOPERATOR Elan. Use this field in conjunction with the `Notify` field.
DOM ID	Specify the name of a SHARED variable that contains the delete-operator-message ID of a previously issued WTO.
Issue WTO Msg	Specify a 1-125 character message to be sent to an MVS console. The message is issued without routing or descriptor codes unless specified with the R (route) or D (descriptor code) keywords (or both), and followed by a colon (:). For example, to issue a message with Descriptor code 2 and Route code 11: `Issue WTO Msg ===> D(2) R(11): EMPMSG88 &WORD1 ISSUED` The message may be selected by another Rule. The default is blank, which means no message is issued. EVENT and SHARED variables are valid.

Selecting Criteria for JRNL-Initiated Rules

When you are creating a BBI Journal-initiated Rule, the following Selection Criteria panel is displayed when you press ENTER on the Rule Detail Processor Control panel.

```
BMC Software --------------- Selection Criteria - JRNL ------------ AutoOPERATOR
COMMAND ===>

       Rule-set === AAORUL00             Rule-id  === ICH70001

Text Description:
Text ID     ===> ICH70001                          First word of Message/CMB
Text String (Enter Below):


Source of Message/Command destined for Journal:
Origin      ===>                                   Name of issuer









Press END to return to Detail Control, ENTER to continue
```

Figure 290. Selection Criteria Panel for JRNL-Initiated Rules

Use the `Selection Criteria - JRNL` panel to specify the attributes of a message destined for the BBI Journal that you want to create a Rule for.

Table 75 describes the Selection Criteria fields for an JRNL-initiated Rule.

Table 75. Selection Criteria for JRNL-Initiated Rules

Field	Description
Text ID	Enter the 1-16 byte Text ID associated with the message. The length of the Text ID is determined by the first blank character in the text. The Text ID for the message is the first word, delimited by a blank, or the first 16 characters if the first word is longer than 16 characters. To select on more information, use the `Text String` field to enter message text. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Text String	Specify message text to be matched in addition to the Text ID. In this field, you must enter the Text ID and additional information from the message text for the Rule to match before it fires. This field supports up to 75 bytes. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Origin	Enter the name of the originating address space that issued the message for the BBI Journal. SHARED variables, EVENT variables, and pattern matching can be used in this field.

“Specifying Actions for JRNL-Initiated Rules” describes the actions you can specify for an JRNL-initiated Rule.

Specifying Actions for JRNL-Initiated Rules

Figure 291 shows an example of the Action Specification panel for JRNL-initiated Rules.

```
BMC Software --------------- Action Specification - JRNL ---------- AutoOPERATOR
COMMAND ===>

       Rule-set === AAORUL00            Rule-id  === ICH70001

Automation Actions:
Display at dest. ===> YES
EXEC Name/Parms  ===>
Send (TSO)       ===>
Cmd              ===>

Set Variable     ===>                                    ===>
Reword Msg       ===>

Notify           ===>                                  Outboard Pager ID
Info             ===>

DOM Id           ===>                                  Delete Operator Message
Issue WTO Msg    ===>



Press END to return to Detail Control, ENTER to continue
```

Figure 291. Action Specification Panel for JRNL-Initiated Rules

Table 76 describes the Action Specification fields for an JRNL-initiated Rule.

Table 76. Action Specifications for JRNL-Initiated Rules

Field	Description
Display at dest.	Specifies whether the message sent to the BBI Journal will be displayed in the BBI Journal. Valid values are YES and NO. YES is the default. If you specify NO, the message is suppressed. Variables cannot be used in this field.
EXEC name|Parms	Allows you to specify the name of an EXEC the Rule can invoke and any parameters that you want to send to the EXEC. This field is 56 characters long. Separate the EXEC name and any parameters by a blank. Note that the first parameter specified in this field will be the first parameter passed to the EXEC. This also means the message ID and any message text will not be passed to the EXEC. To have the message ID and message text passed to the EXECs, use the variable IMFTEXT in this field. EVENT and SHARED variables can be used in this field.
Send (TSO)	Allows you to send a copy of the message using the MVS SEND command to 1-5 TSO user IDs. Separate each TSO user ID by a blank. This field is 40 characters long. EVENT and SHARED variables are valid in this field.

Table 76. Action Specifications for JRNL-Initiated Rules (Continued)

Field	Description
Cmd (Type xxx)	Enter one or more commands to be issued, where xxx represents a user-specified command type. Possible values are: **MVS** Allows you to specify MVS console commands in this field; the address space that originated the event will issue the commands **IMS** Allows you to specify IMS commands **BBI** Allows you to specify BBI commands **SS** Allows you to specify MVS commands that originate from the BBI-SS PAS **CIC** Allows you to specify CICS commands. The CICS command must contain a target; for example: `TARGET:COMMANDTEXT` **MQ** Allows you to specify MQSeries commands In addition: • This field is 126 characters long • EVENT and SHARED variables can be used in this field • You can use multiple commands by separating them with double colon marks: :: • MQSeries commands must begin with the name of the queue manager that will process the command specified. For example: `queuemgrname:command text` For documentation about how to use the Cmd(MQ) field, refer to the AutoOPERATOR *for MQSeries Installation and Reference Guide*.
Set variable	Allows you to set, increment, or decrement the value of a SHARED variable. A variable's numerical value may be incremented or decremented by specifying `+n` or `-n`, where `n` is a number. The first field is 33 characters long and SHARED variables are valid. Use the first field to specify the variable to be modified. The second field is 17 characters long and SHARED variables and constants are valid. Use the second field to assign a new value (with a variable or an unsigned constant) or modify a numeric value (with a signed constant). **Note:** The variables set by using this field are set asynchronously to the firing of the Rule. In other words, there will be a slight delay before the value of the variables are set.

Table 76. Action Specifications for JRNL-Initiated Rules (Continued)

Field	Description
Reword Msg	Use this field to change the message text to be sent to its destination. Note that the reworded text is used for all subsequent Rule matching comparisons if you use an Automation strategy of ALL. The length of the reworded text cannot exceed the length of the original message. EVENT and SHARED variables are valid in this field.
Notify	Enter the telephone number of the AutoOPERATOR Elan operator ID to be notified using an outboard pager. This field is 33 characters long. EVENT and SHARED variables are valid. Specify the telephone number exactly as it must be dialed. The Elan operator ID must exist in the Elan Operator Profile. Use this field in conjunction with the `Info` field.
Info	Specify a 1-80 alphanumeric character string of information to be displayed in the area of an outboard pager managed by AutoOPERATOR Elan. Use this field in conjunction with the `Notify` field.
DOM ID	Specify the name of a SHARED variable that contains the delete-operator-message ID of a previously issued WTO.
Issue WTO Msg	Specify a 1-125 character message to be sent to an MVS console. The message is issued without routing or descriptor codes unless specified with the R (route) or D (descriptor code) keywords (or both), and followed by a colon (:). For example, to issue a message with Descriptor code 2 and Route code 11: `Issue WTO Msg ===> D(2) R(11): EMPMSG88 &WORD1 ISSUED` The message may be selected by another Rule. The default is blank, which means no message is issued. EVENT and SHARED variables are valid.

Selecting Criteria for MSG-Initiated Rules

When you are creating a MSG-initiated Rule, the following Selection Criteria panel is displayed when you press ENTER on the Rule Detail Processor Control panel.

```
BMC Software --------------- Selection Criteria - MSG  ------------ AutoOPERATOR
COMMAND ===>

       Rule-set === AAORUL00             Rule-id  === ICH70001

Text Description:
Text ID     ===> ICH70001                          Ignore leading plus: Y
Text String (Enter Below):


Issuer Identification:
Job name    ===>                                   Name of issuer
Type        ===>                                   (JOB, STC, or TSO)
Jobclass    ===>                                   Job class of issuer
Acct Info   ===>                                   Job accounting information
RACF User   ===>                                   RACF Userid
RACF Group  ===>                                   RACF Group name
Route Codes ===>
Desc Codes  ===>
Console ID  ===>                                   Destination Console
Console Name ==>                                   Destination Console Name

Press END to return to Detail Control, ENTER to continue
```

Figure 292. Selection Criteria Panel for MSG-Initiated Rules

Use the `Selection Criteria - MSG` panel to specify the attributes of a message that you want to create a Rule for.

Table 77 describes the Selection Criteria fields for an MSG-initiated Rule.

Table 77. Selection Criteria for MSG-Initiated Rules

Field	Description
Ignore non-auth	Enter Y to ignore a leading plus **+** in the message. Messages issued from a non-authorized program will have a plus sign inserted at the beginning of the message. Using this option allows messages that are issued from both authorized and non-authorized programs to be trapped by a single rule. If you specify this option, take additional steps to ensure that the message is being issued from a trusted job, such as by verifying the user ID, job name, accounting code, or some other information provided by the Rule or EXEC. Note: Using this option must be done with care to ensure that your Rules do not fire accidentally.
Text ID	Enter the 1-16 byte Text ID associated with the message. The length of the Text ID is determined by the first blank character in the text. The Text ID for the message is the first word, delimited by a blank, or the first 16 characters if the first word is longer than 16 characters. To select on more information, use the `Text String` field to enter message text. SHARED variables, EVENT variables, and pattern matching can be used in this field.

Table 77. Selection Criteria for MSG-Initiated Rules (Continued)

Field	Description
Text String	Specify message text to be matched in addition to the Text ID. In this field, you must enter the Text ID and additional information from the message text for the Rule to match before it fires. This field supports up to 75 bytes. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Job name	Specify a 1 to 8 byte name of the address space that issued the event. This field is usually used with the `Type` field which specifies the type of address space issuing the event. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Type	Specify a 1 to 3 byte name for the type of address space that issued the event. Valid values are JOB, STC, or TSO. This field provides additional information about the address space that issued the event. SHARED variables and EVENT variables can be used in this field.
Jobclass	Specify a one byte identifier for the job class of a batch job issuer. Valid values are 0-9 and A-Z. SHARED variables and EVENT variables can be used in this field.
Acct Info	Enter up to 27 bytes of information about the accounting information of the issuer of the event. SHARED variables, EVENT variables, and pattern matching can be used in this field.
RACF User	A 1 to 8 byte field where you can enter the RACF user ID for the address space that issued the message. The RACF user ID is taken from the USERID keyword of the job card. SHARED variables, EVENT variables, and pattern matching can be used in this field.
RACF Group	A 1 to 8 byte field where you can enter the RACF group ID for the address space that issued the message. The RACF Group ID is taken from the GROUP keyword of the job card. SHARED variables, EVENT variables, and pattern matching can be used in this field.
Route codes	Enter one or more MVS route codes associated with the message. Separate each route code by a blank. Valid codes are 1-128. Variables and pattern matching cannot be used in this field.
Desc codes	Enter one or more MVS descriptor codes associated with the message. Separate each descriptor code by a blank. Valid codes are 1-128. Variables and pattern matching cannot be used in this field.

Table 77. Selection Criteria for MSG-Initiated Rules (Continued)

Field	Description
Console ID	Specify the console ID of a specific MVS console associated with the message. Valid console ID numbers are 1-99. SHARED variables and EVENT variables can be used in this field.
Console Name	Specify the console name of a specific MVS console associated with the message. SHARED variables, EVENT variables, and pattern matching can be used in this field.

"Specifying Actions for MSG-Initiated Rules" describes the actions you can specify for an MSG-initiated Rule.

Specifying Actions for MSG-Initiated Rules

Use this panel to specify the actions for a MSG-initiated Rule.

```
BMC Software --------------- Action Specification - MSG  ---------- AutoOPERATOR
COMMAND ===>

       Rule-set === AAORUL00             Rule-id  === ICH70001
Automation Actions:
Display at dest. ===> YES       Journal ===>           SYSLOG Display ===>
EXEC Name/Parms  ===>
Send (TSO)       ===>
Cmd (Type   MVS) ===>

Set Variable     ===>                                    =
Reword Msg       ===>

Notify           ===>                                   Outboard Pager ID
Info             ===>

Route codes      ===>
Descriptor codes ===>
Console ID       ===>                                   Destination Console ID
Console Name     ===>                                   Dest. Console ID Name
Update Rout/Desc Codes ===>                             (ADD/REPL)
Issue WTO Msg    ===>

DOM MSG          ===>                                   Issue message when DOM
Press END to return to Detail Control, ENTER to continue
```

Figure 293. Action Specification Panel for MSG-Initiated Rules

Use the `Action Specification - MSG` panel to specify what actions an MSG-initiated Rule will take.

Table 78 describes the Action Specification fields for an MSG-initiated Rule.

Table 78. Action Specifications for MSG-Initiated Rules

Field	Description
Display at dest.	Specifies whether the message will be displayed at the destination it was originally sent to. Valid values are YES and NO. YES is the default. If you specify NO, the message is suppressed. Variables cannot be used in this field.
Journal	Specifies whether the message will be recorded in the BBI Journal. Valid values are YES, NO, and REW. The default is NO. A value of REW copies a reworded message to the BBI Journal. Use REW with the `Reword Msg` field.
SYSLOG Display	Specifies whether the message will be recorded in the MVS SYSLOG. Valid values are YES and NO. YES is the default.

Table 78. Action Specifications for MSG-Initiated Rules (Continued)

Field	Description
EXEC name|Parms	Allows you to specify the name of an EXEC the Rule can invoke and any parameters that you want to send to the EXEC. This field is 56 characters long. Separate the EXEC name and any parameters by a blank. Note that the first parameter specified in this field will be the first parameter passed to the EXEC. This also means the message ID and any message text will not be passed to the EXEC. To have the message ID and message text passed to the EXECs, use the variable IMFTEXT in this field. EVENT and SHARED variables can be used in this field.
Send (TSO)	Allows you to send a copy of the message using the MVS SEND command to 1-5 TSO user IDs. Separate each TSO user ID by a blank. This field is 40 characters long. EVENT and SHARED variables are valid in this field.
Cmd (Type xxx)	Enter one or more commands to be issued, where xxx represents a user-specified command type. Possible values are: **MVS** Allows you to specify MVS console commands in this field; the address space that originated the event will issue the commands **IMS** Allows you to specify IMS commands **BBI** Allows you to specify BBI commands **SS** Allows you to specify MVS commands that originate from the BBI-SS PAS **CIC** Allows you to specify CICS commands. The CICS command must contain a target; for example: `TARGET:COMMANDTEXT` **MQ** Allows you to specify MQSeries commands In addition: • This field is 126 characters long • EVENT and SHARED variables can be used in this field • You can use multiple commands by separating them with double colon marks: :: • MQSeries commands must begin with the name of the queue manager that will process the command specified. For example: `queuemgrname:command text` For documentation about how to use the Cmd(MQ) field, refer to the AutoOPERATOR *for MQSeries Installation and Reference Guide*.

Table 78. Action Specifications for MSG-Initiated Rules (Continued)

Field	Description
Set variable	Allows you to set, increment, or decrement the value of a SHARED variable. A variable's numerical value may be incremented or decremented by specifying `+n` or `-n`, where `n` is a number. The first field is 33 characters long and SHARED variables are valid. Use the first field to specify the variable to be modified. The second field is 17 characters long and SHARED variables and constants are valid. Use the second field to assign a new value (with a variable or an unsigned constant) or modify a numeric value (with a signed constant). **Note:** The variables set by using this field are set asynchronously to the firing of the Rule. In other words, there will be a slight delay before the value of the variables are set.
Reword Msg	Use this field to change the message text to be sent to its destination. Note that the reworded text is used for all subsequent Rule matching comparisons if you use an Automation strategy of ALL. The length of the reworded text cannot exceed the length of the original message plus 20 characters. For example, if the length of the original text is 40 characters long, the reworded text cannot exceed 60 characters. EVENT and SHARED variables are valid in this field.
Notify	Enter the telephone number of the AutoOPERATOR Elan operator ID to be notified using an outboard pager. This field is 33 characters long. EVENT and SHARED variables are valid. Specify the telephone number exactly as it must be dialed. The Elan operator ID must exist in the Elan Operator Profile. Use this field in conjunction with the `Info` field.
Info	Specify a 1-80 alphanumeric character string of information to be displayed in the area of an outboard pager managed by AutoOPERATOR Elan. Use this field in conjunction with the `Notify` field.
Console ID	Specify a message's new destination console ID. This field is 33 characters long. SHARED variables are valid. Only one console ID can be specified. A value from 1-99 is valid.
Console Name	Specify a message's new destination console name. This field is 33 characters long. SHARED variables are valid. Only one console name can be specified. Console names are 1-8 characters long. Named consoles are available only on MVS/SP4.

Table 78. Action Specifications for MSG-Initiated Rules (Continued)

Field	Description
Route codes	A 61 character field that allows you to add new MVS route codes or replace the original MVS route codes of the message. Separate each route code entered with a blank. This field must be used in conjunction the `Update Rout/Desc Codes` field to specify either replacement or adding of new codes. Variables cannot be used in this field.
Descriptor codes	A 38-character field that allows you to add new MVS descriptor codes or replace the original MVS descriptor codes of the message. Separate each descriptor code entered with a blank. This field must be used in conjunction the `Update Rout/Desc Codes` field to specify either replacement or adding of new codes. Variables cannot be used in this field.
Update Rout/Desc Codes	A field that allows you add or replace route and descriptor codes specified in the `Route Codes` or `Desc Codes` fields. Valid values are ADD (to add new codes) or REPL (to replace old codes with new ones specified). Variables cannot be used in this field.
Issue WTO Msg	Specify a 1-125 character message to be sent to an MVS console. The message is issued without routing or descriptor codes unless specified with the R (route) or D (descriptor code) keywords (or both), and followed by a colon (:). For example, to issue a message with Descriptor code 2 and Route code 11: `Issue WTO Msg ===> D(2) R(11): EMPMSG88 &WORD1 ISSUED` The message may be selected by another Rule. The default is blank, which means no message is issued. EVENT and SHARED variables are valid.
DOMID	Specify the name of a SHARED variable that contains the delete-operator-message ID of a previously issued WTO.

Selecting Criteria for Time-Initiated Rules

When you are creating a Time-initiated Rule, the following Selection Criteria panel is displayed when you press ENTER on the Rule Detail Processor Control panel.

```
BMC Software --------------- Selection Criteria - TIME ------------ AutoOPERATOR
COMMAND ===>

       Rule-set === AAORULZZ             Rule-id  === ICH70001

Start Specification:
Start Time           ===>                   hh:mm:ss or ATSTART

Frequency:
Interval             ===>                   hh:mm:ss

Stop Specification:
Stop Time            ===>                   hh:mm:ss
          or
Stop Count           ===>









Press END to return to Detail Control, ENTER to continue
```

Figure 294. Selection Criteria Panel for Time-Initiated Rules

Use the `Selection Criteria - TIME` panel to specify at what time and at what intervals you would like to have a Rule automatically fired.

Table 79 describes the Selection Criteria fields for an TIME-initiated Rule.

Table 79. Selection Criteria for TIME-Initiated Rules

Field	Description
Start Time	Specifies (in hours, minutes, and seconds) at what clock time the Rule begins firing. For example, `10:05:30` means the Rule begins firing at 5 minutes and 30 seconds after 10:00 am. You can enter a start time of `ATSTART`. This specifies that Rules will begin firing at BBI-SS start-up.
Interval	Specifies (in hours, minutes, and seconds) how often the Rule fires. The minimum interval is 5 seconds. For example, `00:05:00` means the Rule fires at the start time and every 5 minutes thereafter until the Stop Time or the Stop Count is reached.

Table 79. Selection Criteria for TIME-Initiated Rules (Continued)

Field	Description
Stop Time	Specifies (in hours, minutes, and seconds) at what clock time the Rule should stop firing. If you do not specify a Stop Time or a Count value, the default is Stop Count=1. You can specify a Stop Time or a Stop Count but not both.
Stop Count	Specifies the number of times for the Rule to be fired from the Start Time. For example, a Stop Count value of 12 means that the Rule will begin firing at the specified Start Time and continue firing (at a rate determined by the Interval field) 12 times. If you do not specify a Stop Time or a Count value, the default is Stop Count=1. You can specify a Stop Time or a Stop Count but not both.

A Rule must be enabled at least 121 seconds (two minutes and one second) before the next start interval that it is scheduled to be fired. If it is not enabled, the following interval is used as the start time.

For example, suppose a Rule specifies a start time of 10:00:00 with an interval of one hour. If it is enabled at 10:58:01, it will fire for the first time at 12:00:00. If it is enabled at 10:57:59, it will fire for the first time at 11:00:00.

The following sections show examples for using the fields.

Examples

The following shows two examples of using the fields on the Selection Criteria panel for a Time-initiated Rule.

Example 1: To create a Time-initiated Rule that first fires at 9:00 am and every 15 minutes thereafter until 4:00 pm, fill out the Selection Criteria panel as follows.

```
Start Specification:
Start Time           ===>  09:00:00         hh:mm:ss or ATSTART

Frequency:
Interval             ===>  00:15:00         hh:mm:ss

Stop Specification:
Stop Time            ===>  16:00:00         hh:mm:ss
          or
Stop Count           ===>
```

Example 2: To create a Time-initiated Rule that fires 15 times beginning from 4:00 pm and every half hour thereafter, fill out the Selection Criteria panel as follows.

```
Start Specification:
Start Time           ===>  16:00:00         hh:mm:ss or ATSTART

Frequency:
Interval             ===>  00:30:00         hh:mm:ss

Stop Specification:
Stop Time            ===>                   hh:mm:ss
          or
Stop Count           ===>  15
```

Important

If the subsystem on which Time-initiated Rules are fired comes up after the specified Start Time and before the specified Stop Time or stop time implied by the Stop Count value, the firing of the Rules will behave as though the Rule fired on the Start Time and every interval thereafter.

For example, in example 2, if the subsystem came up at 7:35 pm, then Rule will fire at 8:00 pm and 7 times (and not the specified 15 times) thereafter until 11:00 pm.

“Specifying Actions for Time-Initiated Rules” describes the actions you can specify for an TIME-initiated Rule.

Specifying Actions for Time-Initiated Rules

Figure 295 shows an example of the Action Specification panel for TIME-initiated Rules.

```
BMC Software --------------- Action Specification - TIME ---------- AutoOPERATOR
COMMAND ===>

       Rule-set === AAORULZZ             Rule-id  === ICH70001

Automation Actions:
Journal          ===>
EXEC Name/Parms  ===>
Send (TSO)       ===>
Cmd              ===>

Set Variable     ===>                                  ===>


Notify           ===>                                 Outboard Pager ID
Info             ===>

DOM Id           ===>                                 Delete Operator Message
Issue WTO Msg    ===>



Press END to return to Detail Control, ENTER to continue
```

Figure 295. Action Specification Panel for Time-Initiated Rules

Use the `Action Specification - TIME` panel to specify what actions an Time-initiated Rule will take.

Table 80 describes the Action Specification fields for an TIME-initiated Rule.

Table 80. Action Specifications for TIME-Initiated Rules

Field	Description
Journal	Specifies whether a message is written to the Journal. Valid values are YES and NO. The default is NO. If you specify YES, then the following message is issued: `AU0001I RULE rrrrrrrr IN RULESET ssssssss FIRED.` where rrrrrrrr is the Rule-ID and ssssssss is the Rule Set name.
EXEC name|Parms	Allows you to specify the name of an EXEC the Rule can invoke and any parameters that you want to send to the EXEC. This field is 56 characters long. Separate the EXEC name and any parameters by a blank. Note that the first parameter specified in this field will be the first parameter passed to the EXEC. This also means the message ID and any message text will not be passed to the EXEC. To have the message ID and message text passed to the EXECs, use the variable IMFTEXT in this field. EVENT and SHARED variables can be used in this field.

Table 80. Action Specifications for TIME-Initiated Rules (Continued)

Field	Description
Send (TSO)	Allows you to send a copy of the message using the MVS SEND command to 1-5 TSO user IDs. Separate each TSO user ID by a blank. This field is 40 characters long. EVENT and SHARED variables are valid in this field.
Cmd (Type xxx)	Enter one or more commands to be issued, where xxx represents a user-specified command type. Possible values are: **MVS** Allows you to specify MVS console commands in this field; the address space that originated the event will issue the commands **IMS** Allows you to specify IMS commands **BBI** Allows you to specify BBI commands **SS** Allows you to specify MVS commands that originate from the BBI-SS PAS **CIC** Allows you to specify CICS commands. The CICS command must contain a target; for example: `TARGET:COMMANDTEXT` **MQ** Allows you to specify MQSeries commands In addition: • This field is 126 characters long • EVENT and SHARED variables can be used in this field • You can use multiple commands by separating them with double colon marks: :: • MQSeries commands must begin with the name of the queue manager that will process the command specified. For example: `queuemgrname:command text` For documentation about how to use the Cmd(MQ) field, refer to the AutoOPERATOR *for MQSeries Installation and Reference Guide*.
Set variable	Allows you to set, increment, or decrement the value of a SHARED variable. A variable's numerical value may be incremented or decremented by specifying `+n` or `-n`, where `n` is a number. The first field is 33 characters long and SHARED variables are valid. Use the first field to specify the variable to be modified. The second field is 17 characters long and SHARED variables and constants are valid. Use the second field to assign a new value (with a variable or an unsigned constant) or modify a numeric value (with a signed constant). **Note:** The variables set by using this field are set asynchronously to the firing of the Rule. In other words, there will be a slight delay before the value of the variables are set.

Table 80. Action Specifications for TIME-Initiated Rules (Continued)

Field	Description
Reword Msg	Use this field to change the message text to be sent to its destination. Note that the reworded text is used for all subsequent Rule matching comparisons if you use an Automation strategy of ALL. EVENT and SHARED variables are valid in this field.
Notify	Enter the telephone number of the AutoOPERATOR Elan operator ID to be notified using an outboard pager. This field is 33 characters long. EVENT and SHARED variables are valid. Specify the telephone number exactly as it must be dialed. The Elan operator ID must exist in the Elan Operator Profile. Use this field in conjunction with the `Info` field.
Info	Specify a 1-80 alphanumeric character string of information to be displayed in the area of an outboard pager managed by AutoOPERATOR Elan. Use this field in conjunction with the `Notify` field.
DOM ID	Specify the name of a SHARED variable that contains the delete-operator-message ID of a previously issued WTO.
Issue WTO Msg	Specify a 1-125 character message to be sent to an MVS console. The message is issued without routing or descriptor codes unless specified with the R (route) or D (descriptor code) keywords (or both), and followed by a colon (:). For example, to issue a message with Descriptor code 2 and Route code 11: `Issue WTO Msg ===> D(2) R(11): EMPMSG88 &WORD1 ISSUED` The message may be selected by another Rule. The default is blank, which means no message is issued. EVENT and SHARED variables are valid.

Selecting Criteria for VAR-Initiated Rules

When you are creating a VAR-initiated Rule, the following Selection Criteria panel is displayed when you press ENTER on the Rule Detail Processor Control panel.

```
BMC Software --------------- Selection Criteria - VAR  ------------ AutoOPERATOR
COMMAND ===>                                                 TGT  --- AO62

       Rule-set === RULMAGIE             Rule-id  === RULE0011

Variable Description:  (Shared variable only)
Name        ===>

Fire this Rule when:

    Variable is  ===>                                   (Created,Deleted,Upd)



Press END to return to Detail Control, ENTER to continue
```

Figure 296. Selection Criteria Panel for VAR-Initiated Rules

Use the `Selection Criteria - VAR` panel to specify the attributes of a VAR generated message that you want to create a Rule for.

Table 81 describes the Selection Criteria fields for an VAR-initiated Rule.

Table 81. Selection Criteria for VAR-Initiated Rules

Field	Description
Name	Enter the variable name of a SHARED pool variable that when its value changes, will trigger this Rule
Variable is	Specify the action that will fire the Rule; valid values are: **Created** The Rule will fire when the variable is created in the SHARED pool **Deleted** The Rule will fire when variable is deleted from the SHARED pool **Upd** The Rule will fire when the value of the variable is updated in the SHARED pool

“Specifying Actions for VAR-Initiated Rules” describes the actions you can specify for an VAR-initiated Rule.

Specifying Actions for VAR-Initiated Rules

Figure 297 shows an example of the Action Specification panel for VAR-initiated Rules.

Use this panel to specify the actions for a VAR-initiated Rule.

```
BMC Software --------------- Action Specification - VAR  ---------- AutoOPERATOR
COMMAND ===>                                                  TGT  --- AO62--

       Rule-set === RULMAGIE             Rule-id  === DUMMY001

Automation Actions:
Journal          ===> YES
EXEC Name/Parms  ===>
Send (TSO IDs)   ===>
Cmd (Type  MVS ) ===>

Set Variable     ===>                                      ===>


Notify           ===>                                    Outboard Pager ID
Info             ===> _________

DOM Id           ===>                                    Delete Operator Message
Issue WTO Msg    ===> ________________________________ __  _________________



Press END to return to Detail Control, ENTER to continue
```

Figure 297. Action Specification Panel for VAR-Initiated Rules

Use the `Action Specification - VAR` panel to specify what actions a VAR-initiated Rule will take.

Table 82 describes the Action Specification fields for an VAR-initiated Rule.

Table 82. Action Specifications for VAR-Initiated Rules

Field	Description
Journal	Specifies whether the message will be recorded in the BBI Journal. Valid values are YES and NO. The default is NO.
EXEC name|Parms	Allows you to specify the name of an EXEC the Rule can invoke and any parameters that you want to send to the EXEC. This field is 56 characters long. Separate the EXEC name and any parameters by a blank. Note that the first parameter specified in this field will be the first parameter passed to the EXEC. This also means the message ID and any message text will not be passed to the EXEC. To have the message ID and message text passed to the EXECs, use the variable IMFTEXT in this field. EVENT and SHARED variables can be used in this field.

Table 82. Action Specifications for VAR-Initiated Rules (Continued)

Field	Description
Send (TSO)	Allows you to send a copy of the message using the MVS SEND command to 1-5 TSO user IDs. Separate each TSO user ID by a blank. This field is 40 characters long. EVENT and SHARED variables are valid in this field.
Cmd (Type xxx)	Enter one or more commands to be issued, where xxx represents a user-specified command type. Possible values are: **MVS** Allows you to specify MVS console commands in this field; the address space that originated the event will issue the commands **IMS** Allows you to specify IMS commands **BBI** Allows you to specify BBI commands **SS** Allows you to specify MVS commands that originate from the BBI-SS PAS **CIC** Allows you to specify CICS commands. The CICS command must contain a target; for example: `TARGET:COMMANDTEXT` **MQ** Allows you to specify MQSeries commands In addition: • This field is 126 characters long • EVENT and SHARED variables can be used in this field • You can use multiple commands by separating them with double colon marks: :: • MQSeries commands must begin with the name of the queue manager that will process the command specified. For example: `queuemgrname:command text` For documentation about how to use the Cmd(MQ) field, refer to the AutoOPERATOR *for MQSeries Installation and Reference Guide*.
Set variable	Allows you to set, increment, or decrement the value of a SHARED variable. A variable's numerical value may be incremented or decremented by specifying `+n` or `-n`, where `n` is a number. The first field is 33 characters long and SHARED variables are valid. Use the first field to specify the variable to be modified. The second field is 17 characters long and SHARED variables and constants are valid. Use the second field to assign a new value (with a variable or an unsigned constant) or modify a numeric value (with a signed constant). **Note:** The variables set by using this field are set asynchronously to the firing of the Rule. In other words, there will be a slight delay before the value of the variables are set.

Table 82. Action Specifications for VAR-Initiated Rules (Continued)

Field	Description
Notify	Enter the telephone number of the AutoOPERATOR Elan operator ID to be notified using an outboard pager. This field is 33 characters long. EVENT and SHARED variables are valid. Specify the telephone number exactly as it must be dialed. The Elan operator ID must exist in the Elan Operator Profile. Use this field in conjunction with the `Info` field.
Info	Specify a 1-80 alphanumeric character string of information to be displayed in the area of an outboard pager managed by AutoOPERATOR Elan. Use this field in conjunction with the `Notify` field.
DOM ID	Specify the name of a SHARED variable that contains the delete-operator-message ID of a previously issued WTO.
Issue WTO Msg	Specify a 1-125 character message to be sent to an MVS console. The message is issued without routing or descriptor codes unless specified with the R (route) or D (descriptor code) keywords (or both), and followed by a colon (:). For example, to issue a message with Descriptor code 2 and Route code 11: `Issue WTO Msg ===> D(2) R(11): EMPMSG88 &WORD1 ISSUED` The message may be selected by another Rule. The default is blank, which means no message is issued. EVENT and SHARED variables are valid.

Part 7. Glossary and Index

This part contains the glossary and the index.

Glossary 603

Index 615

Glossary

This glossary defines BMC Software terminology. Other dictionaries and glossaries can be used in conjunction with this glossary.

Since this glossary pertains to BMC Software-related products, some of the terms defined might not appear in this book.

To help you find the information you need, this glossary uses the following cross-references:

Contrast with	Indicates a term that has a contrary or contradictory meaning.
See	Indicates an entry that is a synonym or contains expanded information.
See also	Indicates an entry that contains related information.

A

action. Defined operation, such as modifying a MAINVIEW window, that is performed in response to a command. *See* object.

active window. Any MAINVIEW window in which data can be refreshed. *See* alternate window, current window, window.

administrative view. Display from which a product's management tasks are performed, such as the DSLIST view for managing historical data sets. *See* view.

ALT WIN field. Input field that allows you to specify the window identifier for an alternate window where the results of a hyperlink are displayed. *See* alternate window.

Alternate Access. *See* MAINVIEW Alternate Access.

alternate form. View requested through the FORM command that changes the format of a previously displayed view to show related information. *See also* form, query.

alternate window. (1) Window that is specifically selected to display the results of a hyperlink. (2) Window whose identifier is defined to the ALT WIN field. *Contrast with* current window. *See* active window, window, ALT WIN field.

analyzer. (1) Online display that presents a snapshot of status and activity data and indicates problem areas. (2) Component of CMF MONITOR. *See* CMF MONITOR Analyzer.

application. (1) Program that performs a specific set of tasks within a MAINVIEW product. (2) In MAINVIEW VistaPoint, combination of workloads to enable display of their transaction performance data in a single view.

application trace. *See* trace.

ASCH workload. Workload comprising Advanced Program-to-Program Communication (APPC) address spaces.

AutoCustomization. Online facility for customizing the installation of products. AutoCustomization provides an ISPF panel interface that both presents customization steps in sequence and provides current status information about the progress of the installation.

automatic screen update. Usage mode wherein the currently displayed screen is refreshed automatically with new data at an interval you specify. Invoked by the ASU command.

B

batch workload. Workload consisting of address spaces running batch jobs.

BBI. Basic architecture that distributes work between workstations and multiple OS/390 targets for BMC Software MAINVIEW products.

BBI-SS PAS. *See* BBI subsystem product address space.

BBI subsystem product address space (BBI-SS PAS). OS/390 subsystem address space that manages communication between local and remote systems and that contains one or more of the following products:

- Command MQ for S/390
- MAINVIEW AutoOPERATOR
- MAINVIEW for CICS
- MAINVIEW for DB2
- MAINVIEW for DBCTL
- MAINVIEW for IMS Online
- MAINVIEW for MQSeries
- MAINVIEW SRM
- MAINVIEW VistaPoint (for CICS, DB2, DBCTL, and IMS workloads)

BBPARM. *See* parameter library.

BBPROC. *See* procedure library.

BBPROF. *See* profile library.

BBSAMP. *See* sample library.

BBV. *See* MAINVIEW Alternate Access.

BBXS. BMC Software Subsystem Services. Common set of service routines loaded into common storage and used by several BMC Software MAINVIEW products.

border. Visual indication of the boundaries of a window.

bottleneck analysis. Process of determining which resources have insufficient capacity to provide acceptable service levels and that therefore can cause performance problems.

C

CA-Disk. Data management system by Computer Associates that replaced the DMS product.

CAS. Coordinating address space. One of the address spaces used by the MAINVIEW windows environment architecture. The CAS supplies common services and enables communication between linked systems. Each OS/390 or z/OS image requires a separate CAS. Cross-system communication is established through the CAS using VTAM and XCF communication links.

CFMON. *See* coupling facility monitoring.

chart. Display format for graphical data. *See also* graph.

CICSplex. User-defined set of one or more CICS systems that are controlled and managed as a single functional entity.

CMF MONITOR. Comprehensive Management Facility MONITOR. Product that measures and reports on all critical system resources, such as CPU, channel, and device usage; memory, paging, and swapping activity; and workload performance.

CMF MONITOR Analyzer. Batch component of CMF MONITOR that reads the SMF user and 70 series records created by the CMF MONITOR Extractor and/or the RMF Extractor and formats them into printed system performance reports.

CMF MONITOR Extractor. Component of CMF that collects performance statistics for CMF MONITOR Analyzer, CMF MONITOR Online, MAINVIEW for OS/390, and RMF postprocessor. *See* CMF MONITOR Analyzer, CMF MONITOR Online, MAINVIEW for OS/390.

CMF MONITOR Online. Component of CMF that uses the MAINVIEW window interface to present data on all address spaces, their use of various system resources, and the delays that each address space incurs while waiting for access to these resources. *See* CMF MONITOR, MAINVIEW for OS/390.

CMF Type 79 API. Application programming interface, provided by CMF, that provides access to MAINVIEW SMF-type 79 records.

CMFMON. Component of CMF MONITOR that simplifies online retrieval of information about system hardware and application performance and creates MAINVIEW SMF-type 79 records.

The CMFMON *online facility* can be used to view data in one or more formatted screens.

The CMFMON *write facility* can be used to write collected data as MAINVIEW SMF-type 79 records to an SMF or sequential data set.

CMRDETL. MAINVIEW for CICS data set that stores detail transaction records (type 6E) and abend records (type 6D). Detail records are logged for each successful transaction. Abend records are written when an abend occurs. Both records have the same format when stored on CMRDETL.

CMRSTATS. MAINVIEW for CICS data set that stores both CICS operational statistic records, at five-minute intervals, and other records, at intervals defined by parameters specified during customization (using CMRSOPT).

column. Vertical component of a view or display, typically containing fields of the same type of information, that varies by the objects associated in each row.

collection interval. Length of time data is collected. *See also* delta mode, total mode.

command delimiter. Special character, usually a **;** (semicolon), used to stack commands typed concurrently on the COMMAND line for sequential execution.

COMMAND line. Line in the control area of the display screen where primary commands can be typed. *Contrast with* line command column.

Command MQ Automation D/S. Command MQ agents, which provide local proactive monitoring for both MQSeries and MSMQ (Microsoft message queue manager). The Command MQ agents operate at the local node level where they continue to perform functions regardless of the availability of the MQM (message queue manager) network. Functionality includes automatic monitoring and restarts of channels, queue managers, queues and command servers. In cases where automated recovery is not possible, the agents transport critical alert information to a central console.

Command MQ Automation S/390. Command MQ component, which monitors the MQM (message queue manager) networks and intercedes to perform corrective actions when problems arise. Solutions include:

- Dead-Letter Queue management
- System Queue Archival
- Service Interval Performance solutions
- Channel Availability

These solutions help ensure immediate relief to some of the most pressing MQM operations and performance problems.

Command MQ for D/S. Command MQ for D/S utilizes a true client/server architecture and employs resident agents to provide configuration, administration, performance monitoring and operations management for the MQM (message queue manager) network.

Command MQ for S/390. *See* MAINVIEW for MQSeries.

COMMON STORAGE MONITOR. Component of MAINVIEW for OS/390 that monitors usage and reconfigures OS/390 or z/OS common storage blocks.

composite workload. Workload made up of a WLM workload or other workloads, which are called *constituent workloads.*

constituent workload. Member of a composite workload. Constituent workloads in a composite usually belong to a single workload class, but sometimes are mixed.

contention. Occurs when there are more requests for service than there are servers available.

context. In a Plex Manager view, field that contains the name of a target or group of targets specified with the CONTEXT command. *See* scope, service point, SSI context, target context.

CONTEXT command. Specifies either a MAINVIEW product and a specific target for that product (*see* target context) or a MAINVIEW product and a name representing one or more targets (see *SSI context*) for that product.

control statement. (1) Statement that interrupts a sequence of instructions and transfers control to another part of the program. (2) Statement that names samplers and other parameters that configure the MAINVIEW components to perform specified functions. (3) In CMF MONITOR, statement in a parameter library member used to identify a sampler in the extractor or a report in the analyzer, or to describe either component's processing requirements to the operating system.

coupling facility monitoring (CFMON). Coupling facility views that monitor the activity of your system's coupling facilities.

current data. Data that reflects the system in its current state. The two types of current data are realtime data and interval data. *Contrast with* historical data. *See also* interval data and realtime data.

current window. In the MAINVIEW window environment, window where the main dialog with the application takes place. The current window is used as the default window destination for commands issued on the COMMAND line when no window number is specified. *Contrast with* alternate window. *See* active window, window.

D

DASD. Direct Access Storage Device. (1) A device with rotating recording surfaces that provides immediate access to stored data. (2) Any device that responds to a DASD program.

data collector. Program that belongs to a MAINVIEW product and that collects data from various sources and stores the data in records used by views. For example, MAINVIEW for OS/390 data collectors obtain data from OS/390 or z/OS services, OS/390 or z/OS control blocks, CMF MONITOR Extractor control blocks, and other sources. *Contrast with* extractor.

delta mode. (1) In MAINVIEW for DB2 analyzer displays, difference between the value sampled at the start of the current statistics interval and the value sampled by the current analyzer request. See also *statistics interval.* (2) In CMFMON, usage mode wherein certain columns of data reflect the difference in values between one sample cycle and the next. Invoked by the DELta ON command. *See also* collection interval, sample cycle, total mode.

DFSMS. Data Facility Storage Management System. Data management, backup, and HSM software from IBM for OS/390 or z/OS mainframes.

DMR. *See* MAINVIEW for DB2.

DMS. Data Management System. *See* CA-Disk.

DMS2HSM. *See* MAINVIEW SRM DMS2HSM.

DSO. Data Set Optimizer. CMF MONITOR Extractor component that uses CMF MONITOR Extractor data to produce reports specifying the optimal ordering of data sets on moveable head devices.

E

EasyHSM. *See* MAINVIEW SRM EasyHSM.

EasyPOOL. *See* MAINVIEW SRM EasyPOOL.

EasySMS. *See* MAINVIEW SRM EasySMS.

element. (1) Data component of a data collector record, shown in a view as a field. (2) Internal value of a field in a view, used in product functions.

element help. Online help for a field in a view. The preferred term is *field help.*

Enterprise Storage Automation. *See* MAINVIEW SRM Enterprise Storage Automation.

event. A message issued by Enterprise Storage Automation. User-defined storage occurrences generate events in the form of messages. These events provide an early warning system for storage problems and are routed to user-specified destinations for central viewing and management.

Event Collector. Component for MAINVIEW for IMS Online, MAINVIEW for IMS Offline, and MAINVIEW for DBCTL that collects data about events in the IMS environment. This data is required for Workload Monitor and optional for Workload Analyzer (except for the workload trace service). This data also is recorded as transaction records (X'FA') and program records (X'F9') on the IMS system log for later use by the MAINVIEW for IMS Offline components: Performance Reporter and Transaction Accountant.

expand. Predefined link from one display to a related display. *See also* hyperlink.

extractor. Program that collects data from various sources and keeps the data control blocks to be written as records. Extractors obtain data from services, control blocks, and other sources. *Contrast with* data collector.

extractor interval. *See* collection interval.

F

fast path. Predefined link between one screen and another. To use the fast path, place the cursor on a single value in a field and press Enter. The resulting screen displays more detailed information about the selected value. *See also* hyperlink.

field. Group of character positions within a screen or report used to type or display specific information.

field help. Online help describing the purpose or contents of a field on a screen. To display field help, place the cursor anywhere in a field and press PF1 (HELP). In some products, field help is accessible from the screen help that is displayed when you press PF1.

filter. Selection criteria used to limit the number of rows displayed in a view. Data that does not meet the selection criteria is not displayed. A filter is composed of an element, an operator, and an operand (a number or character string). Filters can be implemented in view customization, through the PARm/QPARm commands, or through the Where/QWhere commands. Filters are established against elements of data.

fire. The term used to indicate that an event has triggered an action. In MAINVIEW AutoOPERATOR, when a rule selection criteria matches an incoming event and *fires*, the user-specified automation actions are performed. This process is also called *handling* the event.

fixed field. Field that remains stationary at the left margin of a screen that is scrolled either right or left.

FOCAL POINT. MAINVIEW product that displays a summary of key performance indicators across systems, sites, and applications from a single terminal.

form. One of two constituent parts of a view; the other is query. A form defines how the data is presented; a query identifies the data required for the view. See also *query, view.*

full-screen mode. Display of a MAINVIEW product application or service on the entire screen. There is no window information line. *Contrast with* windows mode.

G

global command. Any MAINVIEW window interface command that can affect all windows in the window area of a MAINVIEW display.

graph. Graphical display of data that you select from a MAINVIEW window environment view. *See also* chart.

H

hilevel. For MAINVIEW products, high-level data set qualifier required by a site's naming conventions.

historical data. (1) Data that reflects the system as it existed at the end of a past recording interval or the duration of several intervals. (2) Any data stored in the historical database and retrieved using the TIME command. *Contrast with* current data, interval data and realtime data.

historical database. Collection of performance data written at the end of each installation-defined recording interval and containing up to 100 VSAM clusters. Data is extracted from the historical database with the TIME command. *See* historical data.

historical data set. In MAINVIEW products that display historical data, VSAM cluster file in which data is recorded at regular intervals.

HSM. (Hierarchical Storage Management) Automatic movement of files from hard disk to slower, less-expensive storage media. The typical hierarchy is from magnetic disk to optical disk to tape.

hyperlink. (1) Preset field in a view or an EXPAND line on a display that permits you to

- Access cursor-sensitive help
- Issue commands
- Link to another view or display

The transfer can be either within a single product or to a related display/view in a different BMC Software product. Generally, hyperlinked fields are highlighted. (2) Cursor-activated short path from a topic or term in online help to related information. *See also* fast path.

I

Image log. Collection of screen-display records. Image logs can be created for both the BBI-SS PAS and the BBI terminal session (TS).

The BBI-SS PAS Image log consists of two data sets that are used alternately: as one fills up, the other is used. Logging to the BBI-SS PAS Image log stops when both data sets are filled and the first data set is not processed by the archive program.

The TS Image log is a single data set that wraps around when full.

IMSPlex System Manager (IPSM). MVIMS Online and MVDBC service that provides Single System Image views of resources and bottlenecks for applications across one or more IMS regions and systems.

interval data. Cumulative data collected during a collection interval. Intervals usually last from 15 to 30 minutes depending on how the recording interval is specified during product customization. *Contrast with* historical data.

Note: If change is made to the workloads, a new interval will be started.

See also current data and realtime data.

InTune. Product for improving application program performance. It monitors the program and provides information used to reduce bottlenecks and delays.

IRUF. IMS Resource Utilization File (IRUF). IRUFs can be either detail (one event, one record) or summarized (more than one event, one record). A detail IRUF is created by processing the IMS system log through a program called IMFLEDIT. A summarized IRUF is created by processing one or more detail IRUFs, one or more summarized IRUFs, or a combination of both, through a sort program and the TASCOSTR program.

J

job activity view. Report about address space consumption of resources. *See* view.

journal. Special-purpose data set that stores the chronological records of operator and system actions.

Journal log. Collection of messages. Journal logs are created for both the BBI-SS PAS and the BBI terminal session (TS).

The BBI-SS PAS Journal log consists of two data sets that are used alternately: as one fills up, the other is used. Logging to the BBI-SS PAS Journal log stops when both data sets are filled and the first data set is not being processed by the archive program.

The TS Journal log is a single data set that wraps around when full.

L

line command. Command that you type in the line command column in a view or display. Line commands initiate actions that apply to the data displayed in that particular row.

line command column. Command input column on the left side of a view or display. *Contrast with* COMMAND line.

Log Edit. In the MAINVIEW for IMS Offline program named IMFLEDIT, function that extracts transaction (X‘FA’) and program (X‘F9’) records from the IMS system log. IMFLEDIT also extracts certain records that were recorded on the system log by IMS. IMFLEDIT then formats the records into a file called the IMS Resource Utilization File (IRUF).

M

MAINVIEW. BMC Software integrated systems management architecture.

MAINVIEW Alarm Manager. In conjunction with other MAINVIEW products, notifies you when an exception condition occurs. MAINVIEW Alarm Manager is capable of monitoring multiple systems simultaneously, which means that MAINVIEW Alarm Manager installed on one system keeps track of your entire sysplex. You can then display a single view that show exceptions for all MAINVIEW performance monitors within your OS/390 or z/OS enterprise.

MAINVIEW Alternate Access. Enables MAINVIEW products to be used without TSO by providing access through EXCP and VTAM interfaces.

MAINVIEW Application Program Interface. REXX- or CLIST-based, callable interface that allows MAINVIEW AutoOPERATOR EXECs to access MAINVIEW monitor product view data.

MAINVIEW AutoOPERATOR. Product that uses tools, techniques, and facilities to automate routine operator tasks and provide online performance monitoring, and that achieves high availability through error minimization, improved productivity, and problem prediction and prevention.

MAINVIEW control area. In the MAINVIEW window environment, first three lines at the top of the view containing the window information line and the COMMAND, SCROLL, CURR WIN, and ALT WIN lines. The control area cannot be customized and is part of the information display. *Contrast with* MAINVIEW display area, MAINVIEW window area.

MAINVIEW display area. *See* MAINVIEW window area.

MAINVIEW Explorer. Product that provides access to MAINVIEW products from a Web browser running under Windows. MAINVIEW Explorer replaces MAINVIEW Desktop.

MAINVIEW for CICS. Product (formerly MV MANAGER for CICS) that provides realtime application performance analysis and monitoring for CICS system management.

MAINVIEW for DB2. Product (formerly MV MANAGER for DB2) that provides realtime and historical application performance analysis and monitoring for DB2 subsystem management.

MAINVIEW for DBCTL. Product (formerly MV MANAGER for DBCTL) that provides realtime application performance analysis and monitoring for DBCTL management.

MAINVIEW for IMS (MVIMS) Offline. Product with a Performance Reporter component that organizes data and prints reports used to analyze IMS performance and a Transaction Accountant component that produces cost accounting and user charge-back records and reports.

MAINVIEW for IMS (MVIMS) Online. Product that provides realtime application performance analysis and monitoring for IMS management.

MAINVIEW for IP. Product that monitors OS/390 and z/OS mission-critical application performance as it relates to TCP/IP stack usage. Collected data includes availability, connections, response times, routers, service levels, storage, traffic, Web cache, and so on.

MAINVIEW for Linux–Servers. Product that allows you to monitor the performance of your Linux systems from the MAINVIEW windows interface.

MAINVIEW for MQSeries. Delivers comprehensive capabilities for configuration, administration, performance monitoring and operations management for an entire MQM (message queue manager) network.

MAINVIEW for OS/390. System management application (known as MAINVIEW **for MVS** prior to version 2.5). Built upon the MAINVIEW window environment architecture, it uses the window interface to provide access to system performance data and other functions necessary in the overall management of an enterprise.

MAINVIEW for UNIX System Services. System management application that allows you to monitor the performance of the Unix System Services from a MAINVIEW window interface.

MAINVIEW for VTAM. Product that displays application performance data by application, transaction ID, and LU name. This collected data includes: connections, response time statistics, application availability, and application throughput.

MAINVIEW for WebSphere. Product that provides Web monitoring and management for applications integrated with IBM WebSphere Application Server for OS/390 or z/OS.

MAINVIEW Selection Menu. ISPF selection panel that provides access to all MAINVIEW windows-mode and full-screen mode products.

MAINVIEW SRM. *See* MAINVIEW Storage Resource Manager (SRM).

MAINVIEW SRM DMS2HSM. Product that facilitates the conversion of CA-Disk, formerly known as DMS, to HSM.

MAINVIEW SRM EasyHSM. Product that provides online monitoring and reporting to help storage managers use DFHSM efficiently.

MAINVIEW SRM EasyPOOL. Product that provides control over data set allocation and enforcement of allocation and naming standards. EasyPOOL functions operate at the operating system level to intercept normal job processing, thus providing services without any JCL changes.

MAINVIEW SRM EasySMS. Product that provides tools that aid in the conversion to DFSMS and provides enhancement to the DFSMS environment after implementation. EasySMS consists of the EasyACS functions, the SMSACSTE function, and the Monitoring and Positioning Facility.

MAINVIEW SRM Enterprise Storage Automation. Product that delivers powerful event generation and storage automation technology across the storage enterprise. Used in conjunction with MAINVIEW AutoOPERATOR, automated solutions to perform pool, volume, application, or data set-level manipulation can be created and used in response to any condition or invoked to perform ad hoc requests

MAINVIEW SRM SG-Auto. Product that provides early warning notification of storage anomalies and automated responses to those anomalies based on conditions in the storage subsystem.

MAINVIEW SRM SG-Control. Product that provides real-time monitoring, budgeting, and control of DASD space utilization.

MAINVIEW SRM StopX37/II. Product that provides enhancements to OS/390 or z/OS space management, reducing the incidence of space-related processing problems. The StopX37/II functions operate at the system level to intercept abend conditions or standards violations, thus providing services without any JCL changes.

MAINVIEW SRM StorageGUARD. Product that monitors and reports on DASD consumption and provides historical views to help control current and future DASD usage.

MAINVIEW Storage Resource Manager (SRM). Suite of products that assists in all phases of OS/390 or z/OS storage management. MAINVIEW SRM consists of products that perform automation, reporting, trend analysis, and error correction for storage management.

MAINVIEW SYSPROG Services. *See* SYSPROG Services.

MAINVIEW VistaPoint. Product that provides enterprise-wide views of performance. Application and workload views are available for CICS, DB2, DBCTL, IMS, and OS/390. Data is summarized at the level of detail needed; for example, views can be for a single target, an OS/390 or z/OS image, or an entire enterprise.

MAINVIEW window area. Portion of the information display that is not the control area and in which views are displayed and windows opened. It includes all but the first three lines of the information display. *Contrast with* MAINVIEW control area.

monitor. Online service that measures resources or workloads at user-defined intervals and issues warnings when user-defined thresholds are exceeded.

Multi-Level Automation (MLA). The user-defined, multiple step process in Enterprise Storage Automation that implements solutions in a tiered approach, where solutions are invoked one after another until the condition is resolved.

MVALARM. *See* MAINVIEW Alarm Manager.

MVAPI. *See* MAINVIEW Application Program Interface.

MVCICS. *See* MAINVIEW for CICS.

MVDB2. *See* MAINVIEW for DB2.

MVDBC. *See* MAINVIEW for DBCTL.

MVIMS. *See* MAINVIEW for IMS.

MVIP. *See* MAINVIEW for IP.

MVLNX. *See* MAINVIEW for Linux–Servers.

MVMQ. *See* MAINVIEW for MQSeries.

MVMVS. *See* MAINVIEW for OS/390.

MVScope. MAINVIEW for OS/390 application that traces both CPU usage down to the CSECT level and I/O usage down to the channel program level.

MVSRM. *See* MAINVIEW Storage Resource Manager (SRM).

MVSRMHSM. *See* MAINVIEW SRM EasyHSM.

MVSRMSGC. *See* MAINVIEW SRM SG-Control.

MVSRMSGD. *See* MAINVIEW SRM StorageGUARD.

MVSRMSGP. *See* MAINVIEW SRM StorageGUARD.

MVVP. *See* MAINVIEW VistaPoint.

MVVTAM. *See* MAINVIEW for VTAM.

MVWEB. *See* MAINVIEW for WebSphere.

N

nested help. Multiple layers of help pop-up windows. Each successive layer is accessed by clicking a hyperlink from the previous layer.

O

object. Anything you can manipulate as a single unit. MAINVIEW objects can be any of the following: product, secondary window, view, row, column, or field.

You can issue an action against an object by issuing a line command in the line command column to the left of the object. *See* action.

OMVS workload. Workload consisting of OS/390 OpenEdition address spaces.

online help. Help information that is accessible online.

OS/390 and z/OS Installer. BMC Software common installation system for mainframe products.

OS/390 product address space (PAS). Address space containing OS/390 or z/OS data collectors, including the CMF MONITOR Extractor. Used by the MAINVIEW for OS/390, MAINVIEW for Unix System Services, and CMF MONITOR products. *See* PAS.

P

parameter library. Data set consisting of members that contain parameters for specific MAINVIEW products or a support component. There can be several versions:

- The distributed parameter library, called BBPARM
- A site-specific parameter library or libraries

  These can be

  – A library created by AutoCustomization, called UBBPARM

  – A library created manually, with a unique name

PAS. Product address space. Used by the MAINVIEW products. Contains data collectors and other product functions. *See* OS/390 product address space (PAS), BBI subsystem product address space (BBI-SS PAS).

performance group workload. Collection of address spaced defined to OS/390 or z/OS. If you are running OS/390 or z/OS with WLM in compatibility mode, MAINVIEW for OS/390 creates a performance group workload instead of a service class. *See* service class workload, workload definition.

PERFORMANCE MANAGER. MAINVIEW for CICS online service for monitoring and managing current performance of CICS regions.

Performance Reporter (MVIMS Offline). MVIMS Offline component that organizes data and prints reports that can be used to analyze IMS performance.

Performance Reporter. Product component that generates offline batch reports. The following products can generate these reports:

- MAINVIEW for DB2
- MAINVIEW for CICS

Plex Manager. Product through which cross-system communication, MAINVIEW security, and an SSI context are established and controlled. Plex Manager is shipped with MAINVIEW window environment products as part of the coordinating address space (CAS) and is accessible as a menu option from the MAINVIEW Selection Menu.

PRGP workload. In MVS/SP 5.0 or earlier, or in compatibility mode in MVS/SP 5.1 or later, composite of service classes. MAINVIEW for OS/390 creates a performance group workload for each performance group defined in the current IEAIPS*xx* member.

procedure library. Data set consisting of members that contain executable procedures used by MAINVIEW AutoOPERATOR. These procedures are execute command lists (EXECs) that automate site functions. There can be several versions:

- The distributed parameter library, called BBPROC
- A site-specific parameter library or libraries

  These can be

  – A library created by AutoCustomization, called UBBPROC

  – A library created manually, with a unique name

The site-created EXECs can be either user-written or customized MAINVIEW AutoOPERATOR-supplied EXECs from BBPROC.

product address space. *See* PAS.

profile library. Data set consisting of members that contain profile information and cycle refresh definitions for a terminal session connected to a BBI-SS PAS. Other members are dynamically created by MAINVIEW applications. There can be several versions:

- The distributed profile library, called BBPROF
- A site-specific profile library or libraries

  These can be

  – A library created by AutoCustomization, called SBBPROF

  – A library created manually, with a unique name

The site library is a common profile shared by all site users. The terminal session CLIST creates a user profile automatically if one does not exist; it is called userid.BBPROF, where userid is your logon ID. User profile libraries allow each user to specify unique PF keys, CYCLE commands, target system defaults, a Primary Option Menu, and a unique set of application profiles.

Q

query. One of two constituent parts of a view; the other is form. A query defines the data for a view; a form defines the display format. *See also* form, view.

R

realtime data. Performance data as it exists at the moment of inquiry. Realtime data is recorded during the smallest unit of time for data collection. *Contrast with* historical data. *See also* current data and interval data.

Resource Analyzer. Online realtime displays used to analyze IMS resources and determine which are affected by specific workload problems.

Resource Monitor. Online data collection services used to monitor IMS resources and issue warnings when defined utilization thresholds are exceeded.

row. (1) Horizontal component of a view or display comprising all the fields pertaining to a single device, address space, user, etc. (2) Horizontal component of a DB2 table consisting of a sequence of values, one for each column of the table.

RxD2. Product that provides access to DB2 from REXX. It provides tools to query the DB2 catalog, issue dynamic SQL, test DB2 applications, analyze EXPLAIN data, generate DDL or DB2 utility JCL, edit DB2 table spaces, perform security administration, and much more.

S

sample cycle. Time between data samples.

For the CMF MONITOR Extractor, this is the time specified in the extractor control statements (usually 1 to 5 seconds).

For realtime data, the cycle is not fixed. Data is sampled each time you press Enter.

sample library. Data set consisting of members each of which contains one of the following:

- Sample JCL that can be edited to perform specific functions
- A macro that is referenced in the assembly of user-written services
- A sample user exit routine

There can be several versions:

- The distributed sample library, called BBSAMP
- A site-specific sample library or libraries

  These can be

  – A library created by AutoCustomization, called UBBSAMP

  – A library created manually, with a unique name

sampler. Program that monitors a specific aspect of system performance. Includes utilization thresholds used by the Exception Monitor. The CMF MONITOR Extractor contains samplers.

SBBPROF. *See* profile library.

scope. Subset of an SSI context. The scope could be all the data for the context or a subset of data within the context. It is user- or site-defined. *See* SSI context, target.

screen definition. Configuration of one or more views that have been stored with the SAVEScr command and assigned a unique name. A screen includes the layout of the windows and the view, context, system, and product active in each window.

selection view. In MAINVIEW products, view displaying a list of available views.

service class workload. Collection of address spaces defined to OS/390 or z/OS. If you are running Workload Manager (WLM) in goal mode, MAINVIEW for OS/390 creates a service class workload for each service class that you define through WLM definition dialogs.

If you are running MVS 4.3 or earlier, or MVS/SP 5.1 or later with WLM in compatibility mode, MVS creates a performance group workload instead of a service class. *See* performance group workload.

service objective. Workload performance goal, specified in terms of response time for TSO workloads or turnaround time for batch workloads. Performance group workloads can be measured by either objective. Composite workload service objectives consist of user-defined weighting factors assigned to each constituent workload. For compatibility mode, neither OS/390 nor z/OS provides any way to measure service.

service point. Specification, to MAINVIEW, of the services required to enable a specific product. Services can be actions, selectors, or views. Each target (for example, CICS, DB2, or IMS) has its own service point.

The PLEX view lists all the defined service points known to the CAS to which the terminal session is connected.

service request block (SRB). Control block that represents a routine to be dispatched. SRB mode routines generally perform work for the operating system at a high priority. An SRB is similar to a task control block (TCB) in that it identifies a unit of work to the system. *See also* task control block.

service select code. Code entered to invoke analyzers, monitors, and general services. This code is also the name of the individual service.

session. Total period of time an address space has been active. A session begins when monitoring can be performed. If the product address space (PAS) starts after the job, the session starts with the PAS.

SG-Auto. *See* MAINVIEW SRM SG-Auto.

SG-Control. *See* MAINVIEW SRM SG-Control.

single system image (SSI). Feature of the MAINVIEW window environment architecture where you can view and perform actions on multiple OS/390 systems as though they were a single system. The rows of a single tabular view can contain rows from different OS/390 or z/OS images.

Skeleton Tailoring Facility. A facility in MAINVIEW AutoOPERATOR that allows skeleton JCL to be used during job submission. Skeleton JCL can contain variables within the JCL statements to be substituted with data values at job submission time. Directive statements can be used in the skeleton JCL to cause the repetition of a set of skeleton statements. This facility functions similar to the TSO skeleton tailoring facility.

SRB. *See* service request block.

SSI. *See* single system image.

SSI context. Name created to represent one or more targets for a given product. *See* context, target.

started task workload. Address spaces running jobs that were initiated programmatically.

statistics interval. For MAINVIEW for DB2, cumulative count within a predefined interval (30-minute default set by the DB2STATS parameter in the distributed BBPARM member BBIISP00) for an analyzer service DELTA or RATE display. Specifying the DELTA parameter displays the current value as the difference between the value sampled by the current analyzer request and the value sampled at the start of the current interval. Specifying the RATE parameter displays the current value by minute (DELTA divided by the number of elapsed minutes).

stem variables. A REXX facility, supported in MAINVIEW AutoOPERATOR REXX EXECs and the Skeleton Tailoring Facility, where variable names end with a period followed by a number, such as &POOL.1. This configuration allows each variable to actually represent a table or array of data, with the zero variable containing the number of entries in the array. For example, &POOL.0 = 5 would indicate variables &POOL.1 through &POOL.5 exist.

StopX37/II. *See* MAINVIEW SRM StopX37/II.

StorageGUARD. *See* MAINVIEW SRM StorageGUARD.

summary view. View created from a tabular view using the Summarize option in view customization. A summary view compresses several rows of data into a single row based on the summarize criteria.

SYSPROG services. Component of MAINVIEW for OS/390. Over 100 services that detect, diagnose, and correct OS/390 or z/OS system problems as they occur. Accessible from the OS/390 Performance and Control Main Menu. Note that this component is also available as a stand-alone product MAINVIEW SYSPROG Services.

system resource. *See* object.

T

target. Entity monitored by one or more MAINVIEW products, such as an OS/390 or z/OS image, an IMS or DB2 subsystem, a CICS region, or related workloads across systems. *See* context, scope, SSI context.

target context. Single target/product combination. *See* context.

TASCOSTR. MAINVIEW for IMS Offline program that summarizes detail and summary IMS Resource Utilization Files (IRUFs) to be used as input to the offline components.

task control block (TCB). Address space-specific control block that represents a unit of work that is dispatched in the address space in which it was created. *See also* service request block.

TCB. *See* task control block.

terminal session (TS). Single point of control for MAINVIEW products, allowing data manipulation and data display and providing other terminal user services for MAINVIEW products. The terminal session runs in a user address space (either a TSO address space or a standalone address space for EXCP/VTAM access).

TDIR. *See* trace log directory.

threshold. Specified value used to determine whether the data in a field meets specific criteria.

TLDS. *See* trace log data set.

total mode. Usage mode in CMFMON wherein certain columns of data reflect the cumulative value between collection intervals. Invoked by the DELta OFF command. *See also* collection interval, delta mode.

trace. (1) Record of a series of events chronologically listed as they occur. (2) Online data collection and display services that track transaction activity through DB2, IMS, or CICS.

trace log data set (TLDS). Single or multiple external VSAM data sets containing summary or detail trace data for later viewing or printing. The trace log(s) can be defined as needed or dynamically allocated by the BBI-SS PAS. Each trace request is assigned its own trace log data set(s).

trace log directory (TDIR). VSAM linear data set containing one entry for each trace log data set. Each entry indicates the date and time of data set creation, the current status of the data set, the trace target, and other related information.

transaction. Specific set of input data that initiates a predefined process or job.

Transaction Accountant. MVIMS Offline component that produces cost accounting and user charge-back records and reports.

TS. *See* terminal session.

TSO workload. Workload that consists of address spaces running TSO sessions.

U

UAS. *See* user address space.

UBBPARM. *See* parameter library.

UBBPROC. *See* procedure library.

UBBSAMP. *See* sample library.

user address space. Runs a MAINVIEW terminal session (TS) in TSO, VTAM, or EXCP mode.

User BBPROF. *See* profile library.

V

view. Formatted data within a MAINVIEW window, acquired from a product as a result of a view command or action. A view consists of two parts: query and form. *See also* form, job activity view, query.

view definition. Meaning of data that appears online, including source of data, selection criteria for data field inclusion and placement, data format, summarization, context, product, view name, hyperlink fields, and threshold conditions.

view command. Name of a view that you type on the COMMAND line to display that view.

view command stack. Internal stack of up to 10 queries. For each command, the stack contains the filter parameters, sort order, context, product, and timeframe that accompany the view.

view help. Online help describing the purpose of a view. To display view help, place the cursor on the view name on the window information line and press PF1 (HELP).

W

window. Area of the MAINVIEW screen in which views and resources are presented. A window has visible boundaries and can be smaller than or equal in size to the MAINVIEW window area. *See* active window, alternate window, current window, MAINVIEW window area.

window information line. Top border of a window. Shows the window identifier, the name of the view displayed in the window, the system, the scope, the product reflected by the window, and the timeframe for which the data in the window is relevant. *See also* window status field.

window number. Sequential number assigned by MAINVIEW to each window when it is opened. The window number is the second character in the window status field. *See also* window status field.

window status. One-character letter in the window status field that indicates when a window is ready to receive commands, is busy processing commands, is not to be updated, or contains no data. It also indicates when an error has occurred in a window. The window status is the first character in the window status field. *See also* window information line, window status field.

window status field. Field on the window information line that shows the current status and assigned number of the window. *See also* window number, window status.

windows mode. Display of one or more MAINVIEW product views on a screen that can be divided into a maximum of 20 windows. A window information line defines the top border of each window. *Contrast with* full-screen mode.

WLM workload. In goal mode in MVS/SP 5.1 and later, a composite of service classes. MAINVIEW for OS/390 creates a workload for each WLM workload defined in the active service policy.

workflow. Measure of system activity that indicates how efficiently system resources are serving the jobs in a workload.

workload. (1) Systematic grouping of units of work (e.g., address spaces, CICS transactions, IMS transactions) according to classification criteria established by a system administrator. (2) In OS/390 or z/OS, a group of service classes within a service definition.

workload activity view. Tracks workload activity as the workload accesses system resources. A workload activity view measures workload activity in terms of resource consumption and how well the workload activity meets its service objectives.

Workload Analyzer. Online data collection and display services used to analyze IMS workloads and determine problem causes.

workload definition. Workload created through the WKLIST view. Contains a unique name, a description, an initial status, a current status, and selection criteria by which address spaces are selected for inclusion in the workload. *See* Workload Definition Facility.

Workload Definition Facility. In MAINVIEW for OS/390, WKLIST view and its associated dialogs through which workloads are defined and service objectives set.

workload delay view. Tracks workload performance as the workload accesses system resources. A workload delay view measures any delay a workload experiences as it contends for those resources.

Workload Monitor. Online data collection services used to monitor IMS workloads and issue warnings when defined thresholds are exceeded.

workload objectives. Performance goals for a workload, defined in WKLIST. Objectives can include measures of performance such as response times and batch turnaround times.

Index

Symbols

$CSAABV
    Sample Monitoring Solution 503
$CSABLO
    Sample Monitoring Solution 503
$JOBCPU
    Sample Monitoring Solution 503
$STCCPU
    Sample Monitoring Solution 503
$TOTCPU
    Sample Monitoring Solution 503
$TOTSIO
    Sample Monitoring Solution 503
$TOTSIO2
    Sample Monitoring Solution 503
$TSOCPU
    Sample Monitoring Solution 503
$WATCH
    Sample Monitoring Solution 503
/* DOC statement
    EXEC Management application 468

A

A1 primary command
    Rule Processor application
        Rule Processor Detail Control panel 65, 73
AA primary command
    Rule Processor application
        Rule Processor Detail Control panel 65, 73
AAOARPxx
    Automation Reporter
        activating 424
        Dynamic Parameter Manager 416
AAORUL00 493
ABEND command (CICS) 471
ABEND field
    EXEC Management application 464
Acct Info field
    CMD-Initiated Rule Selection panel 552
    EXT-Initiated Rule Selection panel 563
    IMS-Initiated Rule Selection panel 568
    JES3-Initiated Rule Selection panel 573
    MSG-Initiated Rule Selection panel 584
    Rule Processor Selection Criteria panel
        CMD-Initiated Rule 76
        EXT-Initiated Rule 76
        IMS-Initiated Rule 76
        JES3-Initiated Rule 76
        MSG-Initiated Rule 76
Action field
    Rule Processor Detail Control panel 68
action specification
    Rule Processor application 35, 85
Activated at field
    Rule Processor application
        Event Activity Statistics application 51
activating
    Automation Reporter
        Dynamic Parameter Manager 416, 417
ACTN
    Automation Reporter
        data record 427
        example 437
        interval data 430
ADD primary command
    Rule Processor application
        Automation Control Panel 179
        naming conventions 179
        Rule Set Overview panel 203
After command
    Rule Processor application
        Automation Control panel 185
        Rule Set Overview panel 204
AID command (CICS) 471
Alarm field
    ALERT Detail Information display 412
    Rule Processor application
        ALERT Action(s) Specification panel 99
Alarm ID field
    ALRM-Initiated Rule Selection panel 536
    Rule Processor Selection Criteria panel
        ALRM-Initiated Rule 76
Alarm Text field
    ALRM-Initiated Rule Selection panel 536
ALERT Action(s) Specification
    Rules
        fields 97
        sample panel 94, 95
ALERT DETAIL application
    command associated with an ALERT 405
    DETAIL command 407
    displaying from ALERT Overview application 404
    EXECs and ALERTs 405
    EXPAND command 407
    fields 404–407
    GO command 407
    help panel for an ALERT 405
    PROFILE command 406
    sample panel 404
    SET command 407
    TRANSFER command 406
ALERT Detail Information display
    individual viewing 411
    sample panel 411
ALERT DETAIL PROFILE application
    sample panel 408
ALERT Management Facility
    application overview 393
    Clearing ALERT 400

color terminal 400
Critical ALERT 400
defining ALERTs 393
Informational ALERT 400
introduction 391
Major ALERT 400
management by exception 392
Minor ALERT 400
priorities 400
queues 399
Warning ALERT 400
ALERT Overview application
EXPAND command 401
fields 401
GO command 401
PROFILE command 401
Profile Suffix field 402–403
sample panel 399
SET command 401
TRANSFER command 401
ALERT Overview PROFILE application
fields 402
sample panel 402
ALERT queue
Automation Reporter
specifying 423
ALERT Queue Statistics application
fields 397
line commands 397
sample panel 396
sorting 398
ALERTs
Automation Reporter
queue 416
ALERTS command 14, 471
ALERTS ISSUED field
documentation box 497
ALIAS command (CICS) 472
ALL
Rule Processor application
Automation strategy 186
ALRM-Initiated Rule
selecting criteria and actions 536–540
ALRT
Automation Reporter
data record 427
example 443
interval data 430
event type 43
ALRT Variables
AMFKEY 174
ALRT-Initiated Rule
selecting criteria and actions 541–545
AMFALARM 101, 174
AMFCOLOR 101, 174
AMFEDIR 101, 102, 174
AMFEDISP 101, 102, 174
AMFEEXEC 102, 174
AMFEINT 101, 102, 174
AMFEXEC 101, 102, 174
AMFHELP 101, 102, 174
AMFIDATE 101, 102, 174
AMFITIME 101, 102, 174
AMFORGN 102, 175
AMFORGN, AMFPCMD 101, 174
AMFPCMD 102, 175
AMFPRIOR 101, 102, 174, 175
AMFPSYS 101, 102, 174, 175
AMFPUB 101, 102, 174, 175
AMFQUEUE 101, 102, 174, 175
AMFRTAIN 101, 102, 174, 175
AMFSSID 101, 102, 174, 175
AMFTEXT 101, 174
AMFTGT 101, 102, 174, 175
AMFUDATA 101, 102, 174, 175
AMFUSER 101, 102, 174, 175
AN command 15, 17
AND/OR field
Rule Processor application
Variable Dependencies panel 84
AND/OR logic
Rule Processor application
Variable Dependencies panel 81–84
AO
Product Line Transfer name 10
AOCYCnn
BBPROF predefined refresh cycle member 477
APPL command 19
application functions
common in AutoOPERATOR 9
Application Transfer 12–20
AR command 17
AT command 15, 17
ATTN key
Refresh Mode 23
SERVICE REFRESH CYCLE 479
Author field
documentation box 496
EXEC Management application 463, 468
Rule Processor Detail Control panel 69
Rule Set Overview panel 197, 199
Auto Delete
ALERT Action(s) panel field
description 96
Automation Control application
Rule Processor application 40, 41
Automation Control panel
Rule Processor application
Automation Status 178
Automation Strategy 178
fields 181, 182, 184
grouping Rules 183–184
line commands 185–194
primary commands 179
sample panel 177
Automation Reporter
activating
AAOARPxx 416

Dynamic Parameter Manager 416
data records
ACTN 427, 435, 437
ALRT 427, 435, 443
cumulative 429
describing 427, 431, 435
EVENT 435, 445
EVNT 428
EXEC 428, 435, 446
interval 429
lengths 433
RES 428, 436, 451
RULE 428, 435, 447
RULESET (AUTO) 428, 436, 448
RULESET (SET) 428, 436, 449
saving 429
TAPE 453, 455
TAPE (LOCAL) 429, 436
TAPE (PLEX) 429, 436
introduction 415
REXX/370 Alternate library 415
specifying
ALERT queue 423
ALERT queue name 416
collection interval 421
data set 416, 419
data set disposition 420
offload interval 416, 422
time intervals 416
stopping 426
Automation Statistics
Rules 182
Automation Status field
Rule Processor application
Automation Control panel 181
automation strategy
Rule Processor application 39, 41
Rules 186–194
Automation Strategy field
Rule Processor application
Automation Control panel 181
AutoOPERATOR
ALERT Management Facility 391–413
application requesting 9
Application Transfer 12–20
common application functions 9
common display fields 9
EXEC invocation 25
EXEC Management application 457–468
features 4
general services 469–489
introduction 3–8
introduction to using Sample Solutions 493–497
jump function 9
overview figure 3
Primary Option Menu 6
Primary Option Menu retrieval 21
Product Line Transfer request 10–11
requesting an AutoOPERATOR application 9
Rule Processor application 31
Rules 206
Sample CICSPlex Solution 505
Sample Info/System Management Solution 507–508
Sample Monitoring Solution 501–503
Sample VTAM/NCP Solution 499–500
Avg. Size field
ALERT Queue Statistics application 397
AVGCPU field
EXEC Management application 465

B

BBI control commands 24
BBI services
See general services
BBI Terminal Session screen
splitting 24
BBIISP00
TIMEXEC application 487, 488
BBI-SS PAS Journal log messages 25
BBITSP00
CYCLE SETUP application 470
BBPARM data set 493
See also each member name
BBPARM member BBIISP00
TIMEXEC application 487, 488
BBPROC data set 493
See also each member name
BBPROF data set
CYCLE SETUP application 470, 477
BBPROF predefined refresh cycle member
CYCLE SETUP application 477–479
BBUSER data set 507
Before command
Rule Processor application
Automation Control panel 185
Rule Set Overview panel 204
BFRPL command (DB2) 473
BLK parameter
TIMEXEC application 487, 488, 489
bouncing objects
CSM 316
Broadcast command 14
Browse command
EXEC Management application 463

C

C E 2109 command
BBI control 24
CALLX request
TIMEXEC application
block 488–489
purging active 489
setting up 486
single 487
viewing 489
CANCEL command

BBI control 24
Cancel line command
EXEC Management application 466
canceling objects
CSM 321
canceling SERVICE REFRESH CYCLE
general services 479
CAO
Product Line Transfer name 10
CAOCYCnn member name
BBPROF predefined refresh cycle member 477
CDRM command 19
CDRSC command 19
CHANGE LOG field
documentation box 497
Changed field
EXEC Management application 465
Rule Set Overview panel 198
CICS 217
event type 44
Product Line Transfer name 10
CICS TDQ field
CICS-Initiated Rule Selection panel 547
Rule Processor Selection Criteria panel
CICS-Initiated Rule 76
CICSC command (DB2) 473
CICSE command (DB2) 474
CICS-Initiated Rule
selecting criteria and actions 546–550
CICSR command (DB2) 474
CLASSES command (CICS) 472
Clearing 400
client-server relationship
CSM 217
Clrng field
ALERT Queue Statistics application 397
CLSTR command 19
CMD
event type 44
Cmd field
ALRM-Initiated Rule Action Specification panel 539
ALRT-Initiated Rule Action Specification panel 544
CICS-Initiated Rule Action Specification panel 549
CMD-Initiated Rule Action Specification panel 554
DB2-Initiated Rule Action Specification panel 559, 560
EXT-Initiated Rule Action Specification panel 565
IMS-Initiated Rule Action Specification panel 570
JES3-Initiated Rule Action Specification panel 575, 576
JRNL-Initiated Rule Action Specification panel 580, 581
MSG-Initiated Rule Action Specification panel 587
Rule Processor Action Specification panel 87
TIME-Initiated Rule Action Specification panel 594
VAR-Initiated Rule Action Specification panel 598
CMD-Initiated Rule
selecting criteria and actions 551–555
CMDSHOW primary command
Rule Processor application
Automation Control Panel 180
naming conventions 180
CMRCYCnn member name
BBPROF predefined refresh cycle member 477
CMRTOOLS command 15
Code field
documentation box 496
EXEC Management application 463, 468
Rule Processor Detail Control panel 69
Rule Set Overview panel 199
CODES command 13
Col field
ALERT DETAIL application 404
Color field
ALERT Detail Information display 412
Rule Processor ALERT Action(s) panel
ALERT Action(s) Specification panel 98
command
associated with an ALERT 405
Command field
ALERT Detail Information display 413
ALERT Overview application 400
Command ID field
CMD-Initiated Rule Selection panel 551
Command string field
CMD-Initiated Rule Selection panel 551
commands
BBI control 24
scroll 23
common application functions in AutoOPERATOR 9
common display fields in AutoOPERATOR 9
COMPLETE
CSM object state 330
conflicts
CSM
schedules 304
CONNECT command (CICS) 472
connecting
CSM
objects 223
CONNXPN2 command (CICS) 472
CONNXPND command (CICS) 472
Console ID field
CMD-Initiated Rule Selection panel 552
JES3-Initiated Rule Selection panel 573
MSG-Initiated Rule Action Specification panel 588
MSG-Initiated Rule Selection panel 585
Rule Processor Action Specification panel 86
Rule Processor Selection Criteria panel
CMD-Initiated Rule 76
JES3-Initiated Rule 76
MSG-Initiated Rule 76
Console Name field
CMD-Initiated Rule Selection panel 552
JES3-Initiated Rule Selection panel 573
MSG-Initiated Rule Selection panel 585, 588
Rule Processor Action Specification panel 86
Rule Processor Selection Criteria panel
CMD-Initiated Rule 76
JES3-Initiated Rule 76
MSG-Initiated Rule 76

CONSOLES command (CICS) 472
Context field
ALRM-Initiated Rule Selection panel 536
Rule Processor Selection Criteria panel
ALRM-Initiated Rule 76
Continuous State Manager 211
Abnormal Termination Events 226
Object 214
object
CICS 217
DB2 217
JES2 217
MVS 217
NET 217
TMSINIT 217
TSO 217
schedules 220
Start Command Limits 225
Continuous State Manager (CSM)
abnormal termination events 268
accessing 247
associating objects with a group 272
benefits 213
command specification 262
conditional command specification 265
creating a group 256
cross-system management
description 243
enabling 308
defining objects 255
defining parents 274
defining servers 275
describing 218
actual states 219
bouncing objects 316
canceling objects 321
client-server relationship 217
connecting 223
disconnecting 223
grouping object 215, 278
messages 222
moving objects 323
normal object 214
object states 327–337
parent-child relationship 215, 216
resetting objects 318
schedules 220
starting commands 222
states 218
stopping commands 222
transient object 215, 278
examples
exceptional schedules 237
IPL 230
routine schedules 235
shutdown 232
Global Overview panel 249
masking panels 248
moving objects 273
overview 212–238
panel description
Client/Server Relationships 373
Command Specifications 353
Conditional Command Specifications 359
CSM Global Overview 338
Event Specifications 1 361
Event Specifications 2 365
Global Calendar Override 379
Group Detail Control 385
Object Detail Control 349
Object Groups 369, 383
Object Requisites 371
Object Scheduling 376
User Notification 367
parents
adding 282
deleting 286
planning 239
commands 241
object relationships 240
schedules 240
recovering objects 225
schedules
conflicts 304
creating 294
modifying 296, 299
overriding 306
scheduling EXEC 326
scheduling objects 225
servers
adding 288
deleting 291
specifying start/stop events 267
specifying user notification 270
start an object 311
starting CSM 224
stop an object 314
Copy command
Rule Set Overview panel 204
Count field
Rule Processor application
Event Activity Statistics application 51
CPU
Sample Monitoring Solution 502
CPU field
EXEC Management application 467
CPU% field
EXEC Management application 467
CPUL field
EXEC Management application 467
CPUL% field
EXEC Management application 467
creating Rules
See Rule Processor application
Creator field
ALERT Detail Information display 412
Crit. field
ALERT Queue Statistics application 397

Critical ALERT 400
cross-system management
CSM
enabling 308
CSAABV monitor
Sample Monitoring Solution 501
CSABLO monitor
Sample Monitoring Solution 501
CSM Global Overview panel
description 249–251
field descriptions 339
line commands 344
primary commands 340
CSMACT
CSM EXEC
commands 332
description 228
examples 332
modify objects 331
parameters 331
CSMINFO
CSM EXEC
description 330
CSSUM command
Sample Monitoring Solution 501
CSTATUS command (CICS) 472
CT command 15, 17
CTIO command 16
cumulative data
Automation Reporter 430
EVENT 431
EXEC 431
RULE 431
RULESET (AUTO) 431
RULESET (SET) 431
Current Arrival Rate field
Rule Processor application
Automation Control panel 177
customizing 481
Log Display 481
CYCLE command 13
CYCLE SETUP application
general services
BBPROF predefined refresh cycle member 477–479
fields 470
sample panel 469
SAVE command 479
SERVICE REFRESH CYCLE 479
service select code 471–476
CYCxx member name
BBPROF predefined refresh cycle member 477

D

D A command
BBI control 24
D E HIGH command
BBI control 24
D RULES command
BBI control 25
D V command
BBI control 24
DA command 18
DA command (MAO) 471
DASD command 18
data
Automation Reporter
cumulative 429
interval 429
offloading 422
data set
Automation Reporter
disposition 420
specifying 416, 419
DATABASE command 15, 16, 17
DATABL command (CICS) 472
Date field
ALERT Detail Information display 412
Rule Processor application
Automation Control panel 184
DB2 217
event type 44, 53
Product Line Transfer name 10
DB2EX command (DB2) 474
DB2-Initiated Rule
selecting criteria and actions 556–561
DB2ST command (DB2) 474
DB2SYSP command (CICS) 472
DB2TASK command (CICS) 472
DBCTASK command (CICS) 472
DBCTL command (CICS) 472
DBCTTASK command (CICS) 472
DBTS command (DB2) 474
DDFCV command (DB2) 474
DDFDT command (DB2) 474
DDFSM command (DB2) 474
DDFVT command (DB2) 474
DDIR command (CICS) 472
DDIRXPN2 command (CICS) 472
DDIRXPND command (CICS) 472
DE command 17
defining
CSM
exceptional schedules 379
definitions 26
Delete command
ALERT Queue Statistics application 397
Rule Set Overview panel 204
delete operator message 96
Desc codes field
JES3-Initiated Rule Selection panel 573
MSG-Initiated Rule Selection panel 584
Rule Processor Selection Criteria panel
JES3-Initiated Rule 76
MSG-Initiated Rule 76
DESC field
documentation box 496
EXEC Management application 468

Description field
documentation box 496
EXEC Management application 463
Rule Processor Detail Control panel 69
Rule Set Overview panel 199
Descriptor codes field
MSG-Initiated Rule Action Specification panel 589
Rule Processor Action Specification panel 88
desired states 218
DEST command (CICS) 472
DETAIL command
ALERT DETAIL application 407
determining schedules
CSM 294
DIS command (MAO) 471
DISA EXEC
TIMEXEC application 488
Disable command
EXEC Management application 463
Rule Processor application
Automation Control panel 185
Rule Set Overview panel 203
disconnecting
CSM
objects 223
DISP field
documentation box 496
EXEC Management application 468
Display at dest. field
ALRT-Initiated Rule Action Specification panel 543
CICS-Initiated Rule Action Specification panel 548
DB2-Initiated Rule Action Specification panel 558
IMS-Initiated Rule Action Specification panel 569
JES3-Initiated Rule Action Specification panel 574
JRNL-Initiated Rule Action Specification panel 579
MSG-Initiated Rule Action Specification panel 586
DISPLAY command 18
BBI control 24
display fields
common in AutoOPERATOR 9
Display Suppressed field
Rule Processor application
Automation Control panel 182
DL/I command (CICS) 472
DLOGS command (DB2) 474
DM command 15, 17
DMON command (DB2) 474
documentation box
example 495
fields 496–497
DOM 96
DOM ID
ALRM-Initiated Rule Action Specification panel 540
ALRT-Initiated Rule Action Specification panel 545
CICS-Initiated Rule Action Specification panel 550
CMD-Initiated Rule Action Specification panel 555
DB2-Initiated Rule Action Specification panel 561
EXT-Initiated Rule Action Specification panel 566
IMS-Initiated Rule Action Specification panel 571
JES3-Initiated Rule Action Specification panel 577
JRNL-Initiated Rule Action Specification panel 582
MSG-Initiated Rule Action Specification panel 589
TIME-Initiated Rule Action Specification panel 595
VAR-Initiated Rule Action Specification panel 599
DOWN
CSM object state 328
DOWNEARLY
CSM object state 328
DOWNFORCE
CSM object state 329
DOWNLOGICAL
CSM object state 329
DOWNLOGICALE
CSM object state 329
DPM command 14
DRIVER EXEC
TIMEXEC application 488
DSAS command (CICS) 472
DSNAMES command (CICS) 472
DTIME field
general services
CYCLE SETUP application 470
SERVICE REFRESH CYCLE 479
DTRAC command (DB2) 474
DUSER command (DB2) 474
DW command 15, 17
DWARN command (DB2) 474
Dynamic Parameter Manager
activating
AAOARPxx 416
Automation Reporter 416

E

EDMPL command (DB2) 474
EMA command 14, 471
Enable command
EXEC Management application 463
Rule Processor application
Automation Control panel 185
Rule Set Overview panel 203
Enabled field
EXEC Management application 460
enabling
CSM
cross-system management 308
Enhanced Journal Facility 481
ENQ command (MAO) 471
ENQUEUE command (CICS) 472
ENQUEUES command 18
Escalate field
Rule Processor application
ALERT Action(s) Specification panel 100
EVE command 14
event 43–54
Automation Reporter
cumulative data 431
example 445

defining events 43
special events
DB2 53
EXT 53
TIME 54
VAR 54
types
ALRM 43
ALRT 43
CICS 44
CMD 44
DB2 44
EXT 44
IMS 45
JES3 45
JRNL 45
MQS 45
MSG 45
TIME 45
VAR 45
Event Activity Statistics application
accessing 46
MCTSIZE 52
resetting 48
Rule Processor application
fields 51–52
line commands 48
primary commands 49–50
sample panel 47
event type
See event
Event Type field
Rule Processor Detail Control panel 66
event types
CMD-Initiated Rule 551
Rules
ALRM-Initiated Rule 536–540
ALRT-Initiated Rule 541–545
CICS-Initiated Rule 546–550
CMD-Initiated Rule 551–555
DB2-Initiated Rule 556–561
EXT-Initiated Rule 562–566
IMS-Initiated Rule 567–571
JES3-Initiated Rule 572–577
JRNL-Initiated Rule 578–582
MSG-Initiated Rule 583–589
Time-Initiated Rule 590–595
VAR-Initiated Rule 596–599
event variables 161
See also variables
&IMFEVFRD 170
&IMFGROUP 170
&IMFMSTYP 171
Rule Processor application 161, 169
&IMFACCTG 169
&IMFALID 169
&IMFALRM 169
&IMFAPRI 169
&IMFAQID 169
defining 161
IMFCLOCK 169
IMFCNTXT 170
IMFCOMID 170
IMFCONNM 170
IMFDAY 170
IMFDDNAM 170
IMFDOMID 170
IMFETYPE 170
IMFJCLAS 170
IMFJNUM 171
IMFJTYPE 171
IMFLPROD 171
IMFLTYPE 171
IMFLUSER 171
IMFMPFAU 171
IMFMPFSP 171
IMFOASID 171
IMFODATE 171
IMFOJOB 172
IMFOQID 172
IMFORGN 172
IMFORGSS 172
IMFOTIME 173
IMFPCMD 173
IMFREPLY 173
IMFRLFRD 173
IMFRLID 173
IMFRLMAT 173
IMFRLSET 173
IMFRLSTA 173
IMFRUSER 173
IMFSTOKN 173
IMFSYSID 173
IMFTEXT 173
IMFTOKEN 174
IMFVIEW 174
IMSCOPE 173
WORDxxx 174
events
Rule Processor application 32
Events Handled field
Rule Processor application
Automation Control panel 182
EVNT
Automation Reporter
data record 428
EX command 17
EX command (IAO) 471
EXEC
associated with an ALERT 405
Automation Reporter
cumulative data 431
data record 428
example 446
command
application transfer 14, 15, 17
information line 468
EXEC command

service select code 471
EXEC descriptions
TIMEXEC application 488–489
EXEC field
ALERT Detail Information display 413
Rule Processor ALERT Action(s) panel
ALERT Action(s) Specification panel 98
Rule Processor application
Rule Set Overview panel 198
TIMEXEC application 484
EXEC invocation 25
EXEC Management application
/*DOC statement 468
ABEND field 464
AUTHOR field 463
AVGCPU field 465
Changed field 465
CODE field 463
CPU field 467
CPU% field 467
CPUL field 467
Description field 463
documentation box 496
enabled field 460
EXEC Activity 466
EXEC description area 462
EXEC information line 468
EXEC status 460
EXECD field 464
EXECs defined field 460
EXPAND command 460
fields 460–465, 467–468
FUNCTION field 463
GO command 460
GROUP field 463
High Priority field 460
LC field 463, 467
line commands 462, 466
MAXCPU field 465
NAME field 463, 464, 467
Norm Priority field 461
ORIGIN field 467
primary commands 460
sample panels
EXEC Management - View 2 459
EXEC Management for Currently Active EXECs 466
EXEC Management with EXEC Description-View 1 458
Scheduled field 460
SEQ# field 467
SORT command 460
STATE field 467
STATUS field 463, 464
TIME field 467
TOTCPU field 464
TYPE field 467
using 458–465
using for currently active EXECs 466–467
EXEC name/Parms field
Rule Processor Action Specification panel 89
EXEC name|Parms field
ALRM-Initiated Rule Action Specification panel 538
ALRT-Initiated Rule Action Specification panel 544
CICS-Initiated Rule Action Specification panel 549
CMD-Initiated Rule Action Specification panel 553
DB2-Initiated Rule Action Specification panel 559, 560
EXT-Initiated Rule Action Specification panel 564
IMS-Initiated Rule Action Specification panel 570
JES3-Initiated Rule Action Specification panel 575, 576
JRNL-Initiated Rule Action Specification panel 579
MSG-Initiated Rule Action Specification panel 587
Time-Initiated Rule Action Specification panel 593
VAR-Initiated Rule Action Specification panel 597
EXEC parameter
TIMEXEC application 487
EXECD field
EXEC Management application 464
EXECs
$CSAABV 503
$CSABLO 503
$JOBCPU 503
$STCCPU 503
$TOTCPU 503
$TOTSIO 503
$TOTSIO2 503
$TSOCPU 503
$WATCH 503
BBPROC 493
CSM
scheduling 326
IEF450I 507
MMN002C 503
MMN003C 503
naming conventions 494
national characters 497
NETACT 500
NETBOUNC 500
NETINACT 500
OSPINFO1 507
TIOTSIO2 501
EXECs defined field
EXEC Management application 460
Execute CMR command 15
EXITS command (CICS) 472
EXPAND command
ALERT DETAIL application 407
ALERT Overview application 401
EXEC Management application 460
EXT
event type 44, 53
Extended ALERTs
sample panel 410
EXTERNAL ROUTINES CALLED field
documentation box 497
EXT-Initiated Rule
selecting criteria and actions 562–566

F

failed objects
  recovering objects 225
features
  AutoOPERATOR 4
fields
  ALERT DETAIL application 404–413
  ALERT Overview application 399–403
  ALERT Queue Statistics application 397
  common application 22–23
  EXEC Management application 460–465, 467–468
  Rule Processor application
    ALERT Action(s) Specification panel 97
    ALRM-Initiated Action Specification panel 538
    ALRM-Initiated Rule Selection Criteria panel 536
    ALRT-Initiated Rule Action Specification panel 543
    ALRT-Initiated Rule Selection Criteria panel 541
    Automation Control panel 181, 182, 184, 194
    CICS-Initiated Rule Action Specification panel 548
    CICS-Initiated Rule Selection Criteria panel 546
    CMD-Initiated Rule Action Specification panel 553
    CMD-Initiated Rule Selection Criteria panel 551
    DB2-Initiated Rule Action Specification panel 558
    DB2-Initiated Rule Selection Criteria panel 556
    Event Activity Statistics application 51–52
    EXT-Initiated Rule Action Specification panel 564
    EXT-Initiated Rule Selection panel 562
    IMS-Initiated Rule Action Specification panel 569
    IMS-Initiated Rule Selection Criteria panel 567
    JES3-Initiated Rule Action Specification panel 574
    JES3-Initiated Rule Selection Criteria panel 572
    JRNL-Initiated Rule Action Specification panel 579
    JRNL-Initiated Rule Selection panel 578
    MSG-Initiated Rule Action Specification panel 586
    MSG-Initiated Rule Selection panel 583
    Rule Processor Detail Control panel 65, 66–69
    Time-Initiated Rule Action Specification panel 593
    Time-Initiated Rule Selection panel 590
    Variable Dependencies 82
    VAR-Initiated Rule Action Specification panel 597
    VAR-Initiated Rule Selection Criteria panel 596
  TIMEXEC application 484
FILE command (CICS) 472
FILEXPN2 command (CICS) 472
FILEXPND command (CICS) 472
Filter command
  Rule Set Overview panel 203
Filtered field
  Rule Processor application
    Automation Control panel 184
  values 184
filtering
  Rule Sets 183
FIND command
  general services
    Log Display 480
  Rule Processor application
    Event Activity Statistics application 49
Fired field
  Rule Processor application
    Automation Control panel 184
    Rule Set Overview panel 198
FIRST
  Rule Processor application
    Automation strategy 186
FOCAL command 13
FUNC field
  documentation box 496
  EXEC Management application 468
Function field
  EXEC Management application 463
  Rule Processor ALERT Action(s) panel
    ALERT Action(s) Specification panel 97
  Rule Processor application
    Rule Set Overview panel 199
  Rules
    Rule Processor Detail Control panel 69
function key definitions
  See PF key definitions

G

GC command 15, 17
general services
  BBPROF predefined refresh cycle member 477–479
  CALLX request 486–489
  CYCLE SETUP application 469–479
  KEYS 483
  Log Display 480
  MESSAGES 483
  sample panels
    General Services menu 469
    SERVICE REFRESH CYCLE 469
    TIME-INITIATED EXEC REQUESTS 484
  SAVE command 479
  SERVICE REFRESH CYCLE 479
  service select code
    CYCLE SETUP application 471–476
  TIMEXEC application 484–489
GO command
  ALERT DETAIL application 407
  ALERT Overview application 401
  EXEC Management application 460
  Refresh Mode 23
  SERVICE REFRESH CYCLE 479
GRAPH command (CICS) 472
Group field
  documentation box 496
  EXEC Management application 463, 468
  Rule Processor Detail Control panel 69
  Rule Set Overview panel 199
grouping objects
  CSM 215
    adding 278
GT command 15

H

H E command
    BBI control 25
Handled field
    Rule Processor application
        Event Activity Statistics application 51
Hardcopy Suppressed field
    Rule Processor application
        Automation Control panel 182
help
    ALERT 405
HELP command
    BBI control 25
Help field
    ALERT Detail Information display 412
    Rule Processor ALERT Action(s) panel
        ALERT Action(s) Specification panel 98
High Priority field
    EXEC Management application 460
HISTORY command 15
HOLD=YES
    TIMEXEC application 484
Honor MPF Suppression field
    Rule Processor application
        Automation Control panel 181
HT command 15, 16, 17
HTIO command 16

I

IAO
    Product Line Transfer name 10
IAOCYCnn member name
    BBPROF predefined refresh cycle member 477
ICE command (CICS) 472
ID field
    Rule Set Overview panel 198
IEF450I
    Info/System Management Solution 507, 508
If matched field
    Rule Processor Detail Control panel 67
IMFACCTG
    event variables 169
IMFALID
    event variables 169
IMFALRM
    event variables 169
IMFAPRI
    event variables 169
IMFAQID
    event variables 169
IMFCLOCK
    event variables 169
IMFCNTXT
    event variables 170
IMFCOMID
    event variables 170
IMFCONNM
    event variables 170
IMFDAY
    event variables 170
IMFDDNAM
    event variables 170
IMFDOMID
    event variables 170
IMFETYPE
    event variables 170
IMFEVFRD
    event variables 170
IMFGROUP
    event variables 170
IMFJCLAS
    event variables 170
IMFJNUM
    event variables 171
IMFJTYPE
    event variables 171
IMFLPROD
    event variables 171
IMFLTYPE
    event variables 171
IMFLUSER
    event variables 171
IMFMPFAU
    event variables 171
IMFMPFSP
    event variables 171
IMFOASID
    event variables 171
IMFODATE
    event variables 171
IMFOJOB
    event variables 172
IMFOQID
    event variables 172
IMFORGN
    event variables 172
IMFORGSS
    event variables 172
IMFOTIME
    event variables 173
IMFPCMD
    event variables 173
IMFREPLY
    event variables 173
IMFRLFRD
    event variables 173
IMFRLID
    event variables 173
IMFRLMAT
    event variables 173
IMFRLSET
    event variables 173
IMFRLSTA
    event variables 173
IMFRUSER
    event variables 173
IMFSCOPE

event variables 173
IMFSTOKN
event variables 173
IMFSYSID
event variables 173
IMFTEXT
event variables 173
IMFTOKEN
event variables 174
IMFVIEW
event variables 174
IMS
event type 45
Product Line Transfer name 10
IMS-Initiated Rule
selecting criteria and actions 567–571
in seconds field
Rule Processor Detail Control panel 68
Ind field
ALERT DETAIL application 405
INDIVIDUAL
Rule Processor application
Automation strategy 188
Info field
ALERT Queue Statistics application 397
ALRM-Initiated Rule Action Specification panel 540
ALRT-Initiated Rule Action Specification panel 545
CICS-Initiated Rule Action Specification panel 550
CMD-Initiated Rule Action Specification panel 555
DB2-Initiated Rule Action Specification panel 561
EXT-Initiated Rule Action Specification panel 566
IMS-Initiated Rule Action Specification panel 571
JES3-Initiated Rule Action Specification panel 577
JRNL-Initiated Rule Action Specification panel 582
MSG-Initiated Rule Action Specification panel 588
Rule Processor Action Specification panel 89
Time-Initiated Rule Action Specification panel 595
VAR-Initiated Rule Action Specification panel 599
Info/System Management Sample Solution
EXECs 507
invoking 507
Rules 508
Informational ALERT 400
INIT command
Primary Option Menu retrieval 10, 21
Initial ALERT sort order field
ALERT DETAIL PROFILE application 409
Initial Mode field
Rule Processor Detail Control panel 67
INPUT field
Refresh Mode 23
INPUT PARAMETERS field
documentation box 496
Insert command
Rule Set Overview panel 204
interval
Automation Reporter
specifying 421
interval data
Automation Reporter 429
ACTN 430
ALRT 430
RES 430
TAPE (LOCAL) 430
TAPE (PLEX) 430
INTERVAL field
ALERT DETAIL application 404
ALERT Overview application 400
Refresh Mode 23
Time-Initiated Rule Selection panel 590
INTERVAL keyword
Refresh Mode 23
introduction
AutoOPERATOR 3
INTVL field
TIMEXEC application 485
INTVL parameter
TIMEXEC application 486, 487
INTVL= keyword
BBPROF predefined refresh cycle member 478
INVOKED BY field
documentation box 496
invoking AutoOPERATOR EXECs 25
IO command 16
ISC command 16
Issue WTO MSG field
ALRM-Initiated Rule Action Specification panel 540
ALRT-Initiated Rule Action Specification panel 545
CICS-Initiated Rule Action Specification panel 550
CMD-Initiated Rule Action Specification panel 555
DB2-Initiated Rule Action Specification panel 561
EXT-Initiated Rule Action Specification panel 566
IMS-Initiated Rule Action Specification panel 571
JES3-Initiated Rule Action Specification panel 577
JRNL-Initiated Rule Action Specification panel 582
MSG-Initiated Rule Action Specification panel 589
TIME-Initiated Rule Action Specification panel 595
VAR-Initiated Rule Action Specification panel 599

J

JES2 217
JES3
event type 45
JES3-Initiated Rule
selecting criteria and actions 572–577
Job Class field
Rule Set Overview panel 202
Job name field
CMD-Initiated Rule Selection panel 552
DB2-Initiated Rule Selection panel 557
EXT-Initiated Rule Selection panel 563
IMS-Initiated Rule Selection panel 568
JES3-Initiated Rule Selection panel 573
MSG-Initiated Rule Selection panel 584
Rule Processor Selection Criteria panel
CMD-Initiated Rule 77
DB2-Initiated Rule 77

EXT-Initiated Rule 77
IMS-Initiated Rule 77
JES3-Initiated Rule 77
MSG-Initiated Rule 77
Rule Set Overview panel 202
Job Type field
Rule Set Overview panel 202
Jobclass field
CMD-Initiated Rule Selection panel 552
EXT-Initiated Rule Selection panel 563
IMS-Initiated Rule Selection panel 568
JES3-Initiated Rule Selection panel 573
MSG-Initiated Rule Selection panel 584
Rule Processor Selection Criteria panel
CMD-Initiated Rule 77
EXT-Initiated Rule 77
IMS-Initiated Rule 77
JES3-Initiated Rule 77
MSG-Initiated Rule 77
JOBCPU monitor
Sample Monitoring Solution 502
JOU command 471
Journal
ALRT-Initiated Rule Action Specification panel 543
CICS-Initiated Rule Action Specification panel 548
CMD-Initiated Rule Action Specification panel 553
DB2-Initiated Rule Action Specification panel 558
IMS-Initiated Rule Action Specification panel 569
JES3-Initiated Rule Action Specification panel 574
VAR-Initiated Rule Action Specification panel 597
JOURNAL command 13
JOURNAL command (CICS) 472
Journal field
EXT-Initiated Rule Action Specification panel 564
MSG-Initiated Rule Action Specification panel 586
Rule Processor Action Specification panel 90
Time-Initiated Rule Action Specification panel 593
Journal log
customizing views 481
Enhanced 481
Journal log messages
BBI-SS PAS 25
JRNL
event type 45
JRNL-Initiated Rule
selecting criteria and actions 578–582
jump function 9

K

key definitions
See PF key definitions
Key field
ALERT Detail Information display 412
ALRT-Initiated Rule Selection panel 542
Rule Processor ALERT Action(s) panel
ALERT Action(s) Specification panel 97
Rule Processor Selection Criteria panel
ALRT-Initiated Rule 77
KEYS
command 13
general services 483
keywords
BBPROF predefined refresh cycle member
general services 477

L

Last Modified field
Rule Processor Detail Control panel 69
LC field
EXEC Management application 463, 467
Rule Processor application
Event Activity Statistics application 51
TIMEXEC application 484
line command 16, 17, 19
line commands
ALERT Queue Statistics application 397
EXEC Management application 462, 466
Rule Processor application
Automation Control panel 177–194
Event Activity Statistics application 48
Rule Set Overview panel 197
LOCATE command
general services
Log Display 480
Rule Processor application
Event Activity Statistics application 49
LOCKD command (DB2) 474
LOCKE command (DB2) 474
LOCKU command (DB2) 474
log
display
customizing 481
viewing subsets 481
LOG command 13, 471
Log Display
general services 480
LOG field
CYCLE SETUP application 470
log messages
BBI-SS PAS Journal 25
LOG= keyword
BBPROF predefined refresh cycle member 478
LPAS command (CICS) 472
LTERM
command 16
LTRAC command (DB2) 474

M

MAINVIEW Alarm Manager
solution
ALRMSTOP 509
ALRMSTRT 509
disabling 510
enabling 509
MAINVIEW for IMS command 17

MAJNODE command 19
Major ALERT 400
Major field
ALERT Queue Statistics application 397
managing Rules 39
See also Rule Processor application
MAO
Product Line Transfer name 10
MAOCYCnn member name
BBPROF predefined refresh cycle member 477
MAS command 14
masking panel displays
CSM application 248
MAXCPU field
EXEC Management application 465
MDEV command
Sample Monitoring Solution 501
member names
BBPROF predefined refresh cycle member 477
messages
BBI-SS PAS Journal log 25
general services 483
Minor ALERT 400
Minor field
ALERT Queue Statistics application 397
MMN002C
Sample Monitoring Solution 503
MMN003C
Sample Monitoring Solution 503
MN command 16, 17
MODIFY
Alert Action(s) I panel
description 103
modifying Alerts 103
MONITOR command (CICS) 472
Monitoring Sample Solution
Operations Monitors Sample Solution
EXECs 503
invoking 501
processing flow 501
Move command
Rule Processor application
Automation Control panel 185
Rule Set Overview panel 204
MQS
event type 45
MS command 17
MSG
event type 45
MSG command 13
MSG-Initiated Rule 78
selecting criteria and actions 583–589
MSGSTATS command 14
MTODMIN member name
BBPROF predefined refresh cycle member 477
MV Solution Samples
CICSPlex Sample Solution 505
index to 531
Info/System Management Sample Solution 507–508
Monitoring Sample Solution 501–503
VTAM/NCP Sample Solution 499–500
MVS 217

N

Name field
documentation box 496
EXEC Management application 463, 464, 467
Rule Processor Selection Criteria panel
VAR-Initiated Rule 77
VAR-Initiated Rule Selection panel 596
naming convention
exceptions to 494
for Sample Solutions EXECs 494
national characters 497
NET 217
NETACT 500
NETBOUNC 500
NETINACT 500
NODE command 16
non-SNA terminal
SERVICE REFRESH CYCLE 479
Norm Priority field
EXEC Management application 461
normal object
CSM 214
normal objects
CSM
adding parents 274
adding servers 275
Notify field
ALRM-Initiated Rule Action Specification panel 540
ALRT-Initiated Rule Action Specification panel 545
CICS-Initiated Rule Action Specification panel 550
CMD-Initiated Rule Action Specification panel 555
DB2-Initiated Rule Action Specification panel 561
EXT-Initiated Rule Action Specification panel 566
IMS-Initiated Rule Action Specification panel 571
JES3-Initiated Rule Action Specification panel 577
JRNL-Initiated Rule Action Specification panel 582
MSG-Initiated Rule Action Specification panel 588
Rule Processor Action Specification panel 90
Time-Initiated Rule Action Specification panel 595
VAR-Initiated Rule Action Specification panel 599
NUCLEUS command (CICS) 472
NV command 14

O

object relationships
CSM 215
objects
CSM 222
bouncing 316
canceling 321
moving 323
resetting 318
starting 311

stopping 314
offload interval
Automation Reporter
specifying 416, 422
Op field
Variable Dependencies panel 82
OPER command (MAO) 471
Operations Monitors Sample Solution 501–503
OPERATOR command 18
OPERATOR member name
BBPROF predefined refresh cycle member 477
operator-initiated EXECs 494
OPT= keyword
BBPROF predefined refresh cycle member 477
OR command 18
OR command (MAO) 471
organizing Rules 39
See also Rule Processor application
Origin field
ALERT DETAIL application 406
ALERT Detail Information display 412
EXEC Management application 467
JRNL-Initiated Rule Selection panel 578
Rule Processor ALERT Action(s) panel
ALERT Action(s) Specification panel 98
Rule Processor Selection Criteria panel
JRNL-Initiated Rule 77
OSPI application 507
OSPI command 14
OSPINFO1
Info/System Management Solution 507
OUTBOARD CALLS field
documentation box 496
OUTPUT field
documentation box 496
overriding
CSM
schedules 306, 379

P

PA1 key
SERVICE REFRESH CYCLE 479
parameters
TIMEXEC application 487–488
parent-child relationship
CSM 216
parents
CSM
adding 282
deleting 286
pattern matching
Rule Processor application 156–160
debugging 160
example 159
wildcard characters 156
PCMD field
Rule Processor ALERT Action(s) panel
ALERT Action(s) Specification panel 98
PD command 18
Peak Arrival Rate field
Rule Processor application
Automation Control panel 182
PF key definitions
sample panel 26
PF keys
assigning functions 27
assigning labels 27
PF6/18 (GO) command
SERVICE REFRESH CYCLE 479
PLAN command (CICS) 473
PLANXPND command (CICS) 473
PLOT command (DB2) 474
PM command 16, 18
PPST command (CICS) 473
predefined refresh cycle member
CYCLE SETUP application 477
prefix 24
Primary Command
common application field 22
INIT 21
RETURN 21
primary commands
EXEC Management application 460
general services
Log Display 480
Rule Processor application
Automation Control 179
Event Activity Statistics application 50
Rule Processor Detail Control panel 65, 73
Rule Set Overview panel 203
Primary Option Menu 6
Primary Option Menu retrieval 21
priorities for ALERTs 400
Priority field
ALERT Detail Information display 412
ALRM-Initiated Rule Selection panel 537
Rule Processor ALERT Action(s) panel
ALERT Action(s) Specification panel 98
Rule Processor Selection Criteria panel
ALRM-Initiated Rule 77, 78, 79
PROBLEM command (CICS) 473
Product Line Transfer request 10–11
PROFILE command 481
Profile command
ALERT DETAIL application 406
ALERT Overview application 401
Profile Suffix field
ALERT Overview application 400, 402–403
PROGRAM command 15, 17
PROGRAM command (CICS) 473
program function key definitions
See PF key definitions
program function keys 27
assigning functions 27
assigning labels 27
PROGRESS command
Sample Monitoring Solution 502

PSBNAME command (CICS) 473
Publish field
Rule Processor application
ALERT Action(s) Specification panel 99
PUT command 15

Q

QIS parameter
TIMEXEC application 488
QUALIFIED
Rule Processor application
Automation strategy 187
Queue field
ALRM-Initiated Rule Selection panel 537
ALRT-Initiated Rule Selection panel 542
Queue(s) field
ALERT DETAIL application 405
ALERT Detail Information display 412
ALERT Management Facility
ALERT Queue Statistics application 397
Rule Processor ALERT Action(s) panel
ALERT Action(s) Specification panel 98
Rule Processor Selection Criteria panel
ALRM-Initiated Rule 78
ALRT-Initiated Rule 78
queues for ALERTs 399

R

RACF Group field
CMD-Initiated Rule Selection panel 552
EXT-Initiated Rule Selection panel 563
IMS-Initiated Rule Selection panel 568
JES3-Initiated Rule Selection panel 573
MSG-Initiated Rule Selection panel 584
Rule Processor Selection Criteria panel
CMD-Initiated Rule 78
EXT-Initiated Rule 78
IMS-Initiated Rule 78
JES3-Initiated Rule 78
MSG-Initiated Rule 78
Rule Set Overview panel 202
RACF User field
CMD-Initiated Rule Selection panel 552
EXT-Initiated Rule Selection panel 563
IMS-Initiated Rule Selection panel 568
JES3-Initiated Rule Selection panel 573
MSG-Initiated Rule Selection panel 584
Rule Processor Selection Criteria panel
CMD-Initiated Rule 76
EXT-Initiated Rule 76
IMS-Initiated Rule 76
JES3-Initiated Rule 76
MSG-Initiated Rule 76
Rule Set Overview panel 202
RC command 18
RECNUM variable
Info/System Management Solution 507
REFRESH command 13
refresh cycle
starting and stopping 479
refresh cycle member
CYCLE SETUP application 477
refresh mode fields
common application 23
REGION command 15, 16
REGION command (IAO) 471
REGIONS command (CICS) 473
Reject Command field
CMD-Initiated Rule Action Specification panel 553
REMOTES command (CICS) 473
repeat
CSM
line commands 279
Repeat command
Rule Set Overview panel 204
REQ command (MAO) 471
REQ=
TIMEXEC application 489
request 9
AutoOPERATOR application 9
Product Line Transfer 10–11
REQUESTS command 18
RES
Automation Reporter
data record 428
interval data 430
RES command (MAO) 471
RESERVES command 18
RESET command
Rule Set Overview panel 203
resetting objects
CSM 318
resource
inactivating 500
reactivating 500
Retain field
ALERT Detail Information display 412
Rule Processor application
ALERT Action(s) Specification panel 100
retrieving Primary Option Menu 21
RETURN CODES field
documentation box 497
RETURN command
Primary Option Menu retrieval 10, 21
REVIEW command (CICS) 473
Reword Alert field
ALRT-Initiated Rule Action Specification panel 545
Reword Msg field
IMS-Initiated Rule Action Specification panel 571
JES3-Initiated Rule Action Specification panel 577
JRNL-Initiated Rule Action Specification panel 582
MSG-Initiated Rule Action Specification panel 588
Rule Processor Action Specification panel 91
Time-Initiated Rule Action Specification panel 595
RFIND command
Rule Processor application

Event Activity Statistics application 49
RIDPL command (DB2) 474
Route codes field
JES3-Initiated Rule Selection panel 573
MSG-Initiated Rule Action Specification panel 589
MSG-Initiated Rule Selection panel 584
Rule Processor Action Specification panel 91
Rule Processor Selection Criteria panel
JES3-Initiated Rule 78
MSG-Initiated Rule 78
RSP field
ALERT DETAIL application 405
RULE
Automation Reporter
cumulative data 431
data record 428
example 447
Rule generated ALERTs field
Automation Control panel 182
Rule ID field
naming conventions
valid characters 66
Rule Processor Detail Control panel 66
Rule invoked EXECs field
Automation Control panel 182
Rule Processor application
action specification fields 85–92
ALERT Action(s) Specification 93–104
ALRM-Initiated Rule 536–540
ALRT-Initiated Rule 541–545
AND/OR logic 81
Automation Control 177–194
Automation Statistics 182
Automation Status 178
Automation Strategy 178, 186–194
BBPARM 493
CICS-Initiated Rule 546–550
CMD-Initiated Rule 551–555
creating for each event type 535–595
DB2-Initiated Rule 556–561
Event Activity Statistics application 46–52
event types 43–54
EXT-Initiated Rule 562–566
FIRST Automation strategy 186
grouping into Rule Sets 183–184
IEF450I 508
IMS-Initiated Rule 567–571
INDIVIDUAL Automation strategy 188
introduction
action specification 35
Automation Control application 40
automation strategy 39
defining Rule Sets 39
defining Rules 32
events 32
Rule Set Overview application 40
selection criteria 35
JES3-Initiated Rule 572–577
JRNL-Initiated Rule 578–582
managing 39
MSG-Initiated Rule 583–589
organizing 39
pattern matching 155, 156–160
QUALIFIED Automation strategy 187
Rule Processor Detail Control panel 73
Rule Set Overview 197–206
sample panels
action specification 85
ALERT Action(s) I specification 94
ALRM action specification 536
ALRT action specification 543
ALRT selection criteria 541
Automation Control 59, 60, 61
Automation Menu 46
Event Activity Statistics 47
JRNL-Initiated Rule Action Specification 579
JRNL-Initiated Rule Selection Criteria 578
MSG-Initiated Rule Action Specification 586
MSG-Initiated Rule Selection Criteria 583
Rule Processor Detail Control 58, 62, 72
Rule Processor Event Types 63
Rule Set Overview-View 1 197
Rule Set Overview-View 2 199
Rule Set Overview-View 3 200
Rule Set Overview-View 4 201
Rule Set Overview-View 5 202
selection criteria 75
Time-Initiated Rule Action Specification 593
Time-Initiated Rule Selection Criteria 590
selection criteria fields 75–79
setting occurrences 70
Time-Initiated Rule 590–595
variable 161
variable dependencies 80–84
variable dependencies with AND/OR Logic 81
variable usage 155
VAR-Initiated Rule 596–599
wildcard characters 156
Rule Processor Detail Control
field summary 71
fields 66–69
primary commands 65, 73
sample panel 58, 61, 62
setting occurrences 70
Rule Set
filtering 183
Rule Set Filter 188
Rule Set Overview
line commands 203
primary commands 203
sample panels 197–202
Rule Set Overview application
introduction 40
Rule Set to be Updated field
Event Activity Statistics application 51
Rule Sets
adding 179
grouping 183–184

naming conventions 179
Rule-id field
Rule Set Overview panel 198
RULES
command 14
RULES command 471
Rules field
Automation Control panel 184
RULESET (AUTO)
Automation Reporter
cumulative data 431
data record 428
example 448
RULESET (RES)
Automation Reporter
example 451
RULESET (SET)
Automation Reporter
cumulative data 431
data record 428
example 449
Rule-Set field
Automation Control panel 184
RUN/RUNNING state
Refresh Mode fields 23

S

S command
Rule Processor application
Event Activity Statistics application 48
S1 primary command
Rule Processor application
Rule Processor Detail Control panel 65
SAVE command
CYCLE SETUP application 479
Rule Processor application
Automation Control panel 185
Scheduled field
EXEC Management application 460
schedules
CSM 220
conflicts 304
creating 294
determining 294
exceptional 221
modifying 296, 299
overriding 306
routine 220
Scope field
ALRM-Initiated Rule Selection panel 537
scroll commands
common application 23
SD command 15, 16, 18
SEE ALSO field
documentation box 496
Select command
ALERT Queue Statistics application 397
EXEC Management application 463
Rule Processor application
Automation Control panel 185
Rule Set Overview panel 203
SERVICE REFRESH CYCLE application 479
selection criteria
Rule Processor application 35
Selection Criteria fields
Rule Processor application 75–79
Send (TSO) field
ALRM-Initiated Rule Action Specification panel 538
ALRT-Initiated Rule Action Specification panel 544
CICS-Initiated Rule Action Specification panel 549
CMD-Initiated Rule Action Specification panel 554
DB2-Initiated Rule Action Specification panel 559, 560
EXT-Initiated Rule Action Specification panel 565
IMS-Initiated Rule Action Specification panel 570
JES3-Initiated Rule Action Specification panel 575, 576
JRNL-Initiated Rule Action Specification panel 580, 581
MSG-Initiated Rule Action Specification panel 587
Rule Processor Action Specification panel 91
Time-Initiated Rule Action Specification panel 594
VAR-Initiated Rule Action Specification panel 598
SEQ# field
EXEC Management application 467
SERV= keyword
BBPROF predefined refresh cycle member 477
server
CSM
adding 288
deleting 291
SERVICE field
CYCLE SETUP application 470
SERVICE REFRESH CYCLE
general services 479
service select code
CYCLE SETUP application 471–476
services
See general services
SESSIONS command (CICS) 473
SET command
ALERT DETAIL application 407
ALERT Overview application 401
TIMEXEC application 488
Set variable field
ALRM-Initiated Rule Action Specification panel 539
ALRT-Initiated Rule Action Specification panel 545
CICS-Initiated Rule Action Specification panel 550
CMD-Initiated Rule Action Specification panel 555
DB2-Initiated Rule Action Specification panel 561
EXT-Initiated Rule Action Specification panel 565
IMS-Initiated Rule Action Specification panel 571
JES3-Initiated Rule Action Specification panel 577
JRNL-Initiated Rule Action Specification panel 580, 581
MSG-Initiated Rule Action Specification panel 588
Rule Processor Action Specification panel 91
Time-Initiated Rule Action Specification panel 594
VAR-Initiated Rule Action Specification panel 599
setting occurrences
Rule Processor Detail Control panel 70

SHARE command (CICS) 473
short messages
SERVICE REFRESH CYCLE 479
Short-Message
common application field 22
SHOWALL command
Rule Processor application
Event Activity Statistics application 49
SHOWMESSAGE command
Rule Processor application
Event Activity Statistics application 49
SHOWPER command
Rule Processor application
Event Activity Statistics application 49
SHOWTOP command
Rule Processor application
Event Activity Statistics application 50
skeleton tailoring
defined 611
SM command 15, 16, 18
SMFDUMP EXEC
TIMEXEC application 488
SNA terminal
SERVICE REFRESH CYCLE 479
SOF command 14
solution
MVALARM
disabling 510
enabling 509
SORT command
EXEC Management application 460
Rule Processor application
Event Activity Statistics application 50
Rule Set Overview panel 203
Sort Criterion field
Rule Processor application
Event Activity Statistics application 51
Sorted by field
ALERT DETAIL application 406
sorting
ALERT Queue Statistics display 398
ST command 15, 16, 18
STA command (MAO) 471
START field
TIMEXEC application 485
START parameter
TIMEXEC application 486, 487
Start Time field
Time-Initiated Rule Selection panel 590
starting objects
CSM 311
starting SERVICE REFRESH CYCLE
general services 479
Stat field
Rule Processor application
Rule Set Overview panel 198
STAT service (IMS) 18
state
CSM
actual 219
desired 218
STATE field
EXEC Management application 467
STATSHOW primary command
Rule Processor application
Automation Control Panel 180
naming conventions 180
STATUS command
AO 19
CAO 14, 471
IAO 471
MAINVIEW for IMS 16
STATUS command (CICS)
CICS 473
Status field
ALERT DETAIL application 405
ALERT Overview application 400
EXEC Management application 463, 464
Rule Processor application
Automation Control panel 184
TIMEXEC application 484
STCCPU monitor
Sample Monitoring Solution 502
STEPLIB command (CICS) 473
Stop Count field
Time-Initiated Rule Selection panel 591
STOP field
TIMEXEC application 485
STOP parameter
TIMEXEC application 486, 487
Stop Time field
Time-Initiated Rule Selection panel 591
STOPCNT parameter
TIMEXEC application 488
stopping
CSM
objects 314
stopping SERVICE REFRESH CYCLE
general services 479
STRAC command (DB2) 474
Strategy field
Rule Processor application
Automation Control panel 184
SUBPOOL command (CICS) 473
substringing
Rule Processor application 165
variables 165
SUBSYS
EXT event 53
SUFFIX command (CICS) 473
Suffix field
ALERT DETAIL application 404
SUMMARY command (CICS) 473
SV primary command
Rule Processor application
Rule Processor Detail Control panel 65
syntax notation xxiv
SYSLOG Display field

MSG-Initiated Rule Action Specification panel 586
Rule Processor Action Specification panel 92
SYSPROG commands
CPU 502
CSSUM 501
MDEV 501
PROGRESS 502
SYSPROG Services EXECs 503
SYSPROG variable
Info/System Management Solution 507
SYSTEM field
Rule Processor ALERT Action(s) panel
ALERT Action(s) Specification panel 98
System ID field
ALERT Detail Information display 412

T

TAPE (LOCAL)
Automation Reporter
data record 429
example 454
interval data 430
TAPE (PLEX)
Automation Reporter
data record 429
example 456
interval data 430
TAPE command 19
Target field
ALERT Detail Information display 412
CYCLE SETUP application 470
Rule Processor ALERT Action(s) panel
ALERT Action(s) Specification panel 98
Target Input
common application field 22
TARGET= keyword
BBPROF predefined refresh cycle member 478
TASK command (CICS) 473
TCBS command (CICS) 473
TEMPSTRG command (CICS) 473
TEMPXPND command (CICS) 473
TERMINAL command 19
TERMINAL command (CICS) 473
TERMXPND command (CICS) 473
Test command
EXEC Management application 463
Rule Processor application
Automation Control panel 185
Rule Set Overview panel 204
TEST SCRIPT INFO field
documentation box 497
Text field
ALERT Detail Information display 413
Rule Processor ALERT Action(s) panel
ALERT Action(s) Specification panel 98
Text ID field
CICS-Initiated Rule Selection panel 547
DB2-Initiated Rule Selection panel 557
EXT-Initiated Rule Selection panel 562
IMS-Initiated Rule Selection panel 567
JES3-Initiated Rule Selection panel 572
JRNL-Initiated Rule Selection panel 578
MSG-Initiated Rule Selection panel 583
Rule Processor Selection Criteria panel
ALRT-Initiated Rule 78
CICS-Initiated Rule 78
CMD-Initiated Rule 78
DB2-Initiated Rule 78
EXT-Initiated Rule 78
IMS-Initiated Rule 78
JES3-Initiated Rule 78
JRNL-Initiated Rule 78
MSG-Initiated Rule 78
TIME-Initiated Rule 78
text ID-driven EXECs 494
Text String field
ALRT-Initiated Rule Selection panel 541
Text string field
CICS-Initiated Rule Selection panel 547
DB2-Initiated Rule Selection panel 557
EXT-Initiated Rule Selection panel 562
IMS-Initiated Rule Selection panel 567
JES3-Initiated Rule Selection panel 572
JRNL-Initiated Rule Selection panel 578
MSG-Initiated Rule Selection panel 584
Rule Processor Selection Criteria Panel
ALRT-Initiated Rule 79
CICS-Initiated Rule 79
CMD-Initiated Rule 79
DB2-Initiated Rule 79
EXT-Initiated Rule 79
IMS-Initiated Rule 79
JES3-Initiated Rule 79
JRNL-Initiated Rule 79
MSG-Initiated Rule 79
TIME-Initiated Rule 79
Rule Set Overview panel 200
Text-id field
ALRT-Initiated Rule Selection panel 541
Rule Set Overview panel 198, 200
Text-ID/Description field
Rule Processor application
Event Activity Statistics application 52
TGT field
BBPROF predefined refresh cycle member 478
common application 22
Tgt switch field
ALERT Detail Information display 412
TI command 13, 14
TIME
event type 45, 54
Time command
general services
Log Display 480
Time field
ALERT DETAIL application 405
ALERT Detail Information display 412

EXEC Management application 467
Rule Processor application
Automation Control panel 184
time intervals
Automation Reporter
specifying 416
time-initiated EXECs
TIMEXEC application 484
Time-Initiated Rule
selecting criteria and actions 590–595
TIMER facility
TIMEXEC application 488
TIMEXEC application
general services
CALLX request 486–489
description 484
fields 484
parameters 487–488
sample panel 469
using 485
TIOT command (CICS) 473
TIOTSIO2
Sample Monitoring Solution 501
TMSINIT 217
Total Events field
Rule Processor application
Automation Control panel 182
Event Activity Statistics application 51
Total field
ALERT Queue Statistics application 397
Total Size field
ALERT Queue Statistics application 397
TOTCPU field
EXEC Management application 464
TOTCPU monitor
Sample Monitoring Solution 502
TOTSIO monitor
Sample Monitoring Solution 501
TRAN command (CICS) 473
TRANSACTION command 16
TRANSFER command
ALERT DETAIL application 406
ALERT Overview application 401
transient object
CSM 215
transient objects
CSM 215
TRANXPND command (CICS) 473
trimming
Rule Processor application 165
list of 169
variables 165, 168
TSO 217
TSOCPU monitor
Sample Monitoring Solution 502
TSTAT command (DB2) 474
TSUMA command (DB2) 474
TSUMC command (DB2) 474
TSUML command (DB2) 474
TSUMP command (DB2) 474
TSUMT command (DB2) 474
TSUT command (CICS) 473
Type field
ALRM-Initiated Rule Selection panel 537
CMD-Initiated Rule Selection panel 552
CYCLE SETUP application 470
EXEC Management application 467
EXT-Initiated Rule Selection panel 563
IMS-Initiated Rule Selection panel 568
JES3-Initiated Rule Selection panel 573
MSG-Initiated Rule Selection panel 584
Rule Processor application
Event Activity Statistics application 51
Rule Set Overview panel 198
Rule Processor Selection Criteria panel
ALRM-Initiated Rule 79
CMD-Initiated Rule 79
EXT-Initiated Rule 79
IMS-Initiated Rule 79
JES3-Initiated Rule 79
MSG-Initiated Rule 79
TYPE= keyword
BBPROF predefined refresh cycle member 478

U

UBBPROC data set 493
Udata field
Rule Processor ALERT Action(s) panel
ALERT Action(s) Specification panel 98
UGRAPH command 15
UNIQUE field
Rule Processor application
Event Activity Statistics application 52
UNSORT command
Rule Set Overview panel 203
UP
CSM object state 328
Update Rout/Desc Codes field
MSG Rule Action Specification panel 92
MSG-Initiated Rule Action Specification panel 589
UPEARLY
CSM object state 328
UPFORCE
CSM object state 328
UPLOGICAL
CSM object state 328
UPLOGICALE
CSM object state 328
User Data field
ALERT Detail Information display 413
User field
ALERT Detail Information display 412
Rule Processor ALERT Action(s) panel
ALERT Action(s) Specification panel 98
TIMEXEC application 485
Userid field
ALRM-Initiated Rule Selection panel 537

USERS command (DB2) 474
UTRAC command (DB2) 474

V

VAR
  event type 45, 54
Variable Dependencies
  Rule Processor application
    AND/OR logic 81
    fields 82
    primary commands 81
    sample panel 80, 81
Variable field
  Rule Processor Selection Criteria panel
    VAR-Initiated Rule 79
Variable is field
  VAR-Initiated Rule Selection panel 596
Variable-Name field
  Variable Dependencies panel 82
variables 165
  See also event variables
  Rule Processor application
    compound variables 163
    defining 161
    event variables 161
    resolving 161
    search order 161
    simple 162
    substringing 165
    trimming 165, 168
Variables for ALRT
  AMFKEY 101, 174
VARIABLES ISSUED field
  documentation box 497
Variable-Value field
  Variable Dependencies panel 84
VAR-Initiated Rule
  selecting criteria and actions 596–599
VSAM command (CICS) 473
VT command 16, 18
VTAM
  command (CICS) 473
VTAM Commands Sample Solution 499–500
  EXECs 500
VTAM/NCP Sample Solution
  VTAM Commands Sample Solution
    inactivating a resource 500
    invoking 499
    processing flow 500
    reactivating a resource 500
    Rules 500

W

Warn field
  ALERT Queue Statistics application 397
Warning ALERT 400
WATCH monitor
  Sample Monitoring Solution 502
wildcard characters
  Rule Processor application 156
WORDxxx
  event variables 174

X

XALRTS command 14, 471

Z

ZPARM command (DB2) 474

STOP!

IMPORTANT INFORMATION - DO NOT INSTALL THIS PRODUCT UNLESS YOU HAVE READ ALL OF THE FOLLOWING MATERIAL

By clicking the YES or ACCEPT button below (when applicable), or by installing and using this Product or by having it installed and used on your behalf, You are taking affirmative action to signify that You are entering into a legal agreement and are agreeing to be bound by its terms, EVEN WITHOUT YOUR SIGNATURE. BMC is willing to license this Product to You ONLY if You are willing to accept all of these terms. CAREFULLY READ THIS AGREEMENT. If You DO NOT AGREE with its terms, DO NOT install or use this Product; press the NO or REJECT button below (when applicable) or promptly contact BMC or your BMC reseller and your money will be refunded if by such time You have already purchased a full-use License.

SOFTWARE LICENSE AGREEMENT FOR BMC PRODUCTS

SCOPE. This is a legally binding Software License Agreement (**"License"**) between You (either an individual or an entity) and BMC pertaining to the original computer files (including all computer programs and data stored in such files) contained in the enclosed Media (as defined below) or made accessible to You for electronic delivery, if as a prerequisite to such accessibility You are required to indicate your acceptance of the terms of this License, and all whole or partial copies thereof, including modified copies and portions merged into other programs (collectively, the **"Software"**). **"Documentation"** means the related hard-copy or electronically reproducible technical documents furnished in association with the Software, **"Media"** means the original BMC-supplied physical materials (if any) containing the Software and/or Documentation, **"Product"** means collectively the Media, Software, and Documentation, and all Product updates subsequently provided to You, and **"You"** means the owner or lessee of the hardware on which the Software is installed and/or used. **"BMC"** means BMC Software Distribution, Inc. unless You are located in one of the following regions, in which case "BMC" refers to the following indicated BMC Software, Inc. subsidiary: (i) Europe, Middle East or Africa --BMC Software Distribution, B.V., (ii) Asia/Pacific -- BMC Software Asia Pacific Pte Ltd., (iii) Brazil -- BMC Software do Brazil, or (iv) Japan -- BMC Software K.K. **If You enter into a separate, written software license agreement signed by both You and BMC or your authorized BMC reseller granting to you the rights to install and use this Product, then the terms of that separate, signed agreement will apply and this License is void.**

FULL-USE LICENSE. Subject to these terms and payment of the applicable license fees, BMC grants You this non-exclusive License to install and use one copy of the Software for your internal use on the number(s) and type(s) of servers or workstations for which You have paid or agreed to pay to BMC or your BMC reseller the appropriate license fee. If your license fee entitles You only to a License having a limited term, then the duration of this License is limited to that term; otherwise this License is perpetual, subject to the termination provisions below.

TRIAL LICENSE. If You have not paid or agreed to pay to BMC or your BMC Reseller the appropriate license fees for a full use license, then, **NOTWITHSTANDING ANYTHING TO THE CONTRARY CONTAINED IN THIS LICENSE**: (i) this License consists of a non-exclusive evaluation license ("Trial License") to use the Product for a limited time ("Trial Period") only for evaluation; (ii) during the Trial Period, You may not use the Software for development, commercial, production, database management or other purposes than those expressly permitted in clause (i) immediately above; and (iii) your use of the Product is on an ***AS IS*** basis, and **BMC, ITS RESELLERS AND LICENSORS GRANT NO WARRANTIES OR CONDITIONS (INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE) TO YOU AND ACCEPT NO LIABILITY WHATSOEVER RESULTING FROM THE USE OF THIS PRODUCT UNDER THIS TRIAL LICENSE.** If You use this Product for other than evaluation purposes or wish to continue using it after the Trial Period, you must purchase a full-use license. When the Trial Period ends, your right to use this Product automatically expires, though in certain cases You may be able to extend the term of the Trial Period by request. Contact BMC or your BMC reseller for details.

TERM AND TERMINATION. This License takes effect on the first to occur of the date of shipment or accessibility to You for electronic delivery, as applicable (the **"Product Effective Date"**). You may terminate this License at any time for any reason by written notice to BMC or your BMC reseller. This License and your right to use the Product will terminate automatically with or without notice by BMC if You fail to comply with any material term of this License. Upon termination, You must erase or destroy all components of the Product including all copies of the Software, and stop using or accessing the Software. Provisions concerning Title and Copyright, Restrictions (or Restricted Rights, if You are a U.S. Government entity) or limiting BMC's liability or responsibility shall survive any such termination.

TITLE AND COPYRIGHT; RESTRICTIONS. All title and copyrights in and to the Product, including but not limited to all modifications thereto, are owned by BMC and/or its affiliates and licensors, and are protected by both United States copyright law and applicable international copyright treaties. You will not claim or assert title to or ownership of the Product. To the extent expressly permitted by applicable law or treaty notwithstanding this limitation, You may copy the Software only for backup or archival purposes, or as an essential step in utilizing the Software, but for no other purpose. You will not remove or alter any copyright or proprietary notice from copies of the Product. You acknowledge that the Product contains valuable trade secrets of BMC and/or its affiliates and licensors. Except in accordance with the terms of this License, You agree (a) not to decompile, disassemble, reverse engineer or otherwise attempt to derive the Software's source code from object code except to the extent expressly permitted by applicable law or treaty despite this limitation; (b) not to sell, rent, lease, license, sublicense, display, modify, time share, outsource or otherwise transfer the Product to, or permit the use of this Product by, any third party; and (c) to use reasonable care and protection to prevent the unauthorized use, copying, publication or dissemination of the Product and BMC confidential information learned from your use of the Product. **You will not export or re-export any Product without both the written consent of BMC and the appropriate U.S. and/ or foreign government license(s) or license exception(s).** Any programs, utilities, modules or other software or documentation created, developed, modified or enhanced by or for You using this Product shall likewise be subject to these restrictions. BMC has the right to obtain injunctive relief against any actual or threatened violation of these restrictions, in addition to any other available remedies. Additional restrictions may apply to certain files, programs or data supplied by third parties and embedded in the Product; consult the Product installation instructions or Release Notes for details.

LIMITED WARRANTY AND CONDITION. If You have purchased a Full-Use License, BMC warrants that (i) the Media will be, under normal use, free from physical defects, and (ii) for a period of ninety (90) days from the Product Effective Date, the Product will perform in substantial accordance with the operating specifications contained in the Documentation that is most current at the Product Effective Date. BMC's entire liability and your exclusive remedy under this provision will be for BMC to use reasonable best efforts to remedy defects covered by this warranty

and condition within a reasonable period of time or, at BMC's option, either to replace the defective Product or to refund the amount paid by You to license the use of the Product. BMC and its suppliers do not warrant that the Product will satisfy your requirements, that the operation of the Product will be uninterrupted or error free, or that all software defects can be corrected. This warranty and condition shall not apply if: (i) the Product is not used in accordance with BMC's instructions, (ii) a Product defect has been caused by any of your or a third party's malfunctioning equipment, (iii) any other cause within your control causes the Product to malfunction, or (iv) You have made modifications to the Product not expressly authorized in writing by BMC. No employee, agent or representative of BMC has authority to bind BMC to any oral representations, warranties or conditions concerning the Product. **THIS WARRANTY AND CONDITION IS IN LIEU OF ALL OTHER WARRANTIES AND CONDITIONS. THERE ARE NO OTHER EXPRESS OR IMPLIED WARRANTIES OR CONDITIONS, INCLUDING THOSE OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, REGARDING THIS LICENSE OR ANY PRODUCT LICENSED HEREUNDER. THIS PARAGRAPH SHALL NOT APPLY TO A TRIAL LICENSE.** Additional support and maintenance may be available for an additional charge; contact BMC or your BMC reseller for details.

LIMITATION OF LIABILITY. Except as stated in the next succeeding paragraph, BMC's and your BMC reseller's total liability for all damages in connection with this License is limited to the price paid for the License. **IN NO EVENT SHALL BMC BE LIABLE FOR ANY CONSEQUENTIAL, SPECIAL, INCIDENTAL, PUNITIVE OR INDIRECT DAMAGES OF ANY KIND ARISING OUT OF THE USE OF THIS PRODUCT (SUCH AS LOSS OF PROFITS, GOODWILL, BUSINESS, DATA OR COMPUTER TIME, OR THE COSTS OF RECREATING LOST DATA), EVEN IF BMC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Some jurisdictions do not permit the limitation of consequential damages so the above limitation may not apply.

INDEMNIFICATION FOR INFRINGEMENT. BMC will defend or settle, at its own expense, any claim against You by a third party asserting that your use of the Product within the scope of this License violates such third party's patent, copyright, trademark, trade secret or other proprietary rights, and will indemnify You against any damages finally awarded against You arising out of such claim. However, You must promptly notify BMC in writing after first receiving notice of any such claim, and BMC will have sole control of the defense of any action and all negotiations for its settlement or compromise, with your reasonable assistance. BMC will not be liable for any costs or expenditures incurred by You without BMC's prior written consent. If an order is obtained against your use of the Product by reason of any claimed infringement, or if in BMC's opinion the Product is likely to become the subject of such a claim, BMC will at its option and expense either (i) procure for You the right to continue using the product, or (ii) modify or replace the Product with a compatible, functionally equivalent, non-infringing Product, or (iii) if neither (i) nor (ii) is practicable, issue to You a pro-rata refund of your paid license fee(s) proportionate to the number of months remaining in the 36 month period following the Product Effective Date. This paragraph sets forth your only remedies and the total liability to You of BMC, its resellers and licensors arising out of such claims.

GENERAL. This License is the entire understanding between You and BMC concerning this License and may be modified only in a mutually signed writing between You and BMC. If any part of it is invalid or unenforceable, that part will be construed, limited, modified, or, severed so as to eliminate its invalidity or unenforceability. This License will be governed by and interpreted under the laws of the jurisdiction named below, without regard to conflicts of law principles, depending on which BMC Software, Inc. subsidiary is the party to this License: (i) BMC Software Distribution, Inc. - the State of Texas, U.S.A., (ii) BMC Software Distribution, B.V. - The Netherlands, (iii) BMC Software Asia Pacific Pte Ltd. -- Singapore (iv) BMC Software do Brazil -- Brazil, or (v) BMC Software K.K. -- Japan. Any person who accepts or signs changes to the terms of this License promises that they have read and understood these terms, that they have the authority to accept on your behalf and legally obligate You to this License. Under local law and treaties, the restrictions and limitations of this License may not apply to You; You may have other rights and remedies, and be subject to other restrictions and limitations.

U.S. GOVERNMENT RESTRICTED RIGHTS. UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in FAR Section 52.227-14 Alt. III (g)(3), FAR Section 52.227-19, DFARS 252.227-7014 (b) or DFARS 227.7202, 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.

Notes

100042340