Contacting BMC Software

Several methods are available for contacting BMC Software.

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

United States and Canada

<table>
<thead>
<tr>
<th>Address</th>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC SOFTWARE INC</td>
<td>1 713 918 8800</td>
<td></td>
</tr>
<tr>
<td>2103 CITYWEST BLVD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HOUSTON TX 77042-2827 USA</td>
<td>or 1 800 841 2031</td>
<td></td>
</tr>
</tbody>
</table>

Outside United States and Canada

<table>
<thead>
<tr>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>+01 713 918 8800</td>
<td>+01 713 918 8000</td>
</tr>
</tbody>
</table>

© Copyright 2006-2016 BMC Software, Inc.

BMC, BMC Software, the BMC logo, the BMC Software logo, and other BMC marks are the exclusive properties of BMC Software, Inc. and are registered or may be registered in the U.S. and in other countries. BladeLogic and other BladeLogic marks are the exclusive properties of BladeLogic, Inc. and are registered or may be registered in the U.S. and in other countries. All other trademarks or registered trademarks are the property of their respective owners.

Explorer is a trademark or registered trademark of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

CICS, DB2, IBM, IBM z, IMS, MVS, NetView, OS/390, RACF, RETAIN, VTAM, WebSphere, and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Linux is the registered trademark of Linus Torvalds.

The information included in this documentation is the proprietary and confidential information of BMC Software, Inc., its affiliates, or licensors. Your use of this information is subject to the terms and conditions of the applicable End User License agreement for the product and to the proprietary and restricted rights notices included in the product documentation.

Restricted rights legend

U.S. Government Restricted Rights to Computer Software. UNPUBLISHED—RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure of any data and computer software by the U.S. Government is subject to restrictions, as applicable, set forth in FAR Section 52.227-14, DFARS 252.227-7013, DFARS 252.227-7014, DFARS 252.227-7015, and DFARS 252.227-7025, as amended from time to time. Contractor/Manufacturer is BMC SOFTWARE INC, 2103 CITYWEST BLVD, HOUSTON TX 77042-2827, USA. Any contract notices should be sent to this address.
Customer support

Support website
You can obtain technical support from BMC 24 hours a day, 7 days a week at http://www.bmc.com/support. From this website, you can:

- Read overviews about support services and programs that BMC offers
- Find the most current information about BMC products
- Search a database for problems similar to yours and possible solutions
- Order or download product documentation
- Download products and maintenance
- Report a problem or ask a question
- Subscribe to receive proactive e-mail alerts
- Find worldwide BMC support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

Support by telephone or e-mail
In the United States and Canada, if you need technical support and do not have access to the web, call 1 800 537 1813 or send an e-mail message to customer_support@bmc.com. (In the subject line, enter SupID:yourSupportContractID, such as SupID:12345). Outside the United States and Canada, contact your local support center for assistance.

Before contacting BMC
Have the following information available so that Customer Support can begin working on your issue immediately:

- Product information
  - Product name
  - Product version (release number)
  - License number and password (trial or permanent)

- Operating system and environment information
  - Machine type
  - Operating system type, version, and service pack or other maintenance level such as PUT or PTF
  - System hardware configuration
  - Serial numbers
  - Related software (database, application, and communication) including type, version, and service pack or maintenance level

- Sequence of events leading to the problem
- Commands and options that you used
- Messages received (and the time and date that you received them)
  - Product error messages
  - Messages from the operating system
  - Messages from related software
License key and password information

If you have questions about your license key or password, contact Customer Support through one of the following methods:

■ Send an e-mail message to customer_support@bmc.com. (In the Subject line, enter SupID:yourSupportContractID, such as SupID:12345.)

■ In the United States and Canada, call 1 800 537 1813. Outside the United States and Canada, contact your local support center for assistance.

Contents

About this book ................................................................. 11
Related publications .................................................................................. 11
Conventions .......................................................................................... 12

Chapter 1 Overview of MainView AutoOPERATOR 13
How MainView AutoOPERATOR applications are organized .............. 13
  Administration applications ...................................................... 15
Accessing MainView AutoOPERATOR .............................................. 17
How the documentation for MainView AutoOPERATOR is organized .... 19

Chapter 2 Automation with Rules 21
What the Rule Processor is ............................................................... 21
  What events are ........................................................................... 22
  What a Rule is ............................................................................. 22
  Where Rules are kept: Rule Sets ............................................... 24
  What actions a Rule can take ..................................................... 24
How to create a Rule ........................................................................ 25
How to begin identifying events for automation .................................. 29
How to manage and implement basic automation with Rules .............. 30
  Grouping Rules into Rule Sets ................................................ 30
  What automation strategy is .................................................... 31
  Why automation strategy is important ..................................... 31
How to qualify displays in the Rule Processor applications .................. 32
Where to go from here ..................................................................... 33

Chapter 3 Describing events 35
What events are .............................................................................. 35
Using the Event Activity Statistics application .................................. 41
  Viewing event activity statistics ............................................ 42
  Describing fields ...................................................................... 47
Special event types .......................................................................... 48
  BRC ....................................................................................... 49
  CICa .................................................................................... 50
  DB2 ..................................................................................... 51
  EOM .................................................................................... 51
  EOS ..................................................................................... 52
Chapter 4  Using the Rule Processor Detail Control panel 63
What the Rule Processor Detail Control panel is .................................................................63
Accessing the Rule Processor Detail Control panel .......................................................66
Rule Processor Detail Control panel ................................................................................71
  Primary commands ................................................................................................. 73
  Fields description .....................................................................................................75
Criteria match rate thresholds ..........................................................................................78
  Advanced Rule processing techniques ...........................................................................80
Activating and saving modified Rules ............................................................................80

Chapter 5  Creating Rules 83
Overview of creating Rules ...............................................................................................83
Using primary commands .................................................................................................84
Step 1. Using Selection Criteria fields ..............................................................................85
Step 2. Specifying variables as additional selection criteria ...........................................86
  Using AND/OR logic on the Variable Dependencies panel ........................................88
  Using the Advanced Variable Dependencies (ASV) panel ........................................88
Step 3. Specifying a Rule's actions with the Action Specification fields ....................90
  Additional panels to enter EXECs, commands, and messages ...................................91
Step 4. Setting variables as additional action specification ...........................................91
  Using the Action: Advanced Set Variables (AAV) panel ..........................................92
Step 5. Creating ALERTs as additional action specification ........................................94
  Creating ALERTs for all event types except ALRT events ......................................95
  Creating ALERTs for the ALRT event type ..............................................................95
  Specifying a Final Action for an ALERT ...............................................................96
  (optional) Modifying an ALERT in a Rule: using the MODIFY function ..........100
  (optional) Step 6. Sending events to a BMC Impact Manager cell ..............................102
  (optional) Step 7. Enriching ALERTs that are sent to a BMC Impact Manager cell ........103
Chapter 6  Examples of creating Rules  

Creating a Rule to suppress a message ......................................................... 107
  Example 1: creating a suppression Rule ................................................. 107
  Example 2: creating a suppression Rule with variables ...................... 115
Creating TIME-initiated Rules .................................................................... 119
  Example 1: creating simple TIME-initiated Rules .............................. 119
  Example 2: creating a TIME–initiated Rule with ATSTART ............... 121
Creating more complex Rules .................................................................... 124
  Example 1: setting a variable with a Rule .............................................. 124
  Example 2: creating a Rule that fires when a variable value is met ....  126
  Using the Rules .................................................................................... 130
Creating ALERTs from Rules ...................................................................... 130
  Example 1: creating an ALERT retained across BBI-SS PAS restarts and
  MVS IPLs ............................................................................................ 130
  Example 2: creating an ALERT whose priority increases over time ....  133
Creating a VAR–initiated Rule ................................................................... 136
Creating a Rule that sends event and ALERT data to BMC Impact Manager 138
Creating a Rule for MLWTO messages ...................................................... 143
  Example 1: creating a Rule where MLWTO minor = ALL ................. 143
  Example 2: creating a Rule where MLWTO minor = SINGLE .......... 145
Creating a Rule for IMS MLSEG messages .............................................. 147
  Example 1: creating a Rule with MLSEG Minor = ALL ................... 147
  Example 2: creating a Rule where MLSEG Minor = SINGLE .......... 150
Creating a Rule for CICS multiline messages ........................................... 151

Chapter 7  Creating more powerful Rules .................................................. 155

Over of creating more powerful Rules ...................................................... 155
Using pattern matching on Selection Criteria panels ............................... 155
  Examples of using pattern matching ................................................... 156
Using variables in the Rule Processor ...................................................... 160
  Defining variables ............................................................................. 160
  Restrictions for long variables ......................................................... 161
  Resolving variables ........................................................................ 161
  Using variables in Rule Processor application fields ....................... 164
  Variable modifiers .......................................................................... 166
Using special-use variables on the Variable Dependencies panel ............ 168
List of MainView AutoOPERATOR-supplied EVENT variables .............. 169
  Event variables and slot names for BRC events ....................... 181
  Converting slots to event variables ................................................ 182
List of variables for ALRT-initiated Rules .............................................. 183
### List of variables for HWTO-initiated Rules ................................. 184
### Using Rules functions ........................................................................................................... 185
  - Understanding syntax ........................................................................................................... 185
### List of built-in Rules functions .......................................................................................... 189

#### Chapter 8 Managing Rules and automation using the Automation Control panel 207
- What the Automation Control panel is .............................................................. 207
- Setting automation status/strategy (upper portion) ........................................... 209
  - Using primary commands ......................................................................................... 210
- Viewing automation statistics (middle portion) ...................................................... 213
- Grouping Rules into Rule Sets (bottom portion) ....................................................... 214
  - Using line commands .............................................................................................. 217
- Choosing an automation strategy ............................................................................... 218
  - Selecting ALL ........................................................................................................... 218
  - Selecting FIRST ..................................................................................................... 218
  - Selecting INDIVIDUAL .......................................................................................... 219
- Rule Set filter .............................................................................................................. 219
  - Filtering events for Rule Sets .................................................................................. 219
  - Enabling filtering and Rule Set match rate for Rule Sets .................................... 221

#### Chapter 9 Using the Rule Set Overview panel to manage Rules in Rule Sets 229
- What the Rule Set Overview panel is .............................................................. 229
- Using primary commands ......................................................................................... 235
- Using line commands .............................................................................................. 235
- Copying or moving Rules ......................................................................................... 237
  - Copying or moving Rules within Rule Set ............................................................ 237
  - Copying or moving Rules between Rule Sets ...................................................... 238

#### Chapter 10 Rules Management application 241
- Benefits of using a Rules registry ........................................................................... 241
- Rules Management editor ....................................................................................... 241
- Rules .......................................................................................................................... 242
- Deployment and AOPlexes ....................................................................................... 242
- Accessing the Rules Management application ....................................................... 243
- Creating a PAS definition ......................................................................................... 245
- Creating a valid PAS list ............................................................................................ 246
- Creating a setbase ...................................................................................................... 248
- Creating a Rule Set .................................................................................................... 249
- Creating a Rule Pool .................................................................................................. 251
- Creating a Rule ........................................................................................................... 252
- Editing a Rule ............................................................................................................ 257
About this book

The *MainView AutoOPERATOR Basic Automation Guide*, Volume 1: Using Rules is for operators and system programmers who need to perform basic automation tasks in the data center using the MainView AutoOPERATOR Rules Processor.

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

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

Related publications

From the BMC Support Central website, you can use the following methods to access related publications that support your product or solution:

- Link to the BMC Documentation Center (https://webapps.bmc.com/infocenter/index.jsp) to browse documentation sets.

- View Quick Course videos (short overviews of selected product concepts, tasks, or features), which are available from the following locations:
  - Documentation Center
  - Support Central (at http://www.bmc.com/support/mainframe-demonstrations)
  - BMC Mainframe YouTube channel (https://www.youtube.com/user/BMCSoftwareMainframe)

- View individual product documents (books and notices) within the “A – Z Supported Product List.”

You can order hardcopy documentation from your BMC sales representative or from the support site. You can also subscribe to proactive alerts to receive e-mail messages when notices are issued.
Conventions

This document uses the following special conventions:

- All syntax, operating system terms, and literal examples are presented in this typeface.

- Variable text in path names, system messages, or syntax is displayed in italic text: `testsystestsys/instance/fileName`

- Menu sequences use a symbol to convey the sequence. For example, `Actions => Create Test` instructs you to choose the `Create Test` command from the `Actions` menu.

Tip

You can access the BMC Support Central site at http://www.bmc.com/support.
Overview of MainView AutoOPERATOR

MainView AutoOPERATOR is a BMC Software product that provides tools and applications to improve an operator’s ability to provide high system availability. MainView AutoOPERATOR helps the operator through error minimization, improved productivity, and problem determination and prevention.

How MainView AutoOPERATOR applications are organized

MainView AutoOPERATOR allows operators to interact with many other applications that might be running on your system.
Figure 1 on page 14 illustrates some of the areas that MainView AutoOPERATOR communicates with.

Figure 1: MainView AutoOPERATOR applications overview

MainView AutoOPERATOR applications use an online ISPF-like dialog interface (see Figure 2 on page 14).

Figure 2: MainView AutoOPERATOR primary option panel

Topics in this section briefly describe the applications that you can access from the MainView AutoOPERATOR primary option panel and where you can find additional documentation.
Administration applications

The administration applications are divided into the following categories:

- Basic automation
- Advanced automation
- Dynamic Parameter Manager

Basic Automation

The basic automation applications consists of the following MainView AutoOPERATOR applications:

- Rule Processor
- windows mode performance and history tools
- object management with MainView Total Object Manager

Rule Processor

The Rule Processor applications provide a panel-driven interface that you can use to create Rules that automate events on your system. The MainView AutoOPERATOR Basic Automation Guide, Volume 1: Using Rules documents the following topics:

- introduction to Rules
- defining events
- identifying events for automation
- creating Rules
- managing Rules and automation
- managing Rules in Rule Sets

Windows mode performance and history tools

MainView AutoOPERATOR provides two windows mode tools that allow you to view and track how well your automation efforts are performing:

- Automation Reporter: enables you to view in real-time and record historical data about automation; for more information, see the MainView AutoOPERATOR Basic Automation Guide, Volume 2.
- Automation Logger: enables you to view real-time data about Rules and events on your system; for more information, see the *MainView AutoOPERATOR Basic Automation Guide, Volume 2*.

### Object management

MainView AutoOPERATOR includes the MainView Total Object Manager component, which provides a full-featured application that allows you to manage the status of Started Tasks (also referred to as objects), create dependencies between Started Tasks, and automate complex schedules for object availability.

Refer to the *MainView Total Object Manager User Guide* for more information.

### 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.

For more information about the EXEC Management application, see the *MainView AutoOPERATOR Basic Automation Guide Volume 2*.

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

### TIMEXEC: Time-Initiated EXECs

The Time-Initiated EXECs application manages time-initiated EXEC requests by using the CALLX service to schedule your specified EXEC based on time.

For more information about using the Time-Initiated EXECs application, see the *MainView AutoOPERATOR Basic Automation Guide Volume 2*.

### Open System Procedural Interface

The Open System Procedural Interface application is described in the *MainView AutoOPERATOR Advanced Automation Guide*. This interface allows you to capture screen images and keystrokes in an EXEC so that you can programmatically interact with any IBM VTAM application.
Dynamic Parameter Manager

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

<table>
<thead>
<tr>
<th>BBPARM member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAOALSxx</td>
<td>contains MainView AutoOPERATOR ALERT parameters</td>
</tr>
<tr>
<td>AAOARPxx</td>
<td>contains MainView AutoOPERATOR Automation Reporter parameters</td>
</tr>
<tr>
<td>AAOEXPxx</td>
<td>contains MainView AutoOPERATOR EXEC parameters</td>
</tr>
<tr>
<td>AAOPRMxx</td>
<td>contains MainView AutoOPERATOR general parameters</td>
</tr>
<tr>
<td>AAOGMExx</td>
<td>contains General Messages Exchange parameters</td>
</tr>
<tr>
<td>AAOTSPxx</td>
<td>contains TapeSHARE parameters</td>
</tr>
</tbody>
</table>

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

## Accessing MainView AutoOPERATOR

The MainView Selection Menu is displayed when you access MainView from a 3270 display.

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 the *MainView User Guide* for a description of these functions.)

### To access the MainView AutoOPERATOR primary option menu

1. Select **option A — Automated Operations** from the MainView Selection Menu (Figure 3 on page 17).

   **Figure 3: Example of the MainView Selection Menu**

   ![Figure 3: Example of the MainView Selection Menu](image-url)
The MainView AutoOPERATOR primary option menu is displayed (Figure 4 on page 18). From it, you can access the Operator Workstations applications, the Automation applications (both Basic and Advanced), and the Dynamic Parameter Manager application.

Figure 4: Example of the MainView AutoOPERATOR primary option menu

2 Select option 8 — AUTOMATION to display the AUTOMATION MENU (Figure 5 on page 18).

Figure 5: Example of the MainView AutoOPERATOR AUTOMATION MENU
Refer to “How MainView AutoOPERATOR applications are organized” on page 13 for information about the applications shown in these figures.

How the documentation for MainView AutoOPERATOR is organized

The following table describes how the documentation library for MainView AutoOPERATOR is organized.

This table is included because descriptions for the applications from the MainView AutoOPERATOR menu are located in different books within the MainView AutoOPERATOR product library.

<table>
<thead>
<tr>
<th>For documentation about</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MainView AutoOPERATOR Primary Option Menu: Options 1 through 7</strong></td>
<td></td>
</tr>
<tr>
<td>Option 1: ALERT Management</td>
<td>For information about using the MainView AutoOPERATOR ALERTs application, see the <em>MainView AutoOPERATOR Basic Automation Guide Volume 2</em>. The following lists other places in the MainView AutoOPERATOR library that discuss ALERTs:</td>
</tr>
<tr>
<td></td>
<td>- For information about how to create ALERTs from EXECs, see the <em>MainView AutoOPERATOR Advanced Automation Guide</em>.</td>
</tr>
<tr>
<td></td>
<td>- Creating Rules on page 83 that discusses creating rules and provides information about how to create ALERTs from Rules</td>
</tr>
<tr>
<td>Option 7: WebSphere MQ</td>
<td><em>MainView AutoOPERATOR Options User Guide</em>, which describes the workstation tools that allow you to automate routine tasks and monitoring for these resources.</td>
</tr>
<tr>
<td>Options 2, 3, 4, 5, and 6: MVS Resources, CICS Resources, IMS Resources, NetView and TapeSHARE Resources</td>
<td><em>MainView AutoOPERATOR for WebSphere MQ Installation and User Guide</em></td>
</tr>
</tbody>
</table>

Administration Automation: Option 8—Basic Automation, Options 1 though 4
<table>
<thead>
<tr>
<th>For documentation about</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>■ Option 1: Event Activity Statistics: managing events such as messages,</td>
<td>See the chapters “Automation with Rules” on page 21 through “Using the Rule Set Overview panel to manage Rules in Rule Sets” on page 229.</td>
</tr>
<tr>
<td>commands or AutoOPERATOR ALERTS</td>
<td></td>
</tr>
<tr>
<td>■ Option 2: Rules and Rule Sets: defining AutoOPERATOR Rules allows you quickly</td>
<td></td>
</tr>
<tr>
<td>and easily set up conditions where actions and responses to events are automated</td>
<td></td>
</tr>
<tr>
<td>Option 3: Continuous State Manager</td>
<td>The section in the MainView AutoOPERATOR Solutions Guide that discusses managing Critical Applications with Continuous State Manager</td>
</tr>
<tr>
<td>Option 4: Object Management</td>
<td>the MainView Total Object Manager User Guide</td>
</tr>
<tr>
<td>Administration Automation: Option 8—Advanced Automation, Options 7 through 9</td>
<td></td>
</tr>
<tr>
<td>Option 7: Display/Modify EXEC Status</td>
<td>For information about using the MainView AutoOPERATOR EXEC Management applications, see the MainView AutoOPERATOR Basic Automation Guide Volume 2.</td>
</tr>
<tr>
<td>Option 8: Time-Initiated EXEC Requests</td>
<td>For information about using the MainView AutoOPERATOR Time-initiated EXECs, see the MainView AutoOPERATOR Basic Automation Guide Volume 2.</td>
</tr>
<tr>
<td>Option 9: Open Systems Procedural Interface (OSPI)</td>
<td>The topic in the MainView AutoOPERATOR Advanced Automation Guide that discusses using OSPI</td>
</tr>
<tr>
<td>MainView AutoOPERATOR Primary Option Menu: Option 9</td>
<td></td>
</tr>
<tr>
<td>Option 9: Parms</td>
<td>MainView AutoOPERATOR Customization Guide</td>
</tr>
</tbody>
</table>
Automation with Rules

Read this chapter to learn basic concepts and terms to help you write Rules that perform automation on your system. For information about creating Rules for IBM WebSphere MQ events, refer to the MainView AutoOPERATOR for WebSphere MQ 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 MainView AutoOPERATOR ALERTs based on events

In addition, if you have the BMC Impact Integration for MainView installed, you can send events from MainView AutoOPERATOR to a BMC Impact manager cell from a Rule. Refer to the MainView AutoOPERATOR BMC Impact Integration for z/OS User Guide for more information.

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

- Events, “What events are” on page 22 and “Describing events” on page 35
- Rules, “What a Rule is” on page 22 and “Creating Rules” on page 83
- Selection criteria, “What a Rule is” on page 22 and “Step 1. Using Selection Criteria fields” on page 85
- Action specifications, “What a Rule is” on page 22 and “Step 3. Specifying a Rule’s actions with the Action Specification fields” on page 90
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 depending on where they originated in the system. Some examples of events that MainView AutoOPERATOR supports are IBM MVS write-to-operator messages (WTOs), IBM CICS transient data queue (TDQ) messages, BBI-SS PAS Journal messages, and MVS commands. MainView AutoOPERATOR regards each of these events as a different event type, and MainView AutoOPERATOR supports a total of 14 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, MainView AutoOPERATOR provides separate panels for each event type where only the applicable attributes are displayed.

MainView 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 P.M. to 3:00 P.M.

For information about how your automation with time-initiated Rules can be affected when you change the MVS Local Time to observe daylight savings time, refer to “Field descriptions for the Rule Processor” on page 279.

Which event types are available for automation on your system depends on your configuration of MainView AutoOPERATOR products. Refer to “Describing events” on page 35 for more 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” on page 23 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 85.

**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 the Rule fires, the user-specified automation actions are performed. This action 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 versions the action specification panels that contain all the applicable actions that a Rule can perform for the event. For more information about action specifications, refer to “Step 3. Specifying a Rule's actions with the Action Specification fields” on page 90.

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.

---

**Tip**

When specifying action specifications for a Rule, you should ensure that the action specification does not create events that re-trigger the original Rule. This type of error results in a loop and the Rules might fire endlessly.

To prevent this situation from happening, it is recommended that you set a limit for how many a times a Rule fires. “Criteria match rate thresholds” on page 78 describes how you can limit the number of times a Rule fires and therefore, prevent such loops.
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.

You can edit a Rule within a disabled Rule Set without having the Rule fire during the editing process. When you select a Rule for editing from a disabled Rule Set, the status of the Rule Set appears as LOADED on the Automation Control panel.

For more information about Rule Sets, refer to

- “Grouping Rules into Rule Sets” on page 30
- “Managing Rules and automation using the Automation Control panel” on page 207

What actions a Rule can take

Some of the actions that a Rule can take are

- suppress a message from appearing on the console
- create ALERTs in response to a system event
- execute MVS, IBM IMS, BBI-SS PAS, CICS, IMSPlex, or IBM NetView commands at predefined times with TIME-initiated Rules
- set SHARED variables
- send messages to TSO users
- issue WTO messages
How to create a Rule

You can select events for which to create Rules by examining the flow of events on your system or by selecting events from the MainView AutoOPERATOR application -- Event Activity Statistics.

For more information about selecting events from the Event Activity Statistics application, refer to “Using the Event Activity Statistics application” on page 41.
Regardless of how you select events for automation, Rules are created by using panels in the Rule Processor. A series of panels are displayed; refer to Figure 6 on page 26.

Figure 6: Sequence of Rule Processor Detail 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 specification 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 and the Action Specification panel.

Table 1 on page 27 provides a brief synopsis of each of the six panels that you can use while creating a Rule.

### Table 1: Description of panels used to create a Rule

<table>
<thead>
<tr>
<th>Panel name</th>
<th>Allows you to</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Processor Detail Control</td>
<td>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</td>
<td>“What the Rule Processor Detail Control panel is” on page 63</td>
</tr>
</tbody>
</table>
| Selection Criteria          | specify the conditions and attributes the event must match before the Rule will fire  
Although some criteria 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. | “Step 1. Using Selection Criteria fields” on page 85                                                                                                                                                      |
| 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. | “Step 2. Specifying variables as additional selection criteria” on page 86                                                                                                                                |
<table>
<thead>
<tr>
<th>Panel name</th>
<th>Allows you to</th>
<th>See page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action Specification(s) I and II</td>
<td>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.</td>
<td>“Step 3. Specifying a Rule's actions with the Action Specification fields” on page 90</td>
</tr>
<tr>
<td>Alert Action(s) I and Alert Action(s) II</td>
<td><em>(optional)</em> create an MainView AutoOPERATOR ALERT from a Rule The Rule Processor has two panels that you can use to create an MainView 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 5. Creating ALERTs as additional action specification” on page 94 for more information.</td>
<td>“Step 5. Creating ALERTs as additional action specification” on page 94</td>
</tr>
<tr>
<td>Event Action —BEIM panel</td>
<td><strong>Note:</strong> This panel appears only if you have installed the BMC Impact Integration for z/OS product and specified PRODUCT=IIZ in BBPARM member BBISSP00. Specify whether event data should be sent to the BMC IM (BMC Event Manager) cell. Any event type for which you can write a Rule can have their event data sent to BMC IM. <strong>Note:</strong> The only exceptions are the WebSphere MQ non-event messages. WebSphere MQ instrumentation events can have their event data sent to BMC IM.</td>
<td>“(optional) Step 6. Sending events to a BMC Impact Manager cell” on page 102</td>
</tr>
</tbody>
</table>
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 for which to write Rules or for when you want to see the frequency of events on your system. Refer to “Using the Event Activity Statistics application” on page 41 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 MainView AutoOPERATOR Rules.

For these events, you immediately can begin creating Rules with the Rule Processor Detail Control panel. Refer to “Using the Rule Processor Detail Control panel” on page 63 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 explains 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 might be disabled and thus not performing any automation tasks.

When you plan which events you want to automate, you might 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 218 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.
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

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 ------- MAINVIEW AutoOPERATOR
COMMAND ===> TGT ===> SYSA
Activated at: YY-MMM-DD  10:06:24 Total Events:  52     DATE --- DD/MM/YY
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 can also use the wildcard characters asterisk (*) or plus sign (+). Refer to “Using pattern matching on Selection Criteria panels” on page 155 for more information about how to use wildcard characters.

---

**Where to go from here**

The next several chapters are organized sequentially, based on the following major subjects:

<table>
<thead>
<tr>
<th>To</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>learn about events</td>
<td>“Describing events” on page 35 defines what events are and which MainView AutoOPERATOR product components you need to have for each event type.</td>
</tr>
<tr>
<td>see a display of all the events eligible for automation</td>
<td>“Using the Event Activity Statistics application” on page 41 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 from the panel to automate.</td>
</tr>
<tr>
<td>To create Rules with the Rule Processor applications</td>
<td>See</td>
</tr>
<tr>
<td>-----------------------------------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>“Using the Rule Processor Detail Control panel” on page 63 describes how to use the initial panel in creating a Rule.</td>
<td></td>
</tr>
<tr>
<td>“Examples of creating Rules” on page 107 describes the additional panels you can use to create Rules.</td>
<td></td>
</tr>
<tr>
<td>“Examples of creating Rules” on page 107 describes the Selection Criteria and Action Specification panels for each event type, sorted alphabetically by event type.</td>
<td></td>
</tr>
<tr>
<td>“Creating more powerful Rules” on page 155 describes how to use pattern matching and variables to create more powerful and flexible Rules.</td>
<td></td>
</tr>
</tbody>
</table>

Although basic principles exist for creating a Rule, actually creating Rules means that you have to specify information on several panels. For this reason, several chapters cover this material.

<table>
<thead>
<tr>
<th>To manage your Rules and automation</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Managing Rules and automation using the Automation Control panel” on page 207</td>
<td></td>
</tr>
<tr>
<td>“Using the Rule Set Overview panel to manage Rules in Rule Sets” on page 229</td>
<td></td>
</tr>
</tbody>
</table>

These chapters describe how you can implement and control your Rules and Rule Sets.
Describing events

This chapter describes events, the Event Activity Statistics application, and special event types.

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 events that are supported by MainView AutoOPERATOR are MVS write-to-operator messages (WTOs), CICS transient data queue (TDQ) messages, BBI-SS PAS Journal messages, and MVS commands. MainView AutoOPERATOR regards each of these events as a different event type and MainView AutoOPERATOR supports a total of 13 system-based events.

Which event types are available for automation on your system depends on your configuration of MainView AutoOPERATOR options and other BMC Software products. To help you determine which events might be available on your system, Table 2 on page 36 describes

- the event
- the origin or cause of the event
- the product option required for the event type to be eligible for automation
- whether statistics are available for the event in the Event Activity Statistics application

Note

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 “Special event types” on page 48.

Table 2: Product requirements for event availability

<table>
<thead>
<tr>
<th>Event type</th>
<th>Caused by</th>
<th>Product or option required</th>
<th>Event statistics available?</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALRT</td>
<td>the creation of an MainView AutoOPERATOR ALERTs MainView AutoOPERATOR ALERTs can be created by an MainView AutoOPERATOR EXEC or by the Rule Processor.</td>
<td>any MainView AutoOPERATOR</td>
<td>Yes</td>
</tr>
<tr>
<td>ALRM</td>
<td>alarm messages created by the MainView Alarm Manager</td>
<td>any MainView AutoOPERATOR</td>
<td>Yes</td>
</tr>
<tr>
<td>BRC</td>
<td>BAROC messages sent from BMC products</td>
<td>BMC Impact Integration for z/OS product and specified PRODUCT=IIZ in BBPARM member BBISSP00</td>
<td>No</td>
</tr>
<tr>
<td>CICS</td>
<td>CICS Intrapartition or Extrapartition Transient Data Messages These messages are processed by the MainView AutoOPERATOR for CICS Transient Data Program Exit CMRTDPXn, which runs when a message is passed through the XTDOUT exit in CICS Transaction Server. Restrictions: CICS can generate multiline messages queued to the transient data queue (application programs typically cannot do this generation). 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. Therefore 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 Rule Processor. For example, you cannot suppress a part of any multiline CICS message and still receive the rest of the multiline message in a Rule.</td>
<td>MainView AutoOPERATOR for CICS</td>
<td>Yes</td>
</tr>
<tr>
<td>Event type</td>
<td>Caused by</td>
<td>Product or option required</td>
<td>Event statistics available?</td>
</tr>
<tr>
<td>------------</td>
<td>-----------</td>
<td>----------------------------</td>
<td>-----------------------------</td>
</tr>
</tbody>
</table>
| CICa       | CICS Intrapartition or Extrapartition Multiline messages  
The CICa event type is a subset of the CICS event type. CICa events appear on Rule Processor panels when the selection criteria for a CICS-initiated Rule specifies MLTDQ = ALL.  
For more information:  
- “Selection Criteria fields” on page 279  
- “CICa” on page 50  
- “Creating a Rule for CICS multiline messages” on page 151 | MainView  
AutoOPERATOR for CICS | Yes |
| CMD        | commands processed by the MVS subsystem interface | MainView  
AutoOPERATOR for z/OS | Yes |
| DB2        | IBM DB2 generated messages  
Messages are sent from a DB2 subsystem. Refer to “Special event types” on page 48 for more information about this event type. | MainView  
AutoOPERATOR for DB2 | Yes |
| EOM        | End-of-Memory (EOM) event that occurs after an address space ends  
An address space ends when a Started Task (STC) or TSO User (TSU) is stopped, canceled, or ends abnormally (abends). An address space does not end when a batch job terminates. It ends when the initiator is terminated. When a TSO user ID is canceled, the EOM event shows the TSO user session as NORMAL termination (instead of the ABNORMAL termination) because TSO recovers from the cancel request and does a normal logoff to end the session. | MainView  
AutoOPERATOR for z/OS | Yes |
| EOS        | End-of-Step (EOS) event that occurs whenever an address space changes steps | MainView  
AutoOPERATOR for z/OS | Yes |
| EXT        | externally generated messages  
External messages directed at the BBI-SS PAS using SUBSYS= parameter on a DD statement in JCL. Refer to “Special event types” on page 48 for more information about this event type. | any MainView  
AutoOPERATOR | Yes |
<table>
<thead>
<tr>
<th>Event type</th>
<th>Caused by</th>
<th>Product or option required</th>
<th>Event statistics available?</th>
</tr>
</thead>
</table>
| HWTO       | Historical WTO events  
Early WTOs and interim WTOs that were issued while MainView AutoOPERATOR was inactive  
For early HWTOs, you must have the MVS EXIT CNZ_MsgToSyslog exit. For more information, see the *MainView AutoOPERATOR Customization Guide*. | BMC MainView Automation Solution | No |
| IMP        | IMS type-2 (IMSplex) commands  
An OM Input user exit sends these commands before the IMS OM address space processes them.  
The requirements to receive these events are as follows:  
- The IMS OM address space runs on the same z/OS image as the MainView AutoOPERATOR PAS.  
- You must specify the MainView AutoOPERATOR OM Input Exit (BMCAOOM) in the OM exit routine definitions list in the OM Proclib member: for example:  
  EXITDEF=(TYPE=INPUT,EXITS=(BMCAOOM),COMP=OM).  
- Include the MainView AutoOPERATOR load library in the STEPLIB of the IMS OM started task JCL. | MainView AutoOPERATOR for IMS | Yes |
| 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 |
<table>
<thead>
<tr>
<th>Event type</th>
<th>Caused by</th>
<th>Product or option required</th>
<th>Event statistics available?</th>
</tr>
</thead>
</table>
| IMSa       | IMS multisegment MTO messages (MLSEG)  
The IMSa event type is a subset event type of the IMS event type. IMSa events appear on Rule Processor panels when the selection criteria for a Rule initiated by IMS specifies MLSEG Minor = ALL.  
For more information:  
- “Selection Criteria fields” on page 279  
- “IMSa” on page 56  
- “Example 1: creating a Rule with MLSEG Minor = ALL” on page 147 | MainView AutoOPERATOR for IMS | Yes |
| IMSs       | IMS multisegment MTO messages (MLSEG)  
The IMSs event type is a subset event type of the IMS event type. IMSs events appear on Rule Processor panels when the selection criteria for a Rule initiated by IMS specifies MLSEG Minor = SINGLE.  
For more information:  
- “Selection Criteria fields” on page 279  
- “IMSs” on page 57  
- “Example 2: creating a Rule where MLSEG Minor = SINGLE” on page 150 | MainView AutoOPERATOR for IMS | Yes |
| JRNL       | BBI Journal generated messages  
These messages come from BBI products and the IMFEXEC MSG statement. | any MainView AutoOPERATOR | Yes |
| MSG        | WTOs and WTORs  
These messages originated as MVS WTOs (write-to-operator) and WTORs (write-to-operator with response). | MainView AutoOPERATOR for z/OS | Yes |
<table>
<thead>
<tr>
<th>Event type</th>
<th>Caused by</th>
<th>Product or option required</th>
<th>Event statistics available?</th>
</tr>
</thead>
</table>
| MSGa       | multiline WTO (MLWTO) The MSGa event type is a subset event type of the MSG event type. MSGa events appear on Rule Processor panels when the selection criteria for a MSG-initiated Rule specifies MLWTO Minor = ALL. For more information:  
   - “Selection Criteria fields” on page 279  
   - “MSGa” on page 58  
   - “Example 1: creating a Rule where MLWTO minor = ALL” on page 143 | MainView AutoOPERATOR for z/OS | Yes |
| MSGs       | multiline WTO (MLWTO) The MSGs event type is a subset event type of the MSG event type. MSGs events appear on Rule Processor panels when the selection criteria for a MSG-initiated Rule specifies MLWTO Minor = SINGLE. For more information:  
   - “Selection Criteria fields” on page 279  
   - “MSGs” on page 59  
   - “Example 2: creating a Rule where MLWTO minor = SINGLE” on page 145 | MainView AutoOPERATOR for z/OS | Yes |
| MQS        | events that originated as IBM MQ events For more information, refer to the MainView AutoOPERATOR for MQ Installation and User Guide. | MainView AutoOPERATOR for MQ | Yes |
| NVC        | command events that originate from NetView | MainView AutoOPERATOR Access NV | Yes |
| NVM        | message events that originate from NetView | MainView AutoOPERATOR Access NV | Yes |
| TIME       | time-initiated events These Rules are fired based on the time of day. | any MainView AutoOPERATOR | No |
### 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 MainView 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.

---

### Event type

<table>
<thead>
<tr>
<th>Event type</th>
<th>Caused by</th>
<th>Product or option required</th>
<th>Event statistics available?</th>
</tr>
</thead>
</table>
| TLM        | Time Limit (TLM) events occur when one of the following time limits expires:  
- job processor time limit from the JOB statement  
- step processor time limit from the EXEC statement of the default JES  
- continuous wait time for a job from the JWT parameter in SMFPRMXX  
  Wait time is defined as the time waiting while the application program is in control. For example, the time that it takes for a data set to be recalled does not count as wait time unless the recall is by way of dynamic allocation written in a user's program.  
There will be a TLM event when the time limit has expired as specified on the JOB statement, Step statement, or continuous wait time (JWT). | MainView AutoOPERATOR for z/OS | Yes |
| VAR        | the creation, deletion, or change in value of a variable in the SHARED variable pool | any MainView AutoOPERATOR | No |

As this table shows, MainView 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, might be good candidates for automation with Rules. For more information, refer to “Using the Event Activity Statistics application” on page 41.
You can also access events with the windows-mode view, AOEVENTS. See the
MainView AutoOPERATOR Basic Automation Guide, Volume 2 for more
information.

Viewing event activity statistics

Choose the Event Activity Statistics application, Option 1, from the AUTOMATION
MENU (Figure 7 on page 42).

**Figure 7: Example of the AUTOMATION MENU**

<table>
<thead>
<tr>
<th>Basic Automation:</th>
<th>Advanced Automation:</th>
<th>Utilities:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Event Activity Statistics</td>
<td>6 Shared Object Facility</td>
<td>C CSM Conversion</td>
</tr>
<tr>
<td>2 Display/Modify Rules and Rule Sets</td>
<td>7 Display/Modify EXEC Status</td>
<td></td>
</tr>
<tr>
<td>3 Continuous State Manager - Global Overview</td>
<td>8 Time-Initiated EXEC Requests</td>
<td></td>
</tr>
<tr>
<td>4 Object Management</td>
<td>9 Open Systems Procedural Interface (OSPI)</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 8: Example of the Event Activity Statistics application**

<table>
<thead>
<tr>
<th>LC Count</th>
<th>Handled Type</th>
<th>Text-ID/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>MSG</td>
<td>PM0010I BBI/PM INITIALIZATION</td>
</tr>
<tr>
<td>27</td>
<td></td>
<td>PM0010I BBI/PM INITIALIZATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 8 on page 42** shows an example of the Event Activity Statistics application.
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 in “Describing fields” on page 47.

### How many times an event occurred: the Count column

Note that 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.

### How many times a Rule took actions for an event: the Handled column

If you have any automation already enabled, the **Handled** column shows how many times a Rule fired for an event and took the actions specified in the Rule for 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.

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 that 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 44 for more information about using primary commands; refer to “How to qualify displays in the Rule Processor applications” on page 32 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.

**Example**

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 figure shows an example of the Event Activity Statistics after SP 50 was entered.

You might find that only 1 or 2 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 “Using the Rule Processor Detail Control panel” on page 63 for more information about creating a Rule.

### Using primary commands

The following table describes the primary commands for the **COMMAND** line of the Event Activity Statistics panel.
<table>
<thead>
<tr>
<th>Command</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIND</td>
<td>F</td>
<td><code>text-id</code></td>
</tr>
<tr>
<td>LOCATE</td>
<td>L</td>
<td><code>text-id</code> or partial text-ID</td>
</tr>
<tr>
<td>RFIND</td>
<td>n/a</td>
<td>repeats the previous FIND command and searches for the same text-ID</td>
</tr>
<tr>
<td>SHOWMESSAGE</td>
<td>SM</td>
<td><code>text-id</code> or text string</td>
</tr>
<tr>
<td>Command</td>
<td>Parameters</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| 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 state, issue the SHOWALL | SA command. |
| 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 state, 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 32 for more information.

The following table contains descriptions of the fields on this panel.

---

**Note**

Events that do not show up on this panel are still eligible for automation but you must write Rules for them using the Display/Modify Rules and Rule Sets option (refer to “Managing Rules and automation using the Automation Control panel” on page 207).

---

### Table 4: Event activity statistics application field descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Activated at**           | is the date and time of either
|                            | ■ last cold start of the MainView AutoOPERATOR subsystem
|                            | ■ last time the BBI command .RESET STATS was entered
| **Total Events**           | is the total number of events collected since
|                            | ■ last cold start of the MainView AutoOPERATOR subsystem
|                            | ■ 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.
### Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>is the number of times that this event occurred since</td>
</tr>
<tr>
<td></td>
<td>- last cold start of the MainView AutoOPERATOR subsystem</td>
</tr>
<tr>
<td></td>
<td>- last time the command .RESET STATS was entered</td>
</tr>
<tr>
<td>Handled</td>
<td>is the number of times a Rule was fired to handle the event</td>
</tr>
<tr>
<td>Type</td>
<td>is the event origin; possible values for the Event Activity Statistics application are as follows:</td>
</tr>
<tr>
<td></td>
<td>ALRM, ALRT, BRC, CICa, CICS, CMD, DB2, EOM, EOS, EXT, IMP, IMS, IMSa, IMSs, JRNL, MSG, MSGa, MSGs, MQS, NVC, NVM, TIME, TLM, VAR</td>
</tr>
<tr>
<td>Text-ID/Description</td>
<td>is a field containing the Text-ID and Description of the event</td>
</tr>
<tr>
<td></td>
<td>The maximum length of the text-ID is 16 characters and the Text-ID is</td>
</tr>
<tr>
<td></td>
<td>highlighted so that it stands out from the remainder of the message text.</td>
</tr>
<tr>
<td></td>
<td>Only the first 100 characters of the message are displayed. Scrolling left and right affects</td>
</tr>
<tr>
<td></td>
<td>only the description portion of this field.</td>
</tr>
<tr>
<td></td>
<td>To scroll this display, use the LEFT or RIGHT commands from the command line or use the</td>
</tr>
<tr>
<td></td>
<td>assigned PF keys.</td>
</tr>
<tr>
<td>UNIQUE</td>
<td>is the number of unique events collected by the Event Activity Statistics application since the</td>
</tr>
<tr>
<td></td>
<td>last BBI-SS PAS cold start or the last time the BBI command .RESET STATS was entered.</td>
</tr>
<tr>
<td></td>
<td>An event is considered unique by WORD1 of the event's text.</td>
</tr>
<tr>
<td></td>
<td>New events are added to the statistics table in the order they occur after an</td>
</tr>
<tr>
<td></td>
<td>MainView AutoOPERATOR cold start or the last time the .RESET STATS command was entered.</td>
</tr>
<tr>
<td></td>
<td>The default maximum value is 510. When the maximum (510) is reached, new text-IDs are no</td>
</tr>
<tr>
<td></td>
<td>longer added to the table but the count for events already in the table continues to be</td>
</tr>
<tr>
<td></td>
<td>updated.</td>
</tr>
<tr>
<td></td>
<td>Because the Event Activity Statistics collects many event types, you might want to increase</td>
</tr>
<tr>
<td></td>
<td>the MCTSIZE. To change the default maximum, change the MCTSIZE parameter in BBISSP00 (see the</td>
</tr>
<tr>
<td></td>
<td>MainView Customization Reference for more information).</td>
</tr>
</tbody>
</table>

### Special event types

This section describes the following special event types:

- “BRC” on page 49
- “CICa” on page 50
- “DB2” on page 51
BRC events are BAROC messages from external sources that are directed to the Rules Processor.

These events may be generated from BEIM cells, other BMC product adaptors capable of producing BAROC events, and the msend command.

The event generator does not have to be defined in the BBPARM member AAOGMExx.

See the chapter "Implementing the MainView AutoOPERATOR interface to BMC Event Manager and other BMC products" in the MainView AutoOPERATOR Customization Guide for information about implementing support for BRC events.

See “Event variables and slot names for BRC events” on page 181 for more information about the event variables that are available to BRC initiated Rules.
CICa

The CICa event type is a subset of the CICS event type.

CICa events appear on Rule Processor panels when the selection criteria for a CICS-initiated Rule specifies **MLTDQ ==> ALL**.

The CICa event type applies to CICS Extrapartition or Intrapartition Transient Data messages.

A Rule with **MLTDQ ==> ALL** specified does not fire unless the entire multiline transient data queue (MLTDQ) message has been received. It fires only when CICS has sent the last line of a message, or the maximum time to wait between receiving lines has been reached.

**Note**

For a CICS-initiated Rule, when you specify **MLTDQ ==> ALL**, you must also specify values for the selection criteria field **Text ID** or **Text String**.

BMC recommends that you use the variables &LINEx\_WORDn on the Variable Dependencies panel so that you can create Rules that fire based on specific content of the MLTDQ. The variables are as follows:

- **&LINEx** resolves to the line of the MLTDQ message.
  - &LINE1 contains the contents of the major line of the MLTDQ, and &LINE2 through &LINEx contain the subsequent minor lines.

- **WORDn** resolves to words on &LINEx.
  - Individual &WORDn variables that are not part of an &LINEx stem variable always refer to words on &LINE1 (the major line).

**MLTDQ ==> ALL** should be specified in CICS-initiated Rules that need to access the complete CICS multiline message in the Rule and take actions.

For an example, see “Creating a Rule for CICS multiline messages” on page 151.
**DB2**

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 z/OS 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 z/OS 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 z/OS have access to DB2 messages as both event types. For simplicity, these customers should automate DB2 by using the MSG event type.

If you do not have the MainView AutoOPERATOR for z/OS 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 Customization Reference.

**EOM**

End-of-Memory (EOM) events occur after an address space ends.

An address space ends when a Started Task (STC) or TSO User (TSU) is stopped, canceled, or ends abnormally (abends). It is important to note that an address space does not end when a batch job terminates. It ends when the initiator is terminated.

To create Rules for this event type, you should evaluate the contents of the variable IMFTERMT. See “List of MainView AutoOPERATOR-supplied EVENT variables” on page 169 for more information about this variable.

For EOM events, the determination of whether the address space terminates as NORMAL or ABNORMAL depends on the program running in the address space. Programs that have error recovery routines could report the termination type as NORMAL even though an abend may have occurred. You need to make the determination of what termination types will be produced by an address space before coding an EOM rule.

For example, canceling a TSO/E foreground user address space results in an EOM processing termination type as NORMAL instead of ABNORMAL termination. Because TSO recovers from the cancel request and proceeds with a normal log off to
end the session, the termination type is NORMAL. Other programs might also do
the same depending on the error recovery provided by the running program.

There is some overlap between the EOM and EOS events for STC and TSO address
spaces. The exception is that EOS events do not always occur. For example, if an
address space ends by way of the SYSPROG Services commands FORCE or EXIT, an
EOM event is generated, but an EOS event is not generated.

The IMFTEXT event variable contains the following information when the event
fires:

dx0430i ttttttttt eom asid: aaaaa a.s. name: nnnnnnnn
   tttttttt: Type of time EOM. "NORMAL" or "ABNORMAL".
            aaaaa: ASID (decimal) of STC or TSO address space that is
            terminating.
            nnnnnnnn: Name of STC or TSO address space that is terminating.

The DX0430I message is not written to the BBI Journal unless journal => yes is
specified in an EOM Rule.

EOS

The End-of-Step (EOS) event occurs whenever an address space changes steps.

An additional EOS event occurs when the address space ends; for example, at the
EOJ (End of Job) event.

Before EOS events are generated, you must have iefactrt smf exits implemented
and generate SMF Type 30 records. See the MainView AutoOPERATOR
Customization Guide for more information about implementing the EOS event type.

To write Rules that fire for this event type, you should evaluate the contents of the
following event variables to determine what event data is available to you for
automation purposes:

- IMFJSACT
- IMFJSNUM
- IMFJSTEP
- IMFJCPU
- IMFJSCPU
- IMFJCAN
- IMFJSCC
See “List of MainView AutoOPERATOR-supplied EVENT variables” on page 169 for more information about these variables. For an EOJ event, the event variables IMFJNUM, IMFJSTEP, IMFJSCPU, and IMFJCAN will have slightly different values.

There is some overlap between the EOM and EOS events for STC and TSO address spaces. The exception is that EOS events do not always occur. For example, if an address space ends by way of the SYSPROG Services commands FORCE or EXIT, an EOM event is generated, but an EOS event is not generated.

The IMFTEXT event variable contains the following information when the event fires:

```
CE0330I jjjjjjjjjj JS: sssssss PS: pppppppp CC: ccccc SNum=nnn
jjjjjjjjjj: Job, STC or TSO address space name.
ssssssss: Job Step name, or N/A
pppppppp: Proc Step name, or N/A
cccc: Condition code - FLUSH, nnn (return code), Abend code (preceded with "S" or "U"), or N/A.
nnn: Step Number, zero for end of job.
```

The CE0330I message is not written to the BBI Journal unless Journal ==> YES is specified in an EOS Rule.

**EXT**

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 MainView 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 job step

The MainView 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 MainView 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.
You can use the variable &LINEx_WORDy on the Variable Dependencies panel so that you can create Rules that fire based on specific content of an external message where

&LINE1 resolves to the text of the external event up to a maximum of 254 characters. The &IMFTEXT variable truncates the message of the external event at 125 characters and &LINE1 truncates it at 254 characters.

&LINE1 is available in an EXEC scheduled from an EXT event type Rule, but it is only provided as a LOCAL variable. You must use the IMFEXEC VGET command in an EXEC to retrieve the value and use it as a TSO variable.

&LINE_WORDy is available as event variables in EXT Rules where these variables represent each word of LINE1 (ending after 254 characters). You can use these variables in the selection criteria and action panels of the Rule Processor. These variables are not passed to an EXEC scheduled from a Rule.

**HWTO**

The event type HWTO is for early and interim WTOs that were issued when MainView AutoOPERATOR was inactive.

The HWTO event type is available only when you have the BMC MainView Automation Solution active in the MainView AutoOPERATOR BBI-SS PAS. For more information, see the *MainView AutoOPERATOR Customization Guide*.

HWTO events are WTOs that were issued when MainView AutoOPERATOR was inactive. There are two kinds of HWTOs:

- Early WTOs that are issued during IPL until MainView AutoOPERATOR starts
- Interim WTOs that are issued during brief planned outages of MainView AutoOPERATOR

When you specify WTOEARLY=Y in BBPARM member AAOPRMxx, early WTOs are stored in a data space. The QAOCNZ15 utility collects these WTOs until the data space becomes full.

When MainView AutoOPERATOR initializes, these saved early WTOs are available to the Rule Processor as HWTO events. The HWTO events are processed immediately after the PAS starts and do not retain the timing of the original WTO messages.

When you specify WTOINTRM=Y in BBPARM member AAOPRMxx and stop the PAS, the Rule Processor keeps running to record interim WTOs in an extended CSA table. When a WTO occurs, the WTO control block (WQE) is copied to the extended
CSA table. Later when MainView AutoOPERATOR initializes, these WQEs are available to the Rule Processor as HWTO events.

This table can contain about 10,000 WQEs. Note that many WQEs may be associated with a multi-line WTO.

HWTO events are similar to MSG events because they represent WTOs. However fewer Rule actions are available and fewer Rule and EXEC variables are valid for HWTO Rules as compared with MSG Rules.

See “List of variables for HWTO-initiated Rules” on page 184 for a list of variables that are valid in a HWTO Rule. For a list of variables that are valid in an EXEC scheduled by an HWTO Rule, see the MainView AutoOPERATOR Advanced Automation Guide.

**IMP**

The IMP (IMSplex) event type captures IMS type-2 (IMSplex) commands.

The MainView AutoOPERATOR OM Input Exit (BMCAOOM) detects the IMS Type-2 commands before the IMS OM processes the command. This allows a Rule to reject or modify the text of IMS Type-2 commands.

The requirements for the Rule Processor to detect these events are as follows:

- The MainView AutoOPERATOR for IMS component active in the PAS.
- The IMS OM address space runs on the same z/OS image as the MainView AutoOPERATOR PAS.
- You must specify the MainView AutoOPERATOR OM Input Exit (BMCAOOM) in the OM exit routine definitions list in the OM Proclib member: for example: EXITDEF=(TYPE=INPUT,EXITS=(BMCAOOM),COMP=OM).
- Include the MainView AutoOPERATOR load library in the STEPLIB of the IMS OM started task JCL.

The MainView AutoOPERATOR OM Input Exit broadcasts IMP events to all eligible MainView AutoOPERATOR PASs that are running on the z/OS image. To exclude an MainView AutoOPERATOR PAS, specify IMSOMEX=NO in BBPARM member AAOPRMxx for the PAS.

IMP Rules can intercept IMS Type-2 commands with up to 255 characters long text.

The following variables contain information that is specific to the IMP-initiated Rules:
&IMFOMCLT: contains the client member name

&IMFOMUSR: contains the user ID of the application that issued the command

&IMFOMRTE: contains the route list that is associated with the command (up to a maximum of 30 characters)

IMP Rules can take the following actions to reject or reword the IMS Type-2 command:

- **Reject command = YES.** When an Rule takes this action and rejects a command, any eligible IMP-initiated Rules can still detect and process the commands. However, if the other IMS OM Input exits are listed in the proclib after the BMCAOOM exit, MainView AutoOPERATOR does not pass the command to the other IMS OM Input exits.

- **Audit Reject = YES.** When a Rule rejects the IMS Type-2 command, the PAS issues the DX0642I audit message to the Journal.

- **Reword command.** Use this action to modify the text of an IMS Type-2 commands. The maximum length of the new text is the original command length plus 80 characters (up to a maximum length of 255). The command text is immediately modified and the Rule Processor uses the text for subsequent Rule comparisons.

## IMSa

The IMSa event type is a subset event type of the IMS event type.

IMSa events appear on Rule Processor panels when the selection criteria for a Rule initiated by IMS specifies MLSEG Minor = ALL.

The IMSa event type applies to:

- IMS multisegment messages (IMS messages sent in multisegment format through the AOI exit)

- Command response segments to IMS commands issued from a MainView AutoOPERATOR EXEC with response

A Rule with MLSEG Minor = ALL specified is not evaluated until the entire MLSEG message has been received. In other words, it does not fire until IMS has sent the last segment of a message (or last response for a command), or the maximum time to wait between receiving segments (IMSMSTIM) has been reached.
Note
When you specify MLSEG Minor = ALL, you must also specify values for the selection criteria fields Text ID or Text String.

In addition, when you specify MLSEG Minor = ALL, the Rule cannot take the action of suppressing or rewording the message. Therefore, the action fields Display at Dest and Reword Msg (which normally appear for IMS Rules on the Action Specification panel), do not appear when you specify MLSEG Minor = ALL.

BMC Software recommends that you use the variables &LINEx. WORDn on the Variable Dependencies panel so that you can create Rules that fire based on specific content of the MLSEG where

- &LINEx resolves to the segment of the MLSEG where &LINE1 contains the contents of the major segment of the MLSEG and &LINE2 through &LINEn contains each succeeding minor segment
- WORDn resolves to words on the first (major) segment
  Individual &WORDn variables that are not part of an &LINEn stem variable always refer to words on &LINE1 (the major line).

MLSEG Minor = ALL should be specified only in IMS-initiated Rules that need to access the complete IMS multisegment message in the Rule and take actions, other than Journal=YES. The Journal=YES action logs the complete multisegment messages to the Journal by default when MLSEG Minor is not specified.

See “Example 1: creating a Rule with MLSEG Minor = ALL” on page 147.

See “Selecting INDIVIDUAL” on page 219 for information about this event type and the automation strategy INDIVIDUAL.

**IMSs**

The IMSs event type is a subset event type of the IMS event type.

IMSs events appear on Rule Processor panels when the selection criteria for a Rule initiated by IMS specifies MLSEG Minor = SINGLE.

The IMSs event type applies to:

- IMS multisegment messages (IMS messages sent in multisegment format through the AOI exit)
- Command response segments to IMS commands issued from a MainView AutoOPERATOR EXEC with response
With IMSs events, the Rule treats each minor segment of the MLSEG as a separate entity, which means that each minor segment (or all of the minor segments) of the MLSEG can cause the Rule to fire.

---

**Note**

For an IMS-initiated Rule, when you specify MLSEG Minor = SINGLE, you must also specify values for the selection criteria fields Text ID or Text String.

In addition, when you specify MLSEG Minor = SINGLE, the Rule cannot take the action of suppressing or rewording the message. Therefore, the action fields Display at Dest and Reword Msg (which normally appear for IMS Rules on the Action Specification panel), do not appear when you specify MLSEG Minor = SINGLE.

BMC Software recommends that you use the &LINE2_WORDn variables on the Variable Dependencies panel so that you can create Rules that filter out unwanted minor segments where

- &LINE2 resolves to each and every minor segment of the MLSEG (&LINE1 contains the contents of the major segment)

Therefore, with MLSEG Minor = SINGLE specified as selection criteria, the Rule Processor resolves the &LINE1 variable as the major segment and each minor segment is resolved as &LINE2 no matter where a specific minor segment appears within the entire MLSEG.

- WORDn resolves to words on the first (major) segment
- &LINE2_WORDn resolves to each word of the segment

MLSEG Minor = SINGLE should be specified only in IMS Rules that need to access a minor segment in the Rule and take actions, other than Journal=YES. The Journal=YES action logs the complete multisegment messages to the Journal by default when MLSEG Minor is not specified.

See “Example 2: creating a Rule where MLSEG Minor = SINGLE” on page 150 for an example.

See “Selecting INDIVIDUAL” on page 219 for information about this event type and the automation strategy INDIVIDUAL.

---

**MSGa**

The MSGa event type is a subset event type of the MSG event type.

MSGa events appear on Rule Processor panels when the selection criteria for a MSG-initiated Rule specifies MLWTO Minor = ALL.
A Rule with **MLWTO Minor = ALL** specified is not evaluated until the entire MLWTO has been issued. In other words, it does not fire until the MLWTO is considered complete by z/OS or when MainView AutoOPERATOR times out. A MLWTO is considered complete when the text line with a text type of E or DE has been received. For more information about message text type codes, refer to the IBM publication *MVS Programming: Assembler Services Reference, Volume 2*.

---

**Note**

For a MSG-initiated Rule, when you specify **MLWTO Minor = ALL**, you must also specify values for the selection criteria fields Text ID or Text String.

In addition, when you specify **MLWTO Minor = ALL**, the Rule cannot take the action of suppressing the message or preventing it from being recorded in the MVS SYSLOG. Therefore, the action fields **Display at Dest** and **SYSLOG Display** (which normally appear for MSG-initiated Rules on the Action Specification panel), do not appear when you specify **MLWTO Minor = ALL**.

BMC Software recommends that you use the variable &LINEx\_WORDn on the Variable Dependencies panel so that you can create Rules that fire based on specific content of the MLWTO where

- &LINEx resolves to the line of the MLWTO message. &LINE1 contains the contents of the major line of the MLWTO and &LINE2 through &LINEx contain the succeeding minor lines.

- WORDn resolves to words on &LINEx.
  
  Individual &WORDn variables that are not part of an &LINEx stem variable always refer to words on &LINE1 (the major line).

See “Example 1: creating a Rule where MLWTO minor = ALL” on page 143 for an example.

See “Selecting INDIVIDUAL” on page 219 for information about this event type and the automation strategy INDIVIDUAL.

---

**MSGs**

The MSGs event type is a subset event type of the MSG event type.

MSGs events appear on Rule Processor panels when the selection criteria for a MSG-initiated Rule specifies **MLWTO Minor = SINGLE**.

With MSGs events, the Rule treats each minor line of the MLWTO as a separate entity, which means that each minor line (or all of the minor lines) of the MLWTO can cause the Rule to fire.
For a MSG-initiated Rule, when you specify **MLWTO Minor = SINGLE**, you must also specify values for the selection criteria fields Text ID or Text String.

In addition, when you specify **MLWTO Minor = SINGLE**, the Rule cannot take the action of suppressing the message or preventing it from being recorded in the MVS SYSLOG. Therefore, the action fields **Display at Dest** and **SYSLOG Display** (which normally appear for MSG-initiated Rules on the Action Specification panel), do not appear when you specify **MLWTO Minor = SINGLE**.

BMC Software recommends that you use the &LINE2\_WORDn variable on the Variable Dependencies panel so that you can create Rules that filter out unwanted minor lines where

- **&LINE2** resolves to each and every minor line of the MLWTO (**&LINE1** contains the contents of the major line)

Therefore, with **MLWTO Minor = SINGLE** specified as selection criteria, the Rule Processor resolves the **&LINE1** variable as the major line and each minor line is resolved as **&LINE2** no matter where a specific minor line appears within the entire MLWTO.

- **WORDn** resolves to words on the first (major) line

See “Example 2: creating a Rule where MLWTO minor = SINGLE” on page 145 for an example.

See “Selecting INDIVIDUAL” on page 219 for information about this event type and the automation strategy INDIVIDUAL.

**TIME**

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 A.M. every weekday
- every Wednesday from 1:00 P.M. to 3:00 P.M. at 5-minute intervals
- at BBI-SS PAS startup
- at 1:00 P.M. and fires every 10 minutes until 2:00 P.M.
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 the rule 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.

For information about how your automation with time-initiated Rules can be affected when you change the MVS Local Time to observe daylight savings time, refer to “Field descriptions for the Rule Processor” on page 279.

Refer to “Creating TIME-initiated Rules” on page 119 for an example of creating a TIME-initiated Rule.

**TLM**

A Time Limit (TLM) event occurs when one of the following time limits expire:

- Job processor time limit from the JOB statement.
- Step processor time limit from the EXEC statement of the default from JES.
- Continuous wait time for a job from the JWT parameter in SMFPRMXX.

Wait time is defined as the time waiting while the application program is in control. For example, the time that it takes for a data set to be recalled does not count as wait time unless the recall is by way of dynamic allocation written in a user’s program.

Before TLM events are generated, you must have the IEFUTL SMF exit implemented. See the *MainView AutoOPERATOR Customization Guide* for more information about implementing the TLM event type.

To create Rules for this event type, you should evaluate the contents of the following event variables to determine what event data is available to you for automation purposes:

- IMFTTLM
- IMFTSSID
- IMFTSNUM
- IMFTCPCT
See “List of MainView AutoOPERATOR-supplied EVENT variables” on page 169 for more information about these variables. The IMFTEXT event variable will contain the following textual information when the event fires:

```
CE0430I jjjjjjjjj tttt time exceeded by ssss
jjjjjjjjj: The job, STC, or TSO address space name whose time limit has been exceeded.
tttt: Type of time exceeded. Values are JCPU, SCPU or WAIT.
ssss: Subsystem Id that selected the job. JES2, JES3, ASCH (APPC), OMVS, STC, TSO, or jobname if none of the above applies and it is less then 4 characters.
```

The CE0430I message is not written to the BBI Journal unless `Journal ==> YES` is specified in a TLM Rule.

**VAR**

The VAR event is an event where a variable in the SHARED pool is created, deleted, or its value was changed.

You cannot specify long variables (long variables are not supported in the Rule Processor). You can create Rules that are triggered to fire based on this event. These events never appear in the Event Activity Statistics application.
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 MainView 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.

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:

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

Also, seven 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(s) I
4. Action Specification(s) II

5. Set Variable(s)

6. Alert Action(s) I

7. Alert Action(s) II

8. BEIM Event Action panel (this option appears on the Rule Processor Detail Control panel only if you have specified PRODUCT=IIZ in BBPARM member BBISSP00)

9. Alert Action(s) III -- BEIM (this option appears on the Rule Processor Detail Control panel only if you have specified PRODUCT=IIZ in BBPARM member BBISSP00)
The Rule Processor Detail Control panel is redisplayed.

Figure 9: 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 procedure describes how to access the Rule Processor Detail Control panel from either application.

To access the Rule Processor Detail Control panel

1. From the Event Activity Statistics panel, enter S (for the (S)elect line command) in the LC field next to the event that you want to write a Rule for (Figure 10 on page 66).

Figure 10: Selecting an event

The Rule Processor Detail Control panel is displayed (Figure 11 on page 66).

Figure 11: Rule Processor Detail Control panel
The following options are displayed in sequence, or may be selected by entering the two-character code.

- **S1** - Selection Criteria
- **SV** - Variable Dependencies
- **A1** - Action Specification(s) I
- **A2** - Action Specification(s) II
- **AV** - Set Variable(s)
- **AA** - Alert Action(s) I
- **AD** - Alert Action(s) II
- **BA** - Event Action(s) - BEIM
- **AE** - Alert Action(s) III - BEIM

<table>
<thead>
<tr>
<th>Rule ID</th>
<th>Event Type</th>
<th>Initial Mode</th>
<th>Criteria match rate threshold:</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAO002I</td>
<td>ALRT</td>
<td>ENABLED</td>
<td>If matched, then status</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Maximum # times matched within INTERVAL, 0-100)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Interval length, 1-99999 seconds)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(SUSPEND, DISABLE, NOACTION)</td>
</tr>
</tbody>
</table>

Application information:
- **Group**
- **Func**
- **Code**

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

2 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 (Figure 12 on page 67).

--- THE FOLLOWING OPTIONS ARE DISPLAYED IN SEQUENCE. OR MAY BE SELECTED BY ENTERING THE TWO-CHARACTER CODE ---

**RULE ID**
**Event Type**
**Initial Mode**

--- YY/MM/DD

Rule ID => EAO002I
Event Type => ALRT
Initial Mode => ENABLED

Criteria match rate threshold:
If matched =>
then status =>

Application information:
Group =>
Func =>
Code =>

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

--- THE FOLLOWING OPTIONS ARE DISPLAYED IN SEQUENCE. OR MAY BE SELECTED BY ENTERING THE TWO-CHARACTER CODE ---

Accessing the Rule Processor Detail Control panel

--- THE FOLLOWING OPTIONS ARE DISPLAYED IN SEQUENCE. OR MAY BE SELECTED BY ENTERING THE TWO-CHARACTER CODE ---

Chapter 4 Using the Rule Processor Detail Control panel 67
Automation Strategy ===> INDIVIDUAL (Individual, All, First)  
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  

<table>
<thead>
<tr>
<th>Rule Set</th>
<th>Status</th>
<th>Rules</th>
<th>Fired</th>
<th>Filtered</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAORUL00</td>
<td>ENABLED</td>
<td>14</td>
<td>5,140</td>
<td>23,655</td>
</tr>
<tr>
<td>AAORULBC</td>
<td>ENABLED</td>
<td>54</td>
<td>232</td>
<td>23,655</td>
</tr>
<tr>
<td>AAORULCM</td>
<td>ENABLED</td>
<td>52</td>
<td>124</td>
<td>23,655</td>
</tr>
</tbody>
</table>

**Automation Control panel: creating a new Rule**  

Select a Rule Set by entering S (for the (S)elect line command) next to the Rule Set name (Figure 14 on page 68).  

**Figure 14: Automation Control panel: creating a new Rule**  

Accessing the Rule Processor Detail Control panel
Or, add a new Rule Set with the ADD primary command on the COMMAND line (Figure 15 on page 69).

**Figure 15: Adding a new Rule Set**

BMC Software -------------- Automation Control --------- MAINVIEW AutoOPERATOR
COMMAND ===> ADD
RULALRT
AOAO
Primary commands: ADD, STATSHOW, CMDSHOW, PARMSHOW, Locate.
--- YY/MM/DD
EXPAND
17:48:38
Automation Status ===> ACTIVE (Active, Inactive)
Automation Strategy ===> INDIVIDUAL (Individual, All, First)
Honor MPF Suppression ===> NO (NO/YES)

![Automation Statistics](image)

LC CMDS --- (S)elect, (E)nable, (D)isable, (T)est, (SA)ve
(M)ove, (B)efore or (A)fter, (F)ilter Criteria

<table>
<thead>
<tr>
<th>Rule-Set Status</th>
<th>Rules</th>
<th>Fired</th>
<th>Filtered</th>
<th>Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAORUL00 ENABLED</td>
<td>14</td>
<td>5,140</td>
<td>23,655</td>
<td>DD-MMM-YY 14:15:54</td>
<td></td>
</tr>
<tr>
<td>AAORULBC ENABLED</td>
<td>54</td>
<td>232</td>
<td>23,655</td>
<td>DD-MMM-YY 14:15:55</td>
<td></td>
</tr>
<tr>
<td>AAORULCM ENABLED</td>
<td>52</td>
<td>124</td>
<td>23,655</td>
<td>DD-MMM-YY 14:15:55</td>
<td></td>
</tr>
<tr>
<td>RULCICS ENABLED</td>
<td>25</td>
<td>0</td>
<td>23,655</td>
<td>DD-MMM-YY 14:15:57</td>
<td></td>
</tr>
</tbody>
</table>

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

The Rule Processor Detail Control panel is displayed; see Figure 16 on page 69.

**Figure 16: Rule Processor Detail Control panel**

<table>
<thead>
<tr>
<th>Rule ID</th>
<th>Time</th>
<th>Selection Criteria</th>
<th>Set Variable(s)</th>
<th>Variable Dependencies</th>
<th>Alert Action(s) I</th>
<th>Alert Action(s) II</th>
<th>Alert Action(s) III</th>
<th>Event Action(s) - BEIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>EA00002I</td>
<td>17:25:41</td>
<td>S1 -</td>
<td>AAORUL0A DISABLED</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SV -</td>
<td>AAORUL0B DISABLED</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AAORUL0F DISABLED</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AAORUL0S DISABLED</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AAORUL0CS DISABLED</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AAORUL1 DISABLED</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AAORUL1A DISABLED</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AAORUL1B DISABLED</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AAORUL1F DISABLED</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AAORUL1S DISABLED</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AAORUL1CS DISABLED</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AAORUL2 ENABLED</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AAORUL2A ENABLED</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AAORUL2B ENABLED</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AAORUL2F ENABLED</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AAORUL2S ENABLED</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AAORUL2CS ENABLED</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

Chapter 4  Using the Rule Processor Detail Control panel  69
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, MainView 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 MainView 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 “Using the Event Activity Statistics application” on page 41.

- **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 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.

   **Figure 17: Rule Processor Detail Control panel: displaying available event types**

   Accessing the Rule Processor Detail Control panel
The following Rule Processor Event Types panel is displayed.

**Figure 18: Rule Processor Event Types panel**

BMC Software ----------- Rule Processor Event Types ----------- AutoOPERATOR COMMAND

---

TGT --- DL81

MVS Events: (AO for z/OS Required)  IMS Events: (IAO Option Required)
C CMD - Operator commands.  I IMS - IMS MTO messages.
M MSG - WTO or WTOR message.  P IMP - IMSplex (OM) commands.
E DS - End of Step or Job.
E DM - STC/TSO End of Memory.
T LM - CPU/Wait time exceeded.  Q CICS - Messages from TD Queue.

(BMC MainView Automation Req)  MQ Events: (QAO Option Required)
HWTO - Historical WTO.  S MQS - WebSphere MQ instrumentation

AQ Events: (Common to all options)
A ALRT - Alerts added to Queue.
B BRC - BAROC events.
J JRNL - BBI Journal messages.
L ALRM - Alarm Manager events.
T TIME - Time-initiated Rule.
X EXT - External msgs.
V VAR - Shared variables updated, created or deleted

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

For more information about event types, refer to “Describing events” on page 35

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

For more information about the other fields on the Rule Processor Detail Control panel that can be filled in at the beginning or end of Rule creation, refer to “Rule Processor Detail Control panel” on page 71.

**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 MainView 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 \textit{in seconds}, the Rule status is changed to the \textit{then status} value.

Specifying a \textit{then status} of Disable will disable the Rule. You can manually enable this Rule. Disabled Rules are ignored by the Rule Processor.

Specifying a \textit{then status} of SUSPEND temporarily disables the Rule. MainView 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 \textit{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.
Tip

When specifying action specifications for a Rule, you should ensure that the action specification does not create events that re-trigger the original Rule. This type of error results in a loop and the Rules might fire endlessly.

To prevent this situation from happening, it is recommended that you set a limit for how many a times a Rule fires. “Criteria match rate thresholds” on page 78 describes how you can limit the number of times a Rule fires and therefore, prevent such loops.

- Application 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.

Primary commands

You can enter the primary commands shown in the following table on the COMMAND line 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 provides a brief description.

Table 5: Primary commands on the Rule Processor application panels

<table>
<thead>
<tr>
<th>Command</th>
<th>Displays</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blank (Default)</td>
<td>Selection Criteria panel when you press Enter</td>
<td>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.</td>
</tr>
<tr>
<td>S1</td>
<td>Selection Criteria - event type panel</td>
<td>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.</td>
</tr>
<tr>
<td>SV</td>
<td>Variable Dependencies - event type panel</td>
<td>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.</td>
</tr>
<tr>
<td>Command</td>
<td>Displays</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>ASV</td>
<td>Advanced Variables Dependencies - event type panel&lt;br&gt;ASV command is valid only when issued from the SV panel.</td>
<td>Use the Advanced Variable Dependencies panel to create comparisons and equations to further qualify whether the event should be automated by the Rule. Use of this panel is optional.</td>
</tr>
<tr>
<td>A1</td>
<td>Action Specification(s) I - event type panel</td>
<td>When the selection criteria of a Rule matches an event and the Rule is fired, the actions that are specified on this panel are taken by the Rule.</td>
</tr>
<tr>
<td>A2</td>
<td>Action Specification(s) II - event type panel</td>
<td>This panel is a continuation of the Action Specification(s) I panel and includes additional fields that you can use for taking actions when the Rule fires.</td>
</tr>
<tr>
<td>AV</td>
<td>Action: Set Variables - event type panel</td>
<td>Use the Set Variables panel to set, increment, or decrement one or more shared variables.</td>
</tr>
<tr>
<td>AAV</td>
<td>Advanced Set Variables - event type panel&lt;br&gt;AAV command is valid only when issued from the AV panel.</td>
<td>Use this panel to set, increment, or decrement one or more SHARED variables and use powerful built-in Rules functions with variables. Use of this panel is optional.</td>
</tr>
<tr>
<td>AA</td>
<td>Action(s) I - event type panel</td>
<td>This panel applies only to the Rules that 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.</td>
</tr>
<tr>
<td>AD</td>
<td>Alert Action(s) II - event type panel</td>
<td>This panel is a continuation of the Alert Action(s) I panel and includes additional fields that you can use while creating an MainView AutoOPERATOR ALERT from a Rule.</td>
</tr>
<tr>
<td>BA</td>
<td>Event Action - BEIM - event type panel</td>
<td><strong>Note:</strong> This primary command and the associated panels appear only if you have installed the BMC Impact Integration for z/OS product and specified PRODUCT=IIZ in BBPARM member BBISSP00. Specifying BA displays the Event Action - BEIM panel, which allows you to send events from MainView AutoOPERATOR to a BMC Impact Manager cell.</td>
</tr>
</tbody>
</table>
This topic describes the fields on the Rule Processor Detail Control panel.

### Top portion of the panel

The following section describes the fields in the top portion of the panel.

#### Figure 19: Top portion: Rule Processor Detail Control panel

<table>
<thead>
<tr>
<th>Command</th>
<th>Displays</th>
<th>Description</th>
</tr>
</thead>
</table>
| AE       | Alert Action(s) III - BEIM - event type | **Note:** This primary command and the associated panels appear only if you have installed the BMC Impact Integration for z/OS product and specified PRODUCT=IIIZ in BBPARM member BBISSP00. Use this panel to specify:
- whether details for the Alert should be written to the BBI Journal
- a new class name to be associated with the slots and values
- "Component Alias" for event correlation against objects in the Service Impact Model (SIM)
- additional slots and values to be sent to the BIM cell
- additional information about this rule's BiiZ slots |

<table>
<thead>
<tr>
<th>Rule ID</th>
<th>IMSJRNL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Type</td>
<td>Type of event ( ? for list)</td>
</tr>
<tr>
<td>Initial Mode</td>
<td>ENABLED  (ENABLED/DISABLED/TEST)</td>
</tr>
</tbody>
</table>
Table 6: Rule Processor Detail Control panel field descriptions: top portion

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule ID</td>
<td>one - to eight-character unique name of the Rule  \nWhen you assign a Rule ID, you must remember to make it unique from previously created Rules because a duplicate Rule ID overrides the original Rule.  \nValid 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 ($).  \nIf 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.  \nIn 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.</td>
</tr>
<tr>
<td>Event Type</td>
<td>event type that describes the source of the message  \n“Describing events” on page 35 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.  \nYou must enter an event type before you can proceed to the subsequent Rule creation panels.</td>
</tr>
<tr>
<td>Initial Mode</td>
<td>initial mode of the Rule when its Rule Set is enabled  \nAll 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 as follows:  \n- ENABLED means that the Rule is available to process events.  \n- DISABLED means that the Rule is not available to process events.  \n- TEST means that the Rule will be fired in test mode.  \n  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.  \nOnce the Rule Set is enabled, you can change the mode of an individual Rule on the Rule Set Overview panel.</td>
</tr>
</tbody>
</table>

Criteria match rate fields

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 “Criteria match rate thresholds” on page 78.

Figure 20: Middle portion: Rule Processor Detail Control panel

Criteria match rate threshold:  
If matched ===> (Maximum # times matched within INTERVAL, 0-100)
Table 7: Rule Processor Detail Control panel field descriptions: middle portion

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>If matched</td>
<td>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 value range is 0 to 100 times. 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 “Criteria match rate thresholds” on page 78 for more information.</td>
</tr>
<tr>
<td>in seconds</td>
<td>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 “Criteria match rate thresholds” on page 78 for more information.</td>
</tr>
<tr>
<td>then status</td>
<td>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 as follows:</td>
</tr>
<tr>
<td></td>
<td>■ 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.</td>
</tr>
<tr>
<td></td>
<td>■ 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 “Criteria match rate thresholds” on page 78 for more information.</td>
</tr>
<tr>
<td></td>
<td>■ 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.</td>
</tr>
</tbody>
</table>
Application information fields

The following table describes the Application information portion of the panel.

Figure 21: Bottom portion: Rule Processor Detail Control pane

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Group</strong></td>
<td>user-defined 1- to 8-character group name assigned to this Rule</td>
</tr>
<tr>
<td><strong>Function</strong></td>
<td>user-defined 1- to 8-character function name assigned to this Rule</td>
</tr>
<tr>
<td><strong>Code</strong></td>
<td>user-defined 2-character code assigned to this Rule</td>
</tr>
<tr>
<td><strong>Author</strong></td>
<td>user-defined 1- to 8-character name of author of this Rule</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>user-defined 1- to 40-character description of this Rule</td>
</tr>
<tr>
<td></td>
<td>The description can be up to 40 characters but on the Rule Set Overview panel, only the first 25 characters are displayed.</td>
</tr>
<tr>
<td><strong>Last Modified</strong></td>
<td>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</td>
</tr>
</tbody>
</table>

Criteria match rate thresholds

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.

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. For more information about setting automation thresholds and the criteria match rate, see “Managing Rules and automation using the Automation Control panel” on page 207.

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.

---

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

**Figure 22: Criteria match rate example**

<table>
<thead>
<tr>
<th>Criteria match rate threshold:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>If matched</strong> ===&gt; 10 (Maximum # times matched within INTERVAL, 0-100)</td>
</tr>
<tr>
<td><strong>in seconds</strong> ===&gt; 30 (Interval length, 1-9999 seconds)</td>
</tr>
<tr>
<td><strong>then status</strong> ===&gt; SUSPEND (SUSPEND, DISABLE, NOACTION)</td>
</tr>
</tbody>
</table>

---

**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.

---

When a match rate is specified in this Rule Set Filter, every Rule will have a criteria match rate, either an explicit match rate coded in the Rule or an 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 about using Rule Set filtering with criteria match rate, see “Enabling filtering and Rule Set match rate for Rule Sets” on page 221.

---

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.
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 to which the Rule belongs is explicitly saved and written to disk. To save a new Rule or changes made to an existing Rule, you must perform the following steps:

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.

3 If you press PF3/END from the Rule Set Overview panel before you save the Rule Set, Figure 23 on page 81 is displayed.

**Figure 23: Example of Confirm Rule Set panel**

```
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.                       +
+ o A FILTER was added or modified.                                 +
+ 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.                        +
+----------------------------------------------------------------------------+
```

**Note**

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 situation 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.

4 If you press PF3/END on the Rule Set Overview panel and you have moved a Rule between Rule Sets, resulting in two Rule Sets that need to be saved, Figure 24 on page 81 is displayed:

**Figure 24: Example of Confirm Rule Set Modifications panel**

```
BMC Software  ------------ Confirm Rule Set Modifications -------- AutoOPERATOR
COMMAND ===>                                                 TGT  --- DL73

+ WARNING! One or more Rule(s) were moved from Rule set RULAO73 to +
+ Rule set RULEXECS Both Rule sets should be saved.               +
+ o RULAO73 was NOT SAVED                                         +
+ o RULEXECS was NOT SAVED                                         +
+----------------------------------------------------------------------------+
```
Please do one of the following:

- Enter SAVE to save Rule set(s) from the above list that have not been saved to the BBIPARM dataset.
- Enter NOSAVE to exit WITHOUT saving either Rule Set.
- Press END to return to Rule Set Overview.
Creating Rules

This chapter explains the steps for using primary commands to create Rules.

Overview of creating Rules

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 35 for a description of the event types that the Rule Processor supports
- “Examples of creating Rules” on page 107 for examples that show how different types of Rules are created for different event types
- MainView AutoOPERATOR for WebSphere MQ Installation and User Guide for more information about creating Rules for WebSphere MQ events
On the **COMMAND** line of the Rule Processor Detail Control panel (or any of the panels used in Rule creation), you can enter the primary commands that are shown in the following table as a shortcut to a specific panel.

### Table 9: Using primary commands in the Rule Processor

<table>
<thead>
<tr>
<th>Command</th>
<th>Displays</th>
<th>Description</th>
</tr>
</thead>
</table>
| blank (Default) | 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.                                                                                                         |
| ASV             | Advanced Variables Dependencies - *event type* panel  
**ASV** command is valid only when issued from the **SV** panel. | Use the Advanced Variable Dependencies panel to create comparisons and equations to further qualify whether the event should be automated by the Rule.  
Use of this panel is optional.                                                                                                         |
| A1              | **Action Specification(s) I - event type** panel | When the selection criteria of a Rule matches an event and the Rule is fired, the actions that are specified on this panel are taken by the Rule. |
| A2              | **Action Specification(s) II - event type** panel | This panel is a continuation of the **Action Specification(s) I** panel and includes additional fields that you can use for taking actions when the Rule fires. |
| AV              | **Action: Set Variables - event type** panel | Use the Set Variables panel to set, increment, or decrement one or more shared variables.                                                   |
| AAV             | **Action: Advanced Set Variables - event type** panel  
**AAV** command is valid only when issued from the **AV** panel. | Use this panel to set, increment, or decrement one or more SHARED variables and use powerful built-in Rules functions with variables.  
Use of this panel is optional.                                                                                                         |
| AA              | **Action(s) I - event type** panel | This panel applies only to the Rules that 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. |
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 resolves to the name of event type that you specified in the Event Type field on the Rule Processor Detail Control panel.

The Rules Processor provides up to three panels that you can use to specify selection criteria. They are the Selection Criteria (S1) panel, the optional Variable Dependencies (SV) panel, and the optional Advanced Variable Dependencies (ASV)
The Rules Processor uses AND logic to combine any and all of the selection criteria from the three panels to collect the selection criteria for the Rule. For example, if criteria is specified on the S1 and ASV panels, all of the criteria on the S1 panel must match and all of the criteria specified on the ASV panel must match before the Rule fires.

For each event type, a customized selection criteria panel for that event type is displayed and contains a subset of the criteria listed in “Selection Criteria fields” on page 279.

For information about the selection criteria panel for TIME-initiated Rules and an example about how to create a Rule for a TIME event, refer to “Creating TIME-initiated Rules” on page 119.

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

![Selection Criteria Panel Example](image)

**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 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 specify the following kinds of variables on this panel for the Rule to use as selection criteria before it fires:
- Any MainView AutoOPERATOR-supplied EVENT variables that are listed in "List of variables for ALRT-initiated Rules" on page 183.

- Any SHARED or LOCAL variable.

Refer to the chapter describing variables in the MainView AutoOPERATOR Advanced Automation Guide for information regarding SHARED and LOCAL variables. Refer to “Using variables in the Rule Processor” on page 160 for more information about using variables in the Rule Processor applications.

- The variable \textit{WORD}_n, where \( n \) is the number of the word in the message. For example, \textit{WORD}3 represents the third word in the message text.

The following figure shows an example of the Variable Dependencies panel.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{VariableDependenciesPanel.png}
\caption{Variable Dependencies panel}
\end{figure}

In the preceding figure, the QSMFID variable is specified to equal the value SYSB or SYSC, which 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.

\textbf{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.

In addition, when you enter numeric values to be compared on the Variable Dependencies panel, the Rule Processor compares only whole numbers (0-9 digits). For example, the value 1.11 is considered text. When comparing numerics, the values are restricted to 15 digits.
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.

The following figure shows an example.

---

**Figure 26: Variable Dependencies panel with AND/OR logic**

<table>
<thead>
<tr>
<th>Variable-name</th>
<th>Op</th>
<th>Variable-Value</th>
<th>AND/</th>
<th>OR/</th>
</tr>
</thead>
<tbody>
<tr>
<td>QSMFID</td>
<td>EQ</td>
<td>SYSA</td>
<td>AND</td>
<td></td>
</tr>
<tr>
<td>IMFDAY</td>
<td>EQ</td>
<td>TUE</td>
<td>AND</td>
<td></td>
</tr>
<tr>
<td>WORD3</td>
<td>EQ</td>
<td>DUMP</td>
<td>AND</td>
<td></td>
</tr>
<tr>
<td>QSMFID</td>
<td>EQ</td>
<td>SYSB</td>
<td>AND</td>
<td>OR</td>
</tr>
<tr>
<td>IMFDAY</td>
<td>EQ</td>
<td>WED</td>
<td>OR</td>
<td></td>
</tr>
<tr>
<td>QSMFID</td>
<td>EQ</td>
<td>SYSC</td>
<td>OR</td>
<td></td>
</tr>
</tbody>
</table>

---

This information translates into the following statements:

\[
\text{[QSMFID=SYSA) AND (IMFDAY=TUE) AND (WORD3=DUMP)] OR [QSMFID=SYSB) AND (IMFDAY=WED)] OR (QSMFID=SYSC)}
\]

---

Using the Advanced Variable Dependencies (ASV) panel

Use the Advanced Variable Dependencies (ASV) panel to create tests and equations to further qualify the event before it is automated by a Rule.
You can enter up to 512 characters of equation or comparison. You can also use the following operators:

<table>
<thead>
<tr>
<th>Enter</th>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>plus sign</td>
<td>add</td>
</tr>
<tr>
<td>-</td>
<td>minus sign</td>
<td>subtract</td>
</tr>
<tr>
<td>*</td>
<td>asterisk</td>
<td>multiply</td>
</tr>
<tr>
<td>/</td>
<td>forward slash</td>
<td>divide</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>=</td>
<td>equal sign</td>
<td>equal</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>not equal</td>
<td>not equal</td>
</tr>
<tr>
<td>&lt;</td>
<td>less than</td>
<td>less than</td>
</tr>
<tr>
<td>&lt;=</td>
<td>less than or equal</td>
<td>less than or equal</td>
</tr>
<tr>
<td>&gt;</td>
<td>greater than</td>
<td>greater than</td>
</tr>
<tr>
<td>&gt;=</td>
<td>greater than or equal</td>
<td>greater than or equal</td>
</tr>
<tr>
<td>()</td>
<td>used for grouping</td>
<td>parentheses</td>
</tr>
</tbody>
</table>

All variable names that are used on the ASV panel must be preceded with an ampersand (&) to be considered a variable. The following figure shows an example of the ASV. In addition, you can use built-in Rules functions on this panel. See “Using Rules functions” on page 185 for more information.

**Figure 27: Advanced Variable Dependencies (ASV) panel**

BMC Software ------ Advanced Variable Dependencies - MSG ------ AutoOPERATOR COMMAND ===> TGT --- DL72

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

Enter equation(s) and/or comparison(s) below (512 character maximum)

\[ ((\&QSMFID=SYSA) \text{ AND } (\&IMFDAY=TUE) \text{ AND } (\&WORD3=DUMP)) \]

OR

\[ ((\&QSMFID=SYSB) \text{ AND } (\&IMFDAY=WED)) \]

OR

\[ (\&QSMFID=SYSC) \]

Notes: Operators include +, -, *, /, and || (concatenate).

Common comparators include =, <>, <, >, <=, and >=

All variables must be coded with a leading ampersand (&)

Use HELP (PF1) for more details.

Press END to return to SV with changes or CANCEL to cancel changes.
Step 3. Specifying a Rule's actions with the Action Specification fields

The third and fourth panels displayed during Rule creation are the Action Specification(s) I and II --event type panels.

The event type field resolves to the name of the event that you specified in the Event Type field on the Rule Processor Detail Control panel.

For each event type, a customized action specification panel for that event type is displayed and contains a subset of the actions listed in “Describing Action Specification fields” on page 298.

The following figures show examples of the Action Specification(s) panels I and II for a MSG-initiated Rule.

**Figure 28: Action Specification(s) I - MSG panel**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Action Specification(s) I - MSG</th>
<th>TGT</th>
<th>AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule-set</td>
<td>RULIMSOO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rule-id</td>
<td>AUTHIMSI</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Display at dest. -> YES
- Journal -> SYSLOG
- Display -> Aux Log
- EXEC Name/Parms
- Reword Msg
- Issue Msg (WTO)
- DOM Message in (Yes/No)
- POST Target
- POST ID
- POST Parms

* Enter a question mark (?) and blank in any field on the line for more options. Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes.

**Figure 29: Action Specification(s) II - MSG panel**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Action Specification(s) II - MSG</th>
<th>TGT</th>
<th>DL72</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule-set</td>
<td>RULAO72</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rule-id</td>
<td>CTMXSDF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Send (TSO IDs)
- DOM Id
- DOM Id
- Console Name
- Route codes
- Descriptor codes
- Update Rout/Desc Codes (ADD/REPL)
- Color
- Hilite
- Intensity
- Set IOA Variable
- Value

* Enter a question mark (?) and blank in any field on the line for more options. Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes.
Step 4. Setting variables as additional action specification

The following figure shows an example of the Action: Set Variables panel.

You can use the fields on this panel to define a Rule’s actions to do the following:

- set one or more SHARED variables
- increase or decrease a SHARED variable’s numerical value
  To increase or decrease the numerical value, specify \(+n\) or \(-n\), where \(n\) is a number.
- use variable substringing
Refer to “Using variables in Rule Processor application fields” on page 164 for more information about using variables in Rules and using the Action: Set Variables panel.

**Figure 30: Action: Set Variables - MSG panel**

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Variable Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MYVAR1</td>
<td>10</td>
</tr>
<tr>
<td>MYVAR2</td>
<td>-15</td>
</tr>
<tr>
<td>MYVAR3</td>
<td>+20</td>
</tr>
</tbody>
</table>

--> ATTENTION: Use AAV command to use Advanced Set Variables panel <--

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

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 43 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.

**Using the Action: Advanced Set Variables (AAV) panel**

Use the Action: Advanced Set Variables panel (AAV) to set, increment, or decrement one or more SHARED variables.

All event types support this feature. This panel allows you to use built-in Rules functions and equations in the Variable-Value column. The following figure shows an example of the AAV panel.

**Figure 31: Action: Advanced Set Variables (AAV) panel**
The following table describes the two fields:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable-Name</td>
<td>Specify the name of SHARED variable to be set. The maximum length is 33 characters. SHARED variables can be used to form the name. You can also use substringing.</td>
</tr>
<tr>
<td>Variable-Value</td>
<td>Specify the value used to set variable. The value can be a constant, can contain other variables, or be a signed number to increment or decrement the value. You can also use built-in Rules functions, math equations, and constants in this field. The maximum length allowed is 43 characters.</td>
</tr>
</tbody>
</table>

In the following example, the first variable, SYSABENDS, is a counter and it will be increased by one. The next uses variable &WORD3 to help form the variable name. The next variable uses built-in Rules functions STRIP() and X() to change "PLEX001" to PLEX001. The last uses math to create the variable's value.

**Figure 32: Action: Advanced Set Variables example**

<table>
<thead>
<tr>
<th>Variable-Name</th>
<th>Variable-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSABENDS</td>
<td>+1</td>
</tr>
<tr>
<td>ERR&amp;WORD3:3:4</td>
<td>'AN ERROR HAS OCCURED'</td>
</tr>
<tr>
<td>MYSYSTEM</td>
<td>STRIP(&amp;SYSPLEX,B,X(7F))</td>
</tr>
<tr>
<td>WORKRATE</td>
<td>((&amp;WORD4+&amp;WORD9+1)/3)+AVG(&amp;WORD7,&amp;WORD8)</td>
</tr>
</tbody>
</table>

You can also use the following operators:

<table>
<thead>
<tr>
<th>Enter</th>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>plus sign</td>
<td>add</td>
</tr>
<tr>
<td>-</td>
<td>minus sign</td>
<td>subtract</td>
</tr>
<tr>
<td>*</td>
<td>asterisk</td>
<td>multiply</td>
</tr>
<tr>
<td>/</td>
<td>forward slash</td>
<td>divide</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>=</td>
<td>equal sign</td>
<td>equal</td>
</tr>
</tbody>
</table>
Step 5. Creating ALERTs as additional action specification

Occasionally, you might want a Rule to generate a MainView AutoOPERATOR ALERT as part of the actions a Rule takes.

MainView AutoOPERATOR ALERTs are special exception messages that MainView 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 panels; these panels are as follows:

- **Alert Action(s) I - event type**
  This panel 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 that is 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 panel is the secondary ALERT action specification 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.

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; for information about specific fields, refer to “Alert Actions I and II fields” on page 321.

Creating ALERTs for all event types except ALRT events

The following figure shows an example of the Alert Action(s) I Specification panel that is displayed for all event types except ALRT.

**Figure 33: ALERT action(s) I: primary panel for all events except ALRT**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Alert Action(s) I</th>
<th>MSG</th>
<th>MAINVIEW</th>
<th>AutoOPERATOR</th>
<th>COMMAND</th>
<th>TGT</th>
<th>AOAO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule-set</td>
<td>RULIMSOO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rule-id</td>
<td>AUTHIMS1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(ADD, DELETE, DELETEQ)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

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)

* Enter a question mark(?) and blank for more options.
Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes

This panel shows three actions for the **Function** field:

ADD, DELETE, 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.

**Figure 34: ALERT Action(s) I specification panel for the ALRT event type**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Alert Action(s) I</th>
<th>ALRT</th>
<th>MAINVIEW</th>
<th>AutoOPERATOR</th>
<th>COMMAND</th>
<th>TGT</th>
<th>AOAO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule-set</td>
<td>RULIMSOO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rule-id</td>
<td>AUTHIMS1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(ADD, DELETE, DELETEQ, MODIFY)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

Key ===> 

Text ===> 

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes.
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 “(optional) Modifying an ALERT in a Rule: using the MODIFY function” on page 100.

The following figure shows an example of the secondary panel you can use to create ALERTs for all event types.

**Figure 35: ALERT Action(s) II: secondary ALERT creation panel**

![Example Secondary ALERT Creation Panel](image)

The following figure shows an example of the Alert Escalation Final Action Type panel that is displayed when you enter a question mark in the Final Act ( ) field.

**Specify a Final Action for an ALERT**

The following figure shows an example of the Alert Escalation Final Action Type panel that is displayed when you enter a question mark in the Final Act ( ) field.
Use this panel to select the final action that an ALERT should take when it reaches its final priority level and display additional panels where you can specify the action and any additional associated keywords.

**Figure 36: Alert Escalation Final Action Type panel**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>REXX Exec or CLIST</td>
</tr>
<tr>
<td>MVS</td>
<td>MVS/JES command</td>
</tr>
<tr>
<td>BBI</td>
<td>BBI Control command</td>
</tr>
<tr>
<td>IMS</td>
<td>IMS command</td>
</tr>
<tr>
<td>IMP</td>
<td>IMS Plex command</td>
</tr>
<tr>
<td>CICS</td>
<td>CICS command</td>
</tr>
<tr>
<td>MQ</td>
<td>WebSphere MQ command</td>
</tr>
<tr>
<td>NV</td>
<td>NetView command</td>
</tr>
<tr>
<td>TOM</td>
<td>Total Object Manager command</td>
</tr>
</tbody>
</table>

Command:

Press ENTER to continue filling in command. END to return to Alert Actions with selected command type, CANCEL to cancel selection.

The following figure shows an example of the REXX Exec or CLIST panel. This panel is displayed when you select EXEC from the preceding figure.

**Figure 37: REXX Exec or CLIST panel**

priority and Delay keywords apply to all execs

**Figure 38: MVS Command(s) from BBI-SS panel**

The following figure shows an example of the MVS Commands from BBI-SS panel. This panel is displayed when you select MVS from the preceding figure.
Where to find more information about ALERTs

To read more about ALERTs, refer to:

- *MainView AutoOPERATOR Basic Automation Guide*, Volume 2 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*.

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Contents</th>
<th>Length/Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMFKEY</td>
<td>key of the ALERT</td>
<td>1-64 / character</td>
</tr>
<tr>
<td>AMFTEXT</td>
<td>text of the ALERT</td>
<td>0-255 / character</td>
</tr>
<tr>
<td>AMFALARM</td>
<td>alarm value of the ALERT</td>
<td>1 / Y (YES) or N (NO)</td>
</tr>
<tr>
<td>AMFCOLOR</td>
<td>color of the ALERT</td>
<td>6 / As specified by COLOR parameter</td>
</tr>
<tr>
<td>AMFCURIN</td>
<td>current interval number when the AMEVENT variable value is ESC or FINAL. This variable is not created for other ALERT event sub-types (such as ADD, DELETE, and so on).</td>
<td>1 / character 1-5 (normal escalation interval) or &quot;F&quot; (final escalation)</td>
</tr>
<tr>
<td>AMFEDIR</td>
<td>increase or decrease the priority of the ALERT when it is escalated</td>
<td>1 / character (U or D)</td>
</tr>
</tbody>
</table>

Refer to “Alert Action(s) II fields” on page 324 for more information about the fields on these panels.
<table>
<thead>
<tr>
<th>Name</th>
<th>Contents</th>
<th>Length/Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMFEDISP</td>
<td>keep or delete the ALERT at the final escalation level</td>
<td>1 / character (K or D)</td>
</tr>
<tr>
<td>AMFEXEC</td>
<td>name of EXEC and EXEC parameters scheduled at final escalation priority</td>
<td>0 to 255 / character</td>
</tr>
<tr>
<td>AMFEINT1</td>
<td>number (in minutes) from 0 to 9999</td>
<td>4 / numeric (or null)</td>
</tr>
<tr>
<td>AMFEINT2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMFEINT3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMFEINT4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMFEINT5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMFEINT6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMFEVENT</td>
<td>type of ALERT event (for example, DELETEQ)</td>
<td>3-7 / ADD, DELETE, DELETEQ, ESC, FINAL</td>
</tr>
<tr>
<td>AMFHELP</td>
<td>extended ALERT member name</td>
<td>8 / character</td>
</tr>
<tr>
<td>AMFDATE</td>
<td>date ALERT was issued</td>
<td>9 / dd-mmm-yy</td>
</tr>
<tr>
<td>AMFTIME</td>
<td>time ALERT was issued</td>
<td>8 / hh:mm:ss</td>
</tr>
<tr>
<td>AMFORGN</td>
<td>origin of ALERT</td>
<td>1-8 / character</td>
</tr>
<tr>
<td>AMFPCMD</td>
<td>primary command specified in ALERT</td>
<td>0-256 / character</td>
</tr>
<tr>
<td>AMFPTG1</td>
<td>names of the cells in the TGTPUB keyword, or the TARGETS field from the Rule panel</td>
<td>1 - 8 / character</td>
</tr>
<tr>
<td>AMFPTG2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMFPTG3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMFPTG4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMFPTG5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMFPTG6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMFPRIOR</td>
<td>priority of ALERT</td>
<td>13 / As specified in PRIORITY parameter</td>
</tr>
<tr>
<td>AMFPSYS</td>
<td>value for SYSTEM keyword (could be either YES or NO)</td>
<td>1 / character (Y or null)</td>
</tr>
<tr>
<td>AMFPUB</td>
<td>value of the PUBLISH keyword when an ALERT is created</td>
<td>2-7/ADD, REPLACE, or NO</td>
</tr>
<tr>
<td>AMFPUBS</td>
<td>value of the PUBLISH keyword as specified in the request to create the ALERT</td>
<td>2-7/ADD, REPLACE, or NO</td>
</tr>
<tr>
<td>AMFQUEUE</td>
<td>name of queue for ALERT</td>
<td>8 / character</td>
</tr>
<tr>
<td>AMFRtain</td>
<td>specifies whether to retain an ALERT across BBI-SS PAS warm and cold starts</td>
<td>1 / character (Y or N)</td>
</tr>
<tr>
<td>AMFSSID</td>
<td>system from which ALERT was issued</td>
<td>8 / character</td>
</tr>
</tbody>
</table>
(optional) 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 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.

<table>
<thead>
<tr>
<th>Fields whose values are deleted</th>
<th>Fields whose values are reset to default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exec</td>
<td>Alarm</td>
</tr>
<tr>
<td>Help</td>
<td>Color</td>
</tr>
<tr>
<td>Interval</td>
<td>Dispose</td>
</tr>
<tr>
<td>PCMD</td>
<td>Escalate</td>
</tr>
<tr>
<td>Text</td>
<td>Function</td>
</tr>
<tr>
<td>Udata</td>
<td>Origin</td>
</tr>
<tr>
<td>User</td>
<td>Priority</td>
</tr>
<tr>
<td></td>
<td>Retain</td>
</tr>
<tr>
<td></td>
<td>System</td>
</tr>
</tbody>
</table>

### If you change a field related to escalation

If any ALERT field related to escalation is modified, all escalation parameters must be specified again.

*Note*

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 130 for an example of how you can use the MODIFY function.

If you have the BMC Software product MainView Alarm Manager, you can view MainView AutoOPERATOR ALERTs from the Alert Management application (Option 1). Refer to the MainView Alarm Manager User Guide for more information.

(optional) Step 6. Sending events to a BMC Impact Manager cell

Specifying BA displays the Event Action BEIM - event type panel which allows you to send events from MainView AutoOPERATOR to a BMC Impact Manager cell or other BMC products with a cell adaptor.

Note

This panel appears only if you have specified PRODUCT=IIZ in BBPARM member BBISSP00.

For more information about customization steps that must be completed to send MainView AutoOPERATOR events to a BMC Impact Manager cell, refer to Chapter 19, “Implementing the MainView AutoOPERATOR Interface to BMC Impact Manager” in the MainView AutoOPERATOR Customization Guide.

Use this panel to:

- specify event data that the Rule sends to a BMC Impact Manager cell or other BMC products with a cell adaptor
- specify whether the Rule write event diagnostic information to the BBI Journal
- specify targets to the send the event to
- specify a new class name to be associated with the slots and values
- specify a severity to be associated with the event

Any event type can be sent to a BMC Impact Manager cell or other BMC products with a cell adaptor except WebSphere MQ non-event type messages. WebSphere MQ instrumentation events can be sent.
specify a component alias for event correlation against objects in the Service Impact Model (SIM)

For information about fields on this panel, refer to “Event Action — BEIM fields” on page 331.

The following figure shows an example of the panel for the event type MSG.

**Figure 39: Event Action - BEIM panel**

Rules can be used to enrich ALERTs that a Rule sends to a BMC Impact Manager cell in the following ways:

- Add a new ALERT: when you specify that a Rule sends an ALERT to a cell, the Alert Action III BEIM panel provides input fields where you can enrich the ALERT information

- Modify an ALERT: when an ALERT-initiated Rule modifies an existing ALERT, the ALERT being modified can also contain enriched information from the Alert Action III BEIM panel

*(optional) Step 7. Enriching ALERTs that are sent to a BMC Impact Manager cell*
For more information about customization steps that must be completed to send MainView AutoOPERATOR events to a BMC Impact Manager cell, refer to Chapter 17, "Implementing the MainView AutoOPERATOR Interface to BMC Impact Manager" in the MainView AutoOPERATOR Customization Guide.

Use the Alert Action III BEIM panel to specify:

- whether details for the ALERT should be written to the BBI Journal
- specify targets where the ALERT will be published
- a new class name to be associated with the slots and values
- a "Component Alias" for event correlation against objects in the Service Impact Model (SIM)
- additional information about this Rule's BMC II for z/OS slots

For information about fields on this panel, refer to “Alert Action III BEIM - panel fields” on page 337.

The following figure shows an example of the panel for the event type MSG.

**Figure 40: Alert Action III BEIM panel**

An ALRT-initiated Rule can replace the targets that were specified in the original ALERT; for example, if an ALERT was created with a target name of MODEL-1, the Rule can override that value by specifying the following values on this panel:

```
Target(s) ===> MODEL*
```
An ALRT-initiated Rule can also add to the targets that were specified in the original ALERT; for example:

<table>
<thead>
<tr>
<th>Target(s)</th>
<th>&amp;AMFPTG1 MODEL*</th>
</tr>
</thead>
</table>

(optional) Step 7. Enriching ALERTs that are sent to a BMC Impact Manager cell
Step 7. Enriching ALERTs that are sent to a BMC Impact Manager cell
Examples of creating Rules

The *MainView AutoOPERATOR for WebSphere MQ Installation and User Guide* contains information about how to create Rules for WebSphere MQ events.

Creating a Rule to suppress a message

The following section 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 job name. 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, follow these steps:

1. From the Automation Control panel, use the Enable line command to enable the Rule Set to which you want to add the Rule.

   You can add Rules only to enabled Rule Sets.

   ![Figure 41: Creating a suppression Rule (1): Automation Control panel](image)

   **Figure 41: Creating a suppression Rule (1): Automation Control panel**

   BMC Software --------------- Automation Control --------------- RULMAGIE ENABLED
   COMMAND ===> TGT ===> A0A0
   Primary commands: ADD, STATSHOW, CMDSHOW, PARMSHOW, Locate,
   DATE --- YY/MM/DD
   EXPAND                                      TIME --- 17:48:38
   Automation Status     ===> ACTIVE         (Active, Inactive)
   Automation Strategy   ===> INDIVIDUAL     (Individual, All, First)
The Rule Set is enabled.

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

**Figure 42: Creating a suppression Rule (1): Automation Control panel**

```plaintext
BMC Software ---------------- Automation Control ------------ RULMAGIE ENABLED
COMMAND ===>                                                 TGT ===> AOAO
Primary commands: ADD, STATSHOW, CMDSHOW, PARMSHOW, Locate, DATE --- YY/MM/DD
EXPAND                                      TIME --- 17:48:38
Automation Status ====> ACTIVE         (Active, Inactive)
Automation Strategy ====> INDIVIDUAL     (Individual, All, First)
Honor MPF Suppression ====> NO             (NO/YES)

Automation Statistics

| Total Events | Display suppressed | 48,538 | 67 |
| Events Handled | Hardcopy suppressed | 22,407 | 0 |
| Current arrival rate | Rule generated Alerts | 2 / sec | 22,223 |
| Peak arrival rate | Rule invoked Execs | 99 / sec | 276 |
```

**Automation Library**

<table>
<thead>
<tr>
<th>LC</th>
<th>Rule-Set Status</th>
<th>Rules</th>
<th>Fired</th>
<th>Filtered</th>
<th>Date</th>
<th>Time</th>
<th>Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>___</td>
<td>AAORULMY ENABLED</td>
<td>15</td>
<td>22,223</td>
<td>48,573</td>
<td>DD-MMM-YY 08:18:59</td>
<td>FIRST</td>
<td></td>
</tr>
<tr>
<td>___</td>
<td>AAORULBC ENABLED</td>
<td>54</td>
<td>188</td>
<td>48,573</td>
<td>DD-MMM-YY 08:18:59</td>
<td>FIRST</td>
<td></td>
</tr>
<tr>
<td>___</td>
<td>AAORULCM ENABLED</td>
<td>52</td>
<td>59</td>
<td>48,573</td>
<td>DD-MMM-YY 08:18:59</td>
<td>FIRST</td>
<td></td>
</tr>
<tr>
<td>__</td>
<td>RULCICS ENABLED</td>
<td>25</td>
<td>1</td>
<td>48,573</td>
<td>DD-MMM-YY 08:19:00</td>
<td>FIRST</td>
<td></td>
</tr>
<tr>
<td>_e</td>
<td>RULMAGIE DISABLED</td>
<td>8</td>
<td>0</td>
<td>38</td>
<td>DD-MMM-YY 14:14:38</td>
<td>FIRST</td>
<td></td>
</tr>
</tbody>
</table>

---

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

**Figure 43: Creating a suppression Rule (1): Rule Set Overview panel**

```plaintext
BMC Software ------------- Rule Set Overview ------------- MAINVIEW AutoOPERATOR
COMMAND ===> add                                                TGT  --- AOAO
Rule Set ID: RULMAGIE    Ruleset Strategy ===> FIRST            DATE --- YY/MM/DD
Primary commands: Add, Save, Sort, Unsort, Reset, Filter        TIME --- HH:MM:SS
LC CMDS --- (S)elect, (BR)owse, (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, GO
Rule Set Desc.: Sort Criterion: Scroll right/left

<table>
<thead>
<tr>
<th>LC</th>
<th>Rule-id Stat Text-id</th>
<th>Type</th>
<th>Fired EXEC</th>
<th>Changed</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>___</td>
<td>DUMMY001 DIS $HASP*</td>
<td>MSG</td>
<td>0</td>
<td>YY/MM/DD 14:58</td>
<td>BAOMXY2</td>
</tr>
</tbody>
</table>
```

---

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

The Rule Processor Detail Control panel is displayed as shown in the following figure.

```
BMC Software ------ Rule Processor Detail Control ------ MAINVIEW AutoOPERATOR
COMMAND ==>                                                 TGT  --- AAOA
TIME --- HH:MM:SS
The following options are displayed in sequence, or may
be selected by entering the two-character code

S1 - Selection Criteria                   AV - Set Variable(s)
SV - Variable Dependencies               AA - Alert Action(s) I
A1 - Action Specification(s) I           AD - Alert Action(s) II
A2 - Action Specification(s) II          BA - Event Action(s) - BEIM
     AE - Alert Action(s) III - BEIM

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 ===>          Func ===>          Code ===>          Author ===> BAOMXY2
Desc ===>          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.

The following figure shows adding a new Rule named NEWRULE1 with an event type of MSG.
5 Press Enter; the Selection Criteria - MSG panel is displayed as shown in the following figure.

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

**Figure 44: Creating a suppression Rule (1): Selection Criteria panel**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Selection Criteria - MSG</th>
<th>MAINVIEW AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND ====&gt;</td>
<td>TGT ====&gt; A0AO</td>
<td></td>
</tr>
</tbody>
</table>

```
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 Name ===> Destination Console Name
MLWTO Minor ===> Single/All

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

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

**Figure 45: Creating a suppression Rule (1): Variable Dependencies panel**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Variable Dependencies - MSG</th>
<th>MAINVIEW AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND ====&gt;</td>
<td>TGT ====&gt; A0AO</td>
<td></td>
</tr>
</tbody>
</table>

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

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

Issuer Identification:
Job name ===> BAO*
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 Name ===> Destination Console Name
MLWTO Minor ===> Single/All

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

7 Press Enter; the Variable Dependencies - MSG panel is displayed as shown in the following figure.
This example Rule does not use variables: press Enter; the Action Specification(s) I - MSG panel is displayed.

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

- 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. See Figure 46 on page 111.

**Note**

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

---

**Figure 46: Creating a suppression Rule (1): Action Specification I panel**

8 This example Rule does not use variables: press Enter; the Action Specification(s) I - MSG panel is displayed.

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

<table>
<thead>
<tr>
<th>BMC Software ------</th>
<th>Action Specification(s) I - MSG ------</th>
<th>MAINVIEW AutoOPERATOR COMMAND ===&gt;</th>
<th>TGT --- AOAO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule-set === RULMAGIE</td>
<td>Rule-id === NEWRULE1</td>
<td>Display at dest. ===&gt; YES</td>
<td>Journal ===&gt; YES</td>
</tr>
</tbody>
</table>

* Exec Name/Parms ===>

* Cmd(Type MVS ) ===>>

* Issue Msg( WTO ) ===>>

DOM Message ===> (Yes/No)

POST Target ===>>

POST ID ===>>

POST Parms ===>>

* Enter a question mark(?) and blank in any field on the line for more options.

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes.
9 Press **Enter**; the Action Specification(s) II - MSG panel is displayed as shown in the following figure.

**Figure 47: Creating a suppression Rule (1): Action Specification II panel**

```
BMC Software  ---------- Action Specification(s) II - MSG  ------- AutoOPERATOR
COMMAND ===>                                                 TGT --- DL72
Rule-set === RULMAGIE           Rule-id  === NEWRULE1
Send (TSO IDs) ===> BAOMES1
DOM Id ===> Delete Operator Message
Console Name ===> Dest. Console ID Name
Rule-set === RULMAGIE           Rule-id  === NEWRULE1
Send (TSO IDs) ===> BAOMES1
DOM Id ===> Delete Operator Message
Console Name ===> Dest. Console ID Name
```

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

10 Specify that a notification is sent to a TSO user ID, BAOMES1, in the **Send (TSO IDs)** field.

**Figure 48: Creating a suppression Rule (1): Action: Set Variables - MSG panel**

```
Variable-Name                      Variable-Value(functions/equations allowed)
_________________________________  ___________________________________________
_________________________________  ___________________________________________
_________________________________  ___________________________________________
_________________________________  ___________________________________________
_________________________________  ___________________________________________
_________________________________  ___________________________________________
_________________________________  ___________________________________________
_________________________________  ___________________________________________
_________________________________  ___________________________________________
```

11 Press **Enter**; the Action: Set Variables - MSG panel is displayed as shown in the following figure.

**Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes**
This Rule will not set variables; press Enter and the Alert Action(s) I - MSG panel is displayed as shown in the following figure.

**Figure 49: Creating a suppression Rule (1): Alert Action(s) I panel**

13 This Rule will not issue an ALERT; press Enter and the Alert Action(s) II - MSG panel is displayed as shown in the following figure.

**Figure 50: Creating a suppression Rule (1): Alert Action(s) II panel**

14 Press Enter until the Rule Processor Detail Control panel is redisplayed.
Creating a Rule to suppress a message

15 Press END to apply changes; the Rule Set Overview panel is displayed as shown in the following figure.

**Figure 51: Creating a suppression Rule (1): Rule Set Overview panel**

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.

16 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:

**Figure 52: Creating a suppression Rule (1): Confirming Rule Set Modifications panel**

Creating a Rule to suppress a message
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.

The options are as follows:

- 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, follow these steps.

**To create this Rule**

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

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

The Rule Processor Detail Control panel is displayed.

2. Enter the new Rule ID and the event type on the Rule Processor Detail Control panel.
This example adds a new Rule named NEWRULE2 with an event type of MSG.

3 Press Enter; the Selection Criteria - MSG panel is displayed as shown in the following figure.

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

**Figure 53: Creating a suppression Rule (2): Selection Criteria panel**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Selection Criteria - MSG</th>
<th>MAINVIEW AutoOPERATOR</th>
<th>TGT</th>
<th>A0AO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule-set</td>
<td>RULMAGIE</td>
<td>Rule-id</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text Description:</td>
<td></td>
<td>Ignore leading plus: Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text ID</td>
<td>IEF450I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text String (Enter Below):</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 Name: Destination Console Name
- MLWTO Minor: Single/All

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

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

5 Press Enter; the Variable Dependencies - MSG panel is displayed as shown in the following figure.

**Figure 54: Creating a suppression Rule (2): Variable Dependencies panel**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Variable Dependencies - MSG</th>
<th>MAINVIEW AutoOPERATOR</th>
<th>TGT</th>
<th>A0AO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule-set</td>
<td>RULMAGIE</td>
<td>Rule-id</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable-name</th>
<th>Op</th>
<th>Variable-Value</th>
<th>OR/AND</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

--> ATTENTION: Use ASV command to add ASV criteria to the rule <--

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

6 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.

--- ATTENTION: Use ASV command to add ASV criteria to the rule <---
Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes

The following list 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.

7 Press *Enter*; the Action Specification(s) I - MSG panel is displayed as shown in the following figure.

**Figure 55: Creating a suppression Rule (2): Action Specification I panel**
8 Specify the actions that the Rule will take when it is fired.

To specify that the Rule is suppressed from the console and a message is written to the BBI-SS PAS Journal:

- Enter NO for **Display at dest.** and the IEF450I message will be suppressed from 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.

9 Press Enter; the Action Specification(s) II - MSG panel is displayed as shown in the following figure.

**Figure 56: Creating a suppression Rule (2): Action Specification II panel**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Action Specification(s) II - MSG</th>
<th>AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND ====&gt;</td>
<td>Rule-set ====&gt; RULMAGIE</td>
<td>Rule-id ====&gt; NEWRULE2</td>
</tr>
<tr>
<td>TGT --- DL72</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Send (TSO IDs) ====&gt; BAOMES1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DOM Id ===&gt;</td>
<td>Delete Operator Message</td>
</tr>
<tr>
<td></td>
<td>Console Name ===&gt;</td>
<td>Dest. Console ID Name</td>
</tr>
<tr>
<td></td>
<td>Route codes ===&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Descriptor codes ===&gt;</td>
<td>(ADD/REPL)</td>
</tr>
<tr>
<td></td>
<td>Color ===&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hilite ===&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Intensity ===&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Update Rout/Desc Codes ===&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Send ==&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Desc-Set ==&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Update Rout/Desc Codes ===&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Color ===&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Intensity ===&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Set IOA Variable ===&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Value ===&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CTM Force Job: Table...: Job.: Date:</td>
<td>Library:</td>
</tr>
<tr>
<td></td>
<td>CTM Set Cond.: Name...: Date: Action:</td>
<td></td>
</tr>
</tbody>
</table>

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

10 Specify the user ID, BAOMES1, in the **Send (TSO IDs)** field and a message will be sent to the user ID stating that the Rule has fired.

11 Press **Enter**; the Action: Set Variables - MSG panel is displayed.

12 This Rule will not set variables or issue an ALERT; press Enter until the Rule Processor Detail Control panel is re-displayed.

13 Press **END** to apply changes; the Rule Set Overview panel is displayed.

14 Repeat **Step 15 on page 114** and **Step 16 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.

For information about how your automation with time-initiated Rules can be affected when you change the MVS Local Time to observe daylight savings time, refer to “Field descriptions for the Rule Processor” on page 279.

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 that 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.

2 Press Enter; the Selection Criteria - TIME panel is displayed as shown in the following figure.

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

Figure 57: Creating a TIME-initiated Rule (1): Selection Criteria panel

3 Enter the time specifications as follows:

- Start Time: 00:00:01
- Interval: 00:15:00
- Stop Time: 24:00:00

Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes.
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 that a Rule can begin firing is one second after midnight.

To specify that the Rule begins firing at midnight, specify 00:00:01 for the **Start Time** field.

To specify the Rule should fire all day and all night, specify 24:00:00 for the **Stop time** field.

Do not specify anything in the **Stop Count** field because you already specified a time in 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 AU9999I) 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.

---

**Note**

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 behaves as though the Rule fired on the Start Time and every interval thereafter.

4 Press **Enter**; the Variable Dependencies - TIME panel is displayed.

5 This example does not make use of variables; press Enter and the Action Specification(s) I - TIME panel is displayed as shown in the following figure.

**Figure 58: Creating a TIME-initiated Rule (1): Action Specification I panel**
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 causes the initiators to be displayed in the SYSLOG.

This command will be issued every 15 minutes.

Press Enter; the Action Specification(s) - II TIME panel is displayed as shown in the following figure.

**Figure 59: Creating a TIME-initiated Rule (1): Action Specification II panel**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Action Specification(s) II - TIME -------</th>
<th>AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND</td>
<td>===&gt;</td>
<td>TGT --- DL72</td>
</tr>
<tr>
<td>Rule-set</td>
<td>===&gt;</td>
<td>Rule-id ===&gt; CTMXSDF</td>
</tr>
<tr>
<td>Send (TSO IDs)</td>
<td>===&gt;</td>
<td>Delete Operator Message</td>
</tr>
<tr>
<td>DOM Id</td>
<td>===&gt;</td>
<td>Date:</td>
</tr>
<tr>
<td>Set IOA Variable</td>
<td>===&gt;</td>
<td>Action:</td>
</tr>
<tr>
<td>Value</td>
<td>===&gt;</td>
<td>Date:</td>
</tr>
<tr>
<td>CTM Force Job: Table...: Job.: Date:</td>
<td>Library:</td>
<td>Action:</td>
</tr>
<tr>
<td>CTM Set Cond.: Name...: Date: Action:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

Specify that a notification is sent to a TSO user ID, BAOMESI, in the Send (TSO IDs) field.

The Rule will issue the command and a notification will be sent to TSO user IDBAOMES1 every 15 minutes.

The Rule will not set variables or issue an ALERT; press Enter until the RuleProcessor Detail Control panel is redisplayed.

Press END to apply changes; the Rule Set Overview panel is displayed.

Repeat Step 15 on page 114 and Step 16 on page 114 to complete saving the Rule NEWRULE3 to Rule Set RULMAGIE.

---

**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 will not begin firing until the next BBI-SS PAS startup. The action that the Rule takes is to display initiators.
TIME-initiated Rules that use ATSTART are automatically suspended or disabled if the Rule is modified. When you modify TIME-initiated Rules that uses ATSTART, a confirmation panel is displayed that warns you about suspending or disabling the Rule, and allows you to cancel or proceed with the changes.

To create this Rule

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

2. Press Enter; the Selection Criteria - TIME panel is displayed as shown in the following figure.

   **Figure 60: Creating a TIME-initiated Rule (2): Selection Criteria panel**

   - Rule-set: RULMAGIE
   - Rule-id: NEWRULE4
   - Command: MAINVIEW AutoOPERATOR
   - TGT: AOAO
   - TIME: HH:MM:SS
   - Start Specification: ATSTART
   - Interval: 00:15:00
   - Stop Specification: HH:MM:SS or 10

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

3. Specify the start time for the Rule to fire and the time interval; 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 (valid values are 1 to 99999999).

   The Rule will stop firing after 10 times and will not fire again until the next BBI-SS PAS startup.

   For more information about the parameters on the Selection Criteria - TIME panel, see “Selection Criteria fields” on page 279

4. Press **Enter**; the Variable Dependencies - TIME panel is displayed.
5 This example does not make use of variables; press Enter.

The Action Specification - TIME panel is displayed as shown in the following figure.

**Figure 61: Creating a TIME-initiated Rule (2): Action Specification I panel**

---

- BMC Software
- Action Specification(s) I - TIME
- AutoOPERATOR
- COMMAND: $DI
- Rule-set: RULMAGIE
- Rule-id: NEWRULE4
- Automation Actions:
  - Journal
  - Aux Log
  - Exec Name/Parms
  - Issue Msg
  - POST Target
  - POST ID
  - POST Parms

* Enter a question mark(?) and blank in any field on the line for more options.
* Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes

6 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 causes the initiators to be displayed in the SYSLOG.

7 Press Enter; the Action Specification(s) - II MSG panel is displayed as shown in the following figure.

**Figure 62: Creating a TIME-initiated Rule (2): Action Specification II panel**

---

- BMC Software
- Action Specification(s) II - MSG
- AutoOPERATOR
- COMMAND: $DI
- Rule-set: RULAO72
- Rule-id: CTMXSDF
- Send (TSO IDs)
- DOM Id
- Set IOA Variable
- Value
- CTM Force Job: Table...
- Library:
- CTM Set Cond.: Name...
- Date:
- Action:

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

8 Specify that a notification is sent to a TSO user ID, BAOMES1, in the Send (TSO IDs) field.

The Rule will issue the command and a notification will be sent to TSO user ID BAOMES1 every 15 minutes.

9 The Rule will not set variables or issue an ALERT; press Enter until the Rule Processor Detail Control panel is redisplayed.
10 Press END to apply changes; the Confirm Timer Rule Update panel is displayed as shown in the following figure.

**Figure 63: Creating a TIME-initiated Rule (1): Confirm Timer Rule Update panel**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>COMMAND</th>
<th>--</th>
<th>Confirm Timer Rule Update</th>
<th>--</th>
<th>AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>TGT</td>
<td>RB64</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

+----------------------------------------------------------------------------------+
+ WARNING! You have added an ATSTART Timer Rule, NEWRULE4. +
+ The NEWRULE4 Rule will be in a SUSPENDED status and will not be eligible to fire until the next time the AutoOPERATOR PAS is stopped and started again. +
+ Please do one of the following: +
+ - Press Enter to continue updating Rule NEWRULE3 in Rule Set RULMAGIE. +
+ - Enter CANCEL to exit this Rule and cancel changes. +
+ - Press END to return to Rule Processor Detail Control for this Rule. +
+----------------------------------------------------------------------------------+

11 Press END to apply changes; the Rule Set Overview panel is displayed.

12 Repeat Step 15 on page 114 and Step 16 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 how to resolve 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.

1 Enter the new Rule ID and the event type on the Rule Processor Detail Control panel.
2 Press **Enter**; the Selection Criteria - MSG panel is displayed as shown in the following figure. Specify the event criteria that the Rule must match to cause the Rule to fire.

**Figure 64: Setting a variable with a Rule: Selection Criteria panel**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Action: Set Variables - MSG</th>
<th>AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>TGT --- AOAO</td>
<td>Rule-set === RULMAGIE</td>
<td>Rule-id === NEWRULE5</td>
</tr>
</tbody>
</table>

- Text Description:
  - Text ID ===> IEF403I
  - Ignore leading plus: Y

- Issuer Identification:
  - 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 Name ===> Destination Console Name
  - MLWTO Minor ===> Single/All

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

3 Enter **IEF403I** (the text-ID) of the message in the **Text ID** field.

4 Press **Enter** until the Action: Set Variables - MSG panel is displayed as shown in the following figure.

**Figure 65: Setting a variable with a Rule: Variable Dependencies panel**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Action: Set Variables - MSG</th>
<th>AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>TGT --- AOAO</td>
<td>Rule-set === RULMAGIE</td>
<td>Rule-id === NEWRULE5</td>
</tr>
</tbody>
</table>

- Variable-Name: a.&imfjtype..&imfjnum = +1

Press **END** to return to AV with changes or **CANCEL** to cancel changes

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

To specify the variable and the actions for this Rule, do the following steps:

- To set a variable, enter the variable name in the **Variable Name** 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. The variable value is increased by one every time the message is seen by the Rule Processor.

6 Press **Enter** until the Rule Processor Detail Control panel is redisplayed.

7 Press **END** to apply changes; The Rule Set Overview panel is displayed.

8 Repeat Step 15 on page 114 and Step 16 on page 114 to complete saving the Rule NEWRULE5 to Rule Set RULMAGIE.

This Rule should be used in conjunction with Rule NEWRULE6 whose creation is described in “Example 2: creating a Rule that fires when a variable value is met” on page 126. 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, follow these steps:

1 Enter the new Rule ID and the event type on the Rule Processor Detail Control panel as shown in the following figure.

**Figure 66: Triggering a Rule with variables: Rule Processor Detail Control panel**

---
To prevent the Rule in this example from over-firing, fill in the Criteria match rate fields as follows. To specify that the Rule will be suspended when it is matched 10 times in one minute:

- Enter **10** (if matched field) for the number of times the Rule might match incoming events

- Enter **60** (in seconds field)

- Enter **SUSPEND** and the Rule will be suspended if the Rule matches an event 10 times or more within one minute (then status field)

Press **Enter**; the Selection Criteria - MSG panel is displayed as shown in the following figure.

Next, specify the event criteria that the Rule must match to cause the Rule to fire.

**Figure 67: Triggering a Rule with variables: Selection Criteria panel**

<table>
<thead>
<tr>
<th>Rule-set</th>
<th>Rule-id</th>
<th>Text Description:</th>
</tr>
</thead>
<tbody>
<tr>
<td>RULMAGIE</td>
<td>NEWRULE6</td>
<td>Ignore leading plus: Y</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Text ID</th>
<th>Ignore leading plus: Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEF403I</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Text String (Enter Below):</th>
</tr>
</thead>
</table>

- Issuer Identification:
  - Job name
  - Type
  - Jobclass
  - Acct Info
  - RACF User
  - RACF Group
  - Route Codes

- Desc Codes

- Console Name

- MLWTO Minor

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

Enter **IEF030I** (the text-ID) of the message in the **Text ID** field.

Press **Enter**; the Variable Dependencies - MSG panel is displayed as shown in the following figure.

**Figure 68: Triggering a Rule with variables: Variable Dependencies panel**

<table>
<thead>
<tr>
<th>Variable-name</th>
<th>Op</th>
<th>Variable-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;!a.imfjtype.imfjnum</td>
<td>gt</td>
<td>12</td>
</tr>
</tbody>
</table>

Creating more complex Rules

Chapter 6  Examples of creating Rules  127
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 list 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.

Press **Enter**; the Action Specifications - MSG panel is displayed as shown in the following figure.

![Figure 69: Triggering a Rule with variables: Action Specification panel](image)

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

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**.
Press Enter until the Alert Actions I - MSG panel is displayed as shown in the following figure.

**Figure 70: Triggering a Rule with variables: Alert Action(s) I panel**

```
BMC Software ---------- Alert Action(s) I - MSG  --------- MAINVIEW AutoOPERATOR
COMMAND ===>                                                 TGT --- A0AO
Rule-set === RULMAGIE             Rule-id  === NEWRULE6
Function ==== add                     (ADD, DELETE, DELETEQ)
Key       ===> &imfday&imfdomid
Text      ===> job &word2 started more than 12 times
```

<table>
<thead>
<tr>
<th>Queues</th>
<th>Alert Queue Name(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>(CRIT, MAJ, MIN, WARN, INFO, CLEAR)</td>
</tr>
<tr>
<td>Color</td>
<td>RED, PINK, YEL, DKBL, LTBL, GRE, WHI</td>
</tr>
<tr>
<td>System</td>
<td>Return to target after PCMD</td>
</tr>
<tr>
<td>Exec</td>
<td>Follow-up EXEC</td>
</tr>
<tr>
<td>Help</td>
<td>Associated HELP Panel</td>
</tr>
<tr>
<td>Targets</td>
<td>Target System</td>
</tr>
<tr>
<td>Udata</td>
<td>User Data</td>
</tr>
<tr>
<td>Origin</td>
<td>Origin</td>
</tr>
<tr>
<td>User</td>
<td>Userid</td>
</tr>
<tr>
<td>Alarm</td>
<td>Sound Alarm (YES/NO)</td>
</tr>
<tr>
<td>Publish</td>
<td>Messages (ADD/REPLACE/NO)</td>
</tr>
</tbody>
</table>

Enter a question mark (?) and blank for more options.
Press ENTER to continue. END return to Detail Control, CANCEL to cancel change

10 To specify that an ALERT is 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 IMFDOMID variables. These variables will resolve to the day of the week and the number of times the Rule was fired. This combination ensures that 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 might be having difficulty starting and might require operator attention.

11 Press **Enter** until the Rule Processor Detail Control panel is redisplayed.

12 Press **END** to apply changes; the Rule Set Overview panel is displayed.

13 Repeat **Step 15 on page 114** and **Step 16 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**, which 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 the 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.&imfjnum 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**

Use the following procedure to create an ALERT retained 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.
2 Press Enter; the Selection Criteria - ALRT panel is displayed as shown in the following figure.

**Figure 71: Modifying an ALERT to survive MVS IPLs: Selection Criteria panel**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Selection Criteria - ALRT</th>
<th>MAINVIEW AutoOPERATOR COMMAND ===&gt;</th>
<th>TGT --- AOAO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule-set === RULMAGIE</td>
<td>Rule-id === NEWRULE7</td>
<td>Text Description:</td>
<td></td>
</tr>
<tr>
<td>Text ID ===&gt; *</td>
<td>First Word of Alert</td>
<td>Text String (Enter Below):</td>
<td></td>
</tr>
<tr>
<td><em>payroll</em>failed*</td>
<td></td>
<td>Key ===&gt;</td>
<td></td>
</tr>
<tr>
<td>Queue ===&gt; Alert Queue</td>
<td>Event ===&gt; ADD, DELETE, DELETEQ, ANY ESC, ALL (default=ADD)</td>
<td>Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes</td>
<td></td>
</tr>
</tbody>
</table>

3 Specify the event criteria that the Rule must match to cause the Rule to fire.

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*.

4 Press Enter until the Action Specification - ALRT panel is displayed as shown in the following figure.

**Figure 72: Modifying an ALERT to survive MVS IPLs: Action Specification I panel**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Action Specification(s) I - ALRT</th>
<th>MAINVIEW AutoOPERATOR COMMAND ===&gt;</th>
<th>TGT --- AOAO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule-set === RULMAGIE</td>
<td>Rule-id === NEWRULE7</td>
<td>Automation Actions:</td>
<td></td>
</tr>
<tr>
<td>Display at dest. ===&gt;</td>
<td>YES</td>
<td>* Exec Name/Parms ===&gt;</td>
<td></td>
</tr>
<tr>
<td>Journal ===&gt;</td>
<td></td>
<td>* Cmd(Type MVS ) ===&gt;</td>
<td></td>
</tr>
<tr>
<td>Aux Log ===&gt;</td>
<td></td>
<td>Reword Alert ===&gt;</td>
<td></td>
</tr>
<tr>
<td>POST Target ===&gt;</td>
<td></td>
<td>* Issue Msg( WTO ) =&gt;</td>
<td></td>
</tr>
<tr>
<td>POST ID ===&gt;</td>
<td></td>
<td>POST Parms ===&gt;</td>
<td></td>
</tr>
</tbody>
</table>

* Enter a question mark(?) and blank in any field on the line for more options. Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes

5 Specify the actions that the Rule takes when it is fired.
Specify **YES** for the Display at dest. field to cause the message to be written to the console.

6 Press **Enter** until the Alert Action(s) I - ALRT panel is displayed as shown in the following figure.

**Figure 73: Modifying an ALERT to survive MVS IPLs: Alert Action(s) I panel**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule-set</td>
<td>RULMAGIE</td>
</tr>
<tr>
<td>Rule-id</td>
<td>NEWRULE7</td>
</tr>
<tr>
<td>Function</td>
<td>modify</td>
</tr>
<tr>
<td>Key</td>
<td></td>
</tr>
<tr>
<td>Text</td>
<td></td>
</tr>
<tr>
<td>Queues</td>
<td>Alert Queue Name(s)</td>
</tr>
<tr>
<td>Priority</td>
<td>(CRIT,MAY,MN,WARN,INFO,CLEAR)</td>
</tr>
<tr>
<td>Color</td>
<td>RED,PINK,YEL,DKBL,LTBL,GRY,WHI</td>
</tr>
<tr>
<td>System</td>
<td>Return to target after PCMD</td>
</tr>
<tr>
<td>* Exec</td>
<td>Follow-up EXEC</td>
</tr>
<tr>
<td>Help</td>
<td>Associated HELP Panel</td>
</tr>
<tr>
<td>Targets</td>
<td>Target System</td>
</tr>
<tr>
<td>Udata</td>
<td>User Data</td>
</tr>
<tr>
<td>Origin</td>
<td>Origin</td>
</tr>
<tr>
<td>User</td>
<td>Userid</td>
</tr>
<tr>
<td>Alarm</td>
<td>Sound Alarm (YES/NO)</td>
</tr>
<tr>
<td>Publish</td>
<td>Messages (ADD/REPLACE/NO)</td>
</tr>
</tbody>
</table>

* Enter a question mark(?) and blank for more options.  
Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes

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 that 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 as shown in the following figure.

**Figure 74: Modifying an ALERT to survive MVS IPLs: Alert Action(s) II panel**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retain</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Escalate Direction</td>
<td>Up/Down</td>
</tr>
<tr>
<td>Interval</td>
<td></td>
</tr>
<tr>
<td>Disposition</td>
<td>Keep/Delete</td>
</tr>
</tbody>
</table>

* Enter a question mark(?) and blank in any field on the line for more options.  
Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
To specify that the ALERT is saved to survive BBI-SS PAS restarts and IPLs, enter **YES** in the **Retain** field.

Press **Enter** until the Rule Processor Detail Control panel is displayed.

Press **END** to apply changes; The Rule Set Overview panel is displayed.

Repeat Step 15 on page 114 and Step 16 on page 114 to complete saving the Rule.

---

**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**

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

2. Press **Enter**; the Selection Criteria - MSG panel is displayed as shown in the following figure.

   ![Figure 75: Escalating an ALERT: Selection Criteria panel](image)

   **Figure 75: Escalating an ALERT: Selection Criteria panel**

   BMC Software --------- Selection Criteria - MSG --------- MAINVIEW AutoOPERATOR
   COMMAND ===> TGT --- AOAD
   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 Name ==> Destination Console Name
   MLWTO Minor ===> Single/All
   Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes

3. Specify the event criteria that the Rule must match to cause the Rule to fire by entering **IEF863I** in the **Text-ID** field.
4 Press Enter until the Action Specification I - MSG panel is displayed as shown in the following figure.

**Figure 76: Escalating an ALERT: Action Specification I panel**

5 Specify the actions that the Rule will take when it is fired by specifying NO for Display at dest., and the IEF863I message will be suppressed from the console.

6 Press Enter until the Alert Action(s) I - MSG panel is displayed as shown in the following figure.

**Figure 77: Escalating an ALERT: Alert Action(s) I panel**

To issue an ALERT from a Rule, do the following steps:

- 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 IMPDOMID variables. These variables will resolve to the day of the week, the job name, and the number of times the Rule was fired. This combination ensures that 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 defaults to green, which is the default color associated with INFORMATIONAL-level ALERTs.

Press Enter; the Alert Actions II - MSG panel is displayed as shown in the following figure.

8 Press Enter; the Alert Actions II - MSG panel is displayed as shown in the following figure.

Figure 78: Escalating an ALERT: Alert Action(s) II panel

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 five minutes); at the end of five minutes the ALERT will become a WARNING level ALERT and remain there for five minutes. The ALERT will then progress through MINOR, MAJOR, and finally, CRITICAL priorities.

When the last time interval expires, you can specify in the Disposition field that the ALERT is either kept in the queue or deleted. This example shows the ALERT is kept.
10 Press **Enter** until the Rule Processor Detail Control panel is displayed.

11 Press **END** to apply changes; the Rule Set Overview panel is displayed.

12 Repeat Step 15 on page 114 and Step 16 on page 114 to complete saving the Rule NEWRULE8 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 when a variable value that you specify is 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.

2. Press **Enter**; the Selection Criteria - VAR panel is displayed as shown in the following figure.

3. Specify the variable name and when the Rule should fire (for example, when the variable is updated, deleted, or created).

   **Figure 79: Creating a VAR-initiated Rule: Selection Criteria panel**

   ![Selection Criteria panel](image)

   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

4. Enter the variable name in the **Name** field (in this example, the variable name is MYVAR).

5. 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.

6 Press **Enter**; the Variable Dependencies - VAR panel is displayed as shown in the following figure.

**Figure 80: Creating a VAR-initiated Rule: Variable Dependencies panel**

7 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.

8 Press **Enter** twice; the Action Specification(s) II - VAR panel is displayed as shown in the following figure.

**Figure 81: Creating a VAR-initiated Rule: Action Specification(s) I panel**
9 Specify the action that the Rule takes when it fire. In this example, the only action taken is TSO user USERID01 will be notified when the Rules fires.

10 Press Enter until the Rule Processor Detail Control panel is redisplayed.

11 Press END to apply changes; the Rule Set Overview panel is displayed.

12 Repeat Step 15 on page 114 and Step 16 on page 114 to complete saving the Rule NEWRULE9 to Rule Set RULMAGIE.

Creating a Rule that sends event and ALERT data to BMC Impact Manager

The following example shows how to code a Rule that captures the IEF450I WTO when a batch job abends.

The WTO is

```
11.11.41 JOB08218  IEF450I JDB1ABND - ABEND=S0C1 U0000 REASON=00000001  509
                      TIME=11.11.41
```

One of the actions that the Rule takes is to create an ALERT and also to send the event data and ALERT data to BMC IM.

For information about how to implement MainView AutoOPERATOR communications with BMC IM, refer to the MainView AutoOPERATOR Customization Guide.

All of the event data that is normally found in local Rule variables (such as IMFTEXT and so on) are sent in slots. Refer to the MainView AutoOPERATOR BMC Impact Integration for z/OS User Guide for more information about slots.

To create this Rule

1 Enter the new Rule ID and the event type on the Rule Processor Detail Control panel as shown in the following figure.

Figure 82: Creating a Rule that sends data to BMC IM: Rule Processor Detail Control panel

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Rule Processor Detail Control</th>
<th>AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND ====&gt;</td>
<td>TGT --- AOAO&lt;br&gt;TIME --- HH:MM:SS&lt;br&gt;DATE --- YY/MM/DD</td>
<td>Be selected by entering the two-character code</td>
</tr>
<tr>
<td>S1 - Selection Criteria&lt;br&gt;SV - Variable Dependencies&lt;br&gt;A1 - Action Specification(s) I&lt;br&gt;A2 - Action Specification(s) II</td>
<td>AV - Set Variable(s)&lt;br&gt;AA - Alert Action(s) I&lt;br&gt;AD - Alert Action(s) II&lt;br&gt;BA - Impact Integration for z/OS Action</td>
<td></td>
</tr>
</tbody>
</table>
Enter the Rule ID and the Event Type; in the preceding figure the Rule ID is IEF450I and the Event Type is MSG.

2 Press Enter; the Selection Criteria panel for MSG event type is displayed as shown in the following figure.

**Figure 83: Creating a Rule that sends data to BMC IM: Selection Criteria- MSG panel**

3 Enter the **Text ID** of the message, which is the message ID of the WTO, and the Job Name.

   The Text ID is IEF450I and the Job Name is JDB1ABND.

4 Press Enter until the Alert Action(s) I - MSG panel is displayed as shown in the following figure.

**Figure 84: Creating a Rule that sends data to BMC IM: Alert Action(s) I - MSG panel**
5 Begin on this panel to enter the ALERT specifications that the Rule sends to BMC IM. In the preceding figure, the Function, Key, Text, Queue and Priority fields are filled in.

Note
Some of the fields use LOCAL variables.

The ALERT that is created based on the specifications in the preceding figure is queued to a queue named Abendjob and have a priority of MAJOR. The text of the ALERT will state

Job JDB1ABND has abended.

6 Press Enter until the Event Action BEIM - MSG panel is displayed as shown in the following figure.

Fields are not populated automatically on this panel and the default action is that no data is sent to BMC IM.

Figure 85: Creating a Rule that sends data to BMC IM: BMC EM/SIM Event Action - MSG panel

<table>
<thead>
<tr>
<th>BMC Software COMMAND</th>
<th>Event Action - BEIM - MSG</th>
<th>AutoOPERATOR TGT</th>
<th>Command AOAO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule-set RULMAGIE</td>
<td>Rule-id IEF450I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Send Event</td>
<td>(Y</td>
<td>N)</td>
<td></td>
</tr>
<tr>
<td>Journal Details</td>
<td>(Y</td>
<td>N)</td>
<td></td>
</tr>
<tr>
<td>Target(s)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event Class</td>
<td>(UNKNOWN, OK, INFO, WARNING, MINOR, MAJOR, CRITICAL) (Default=WARNING)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Severity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component Alias</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event Slot Slot Value</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>____________________</td>
<td>_________________________</td>
<td></td>
<td></td>
</tr>
<tr>
<td>____________________</td>
<td>_________________________</td>
<td></td>
<td></td>
</tr>
<tr>
<td>____________________</td>
<td>_________________________</td>
<td></td>
<td></td>
</tr>
<tr>
<td>____________________</td>
<td>_________________________</td>
<td></td>
<td></td>
</tr>
<tr>
<td>____________________</td>
<td>_________________________</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Press ENTER or END to continue, or CANCEL to cancel changes.
In this example, fill in the following fields as specified in this table:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Enter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send Event</td>
<td>Y</td>
<td>specifies that the slots containing all the event data should be sent to the BMC IM cell</td>
</tr>
<tr>
<td>Event Class</td>
<td>Abend_Data</td>
<td>specifies that these slots will contain data related to an abend Note that this data is mixed case and is not translated into uppercase characters by MainView AutoOPERATOR.</td>
</tr>
<tr>
<td>Severity</td>
<td>MAJOR</td>
<td>specifies that this event is assigned a severity of MAJOR</td>
</tr>
<tr>
<td>Event Slot</td>
<td>Abend_Code</td>
<td>identifies the abend code</td>
</tr>
<tr>
<td></td>
<td>Abend_Reason</td>
<td>identifies the abend reason code</td>
</tr>
<tr>
<td></td>
<td>Abend_Time</td>
<td>identifies the time of the abend</td>
</tr>
</tbody>
</table>

To add new Event Slots or Event Classes, you must have defined a new class on the cell. In addition, you must make this new class a subclass of the default class. In this example, the class Abend_Code is a subclass of MV_EVENT_MSG.

Refer to the MainView AutoOPERATOR BMC Impact Integration for z/OS User Guide for more information about Event Slots and Event Classes.

The following figure shows an example of the panel with the fields filled in as described in this example.

**Figure 86: Creating a Rule that sends data to BMC IM: BMC EM/SIM Event Action - MSG panel**
In this example, the data for the slots is obtained from the LOCAL variables \texttt{WORD4}, \texttt{WORD6} and \texttt{WORD7}. Variable substringing is also used to resolve the values up from certain word positions within the WTO. Refer to the “Variable modifiers” on page 166 for more information on substringing.

7 Press \texttt{Enter} until the Rule Set Processor Detail Control panel is re-displayed.

8 Press \texttt{PF3} or \texttt{END}; the Rule Set Overview panel is re-displayed as shown in the following figure.

\textbf{Figure 87: Creating a Rule that sends data to BMC IM: Rule Set Overview panel}

<table>
<thead>
<tr>
<th>Rule Set ID: RULMAGIE</th>
<th>Ruleset Strategy ===&gt; ALL</th>
<th>TGT --- AOAO</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textbf{Command}</td>
<td></td>
<td>\textbf{TGT}</td>
</tr>
<tr>
<td>\textbf{Rule Set ID}</td>
<td></td>
<td>\textbf{TGT}</td>
</tr>
</tbody>
</table>

The new Rule is added at the bottom of this display.

9 Press \texttt{PF/11} five times to scroll this display to the right.

The following figure shows the BMC Impact Integration \textit{for z/OS} action data that was created with this Rule.

\textbf{Figure 88: Creating a Rule that sends data to BMC IM: Rule Set Overview panel with BiiZ Action data}

\textbf{BMC Software} \texttt{---------} \textbf{Rule Set Overview} \texttt{---------} \textbf{AutoOPERATOR}
\textbf{COMMAND} \texttt{-----} \textbf{TGT} \texttt{--- AOAO}
\textbf{Rule Set ID: RULMAGIE} \textbf{Ruleset Strategy} \texttt{----> ALL} \textbf{TGT} \texttt{--- AOAO}

The new Rule is added at the bottom of this display.

---

Creating a Rule that sends event and ALERT data to BMC Impact Manager

\textbf{BMC Software} \texttt{---------} \textbf{Rule Set Overview} \texttt{---------} \textbf{AutoOPERATOR}
\textbf{COMMAND} \texttt{-----} \textbf{TGT} \texttt{--- AOAO}
\textbf{Rule Set ID: RULMAGIE} \textbf{Ruleset Strategy} \texttt{----> ALL} \textbf{TGT} \texttt{--- AOAO}

The new Rule is added at the bottom of this display.
Creating a Rule for MLWTO messages

The Selection Criteria panel for MSG events allows you to create Rules that fire for MLWTO messages and based on the specification in the field MLTW0 minor, the Rule processes MLWTO messages either as a single entity (specify MLWT0 minor ==> ALL) or where each minor line of the message is treated as a separate entity (specify MLWTO minor ==> SINGLE).

The following 2 examples show how to use variables and the Variable Dependencies panel in the Rule to refine the selection criteria by using the with variable dependencies and variables process the lines of a MLWTO minor message.

Example 1: creating a Rule where MLWTO minor = ALL

In this example, a MLWTO message is processed by the Rule as a complete entity because MLWT0 minor = ALL is specified on the Selection Criteria panel.

The Rule in this example uses the Variable Dependencies panel and the variable LINEx_WORDn to help the Rule fire based on content within the MLWTO such as the message ID of the major message, the job name, and some additional information in the first and fifth minor lines of the message.

When all of search criteria and variable dependency criteria is met, this example Rule creates an ALERT with the appropriate message.

The MLWTO message is

```
M 05109 02:28:39.43 STC07239 00000281  IEA995I SYMPTOM DUMP OUTPUT 672
D 672 00000281 SYSTEM COMPLETION CODE=0C4
D 672 00000281 TIME=02.28.39  SEQ=00112  CPU=0000  ASID=005A
D 672 00000281 PSW AT TIME OF ERROR  87801000  80007954  ILC 2  INTC 01
D 672 00000281 ACTIVE LOAD MODULE  ADDRESS=00007708  OFFSET=0000024C
D 672 00000281 NAME=ACCTPAY
D 672 00000281 DATA AT PSW  0000794E - 00011311  0A019140  B7F14770
D 672 00000281 GR 0: 00000000_00000001  1: 00000000_FFF6E70
D 672 00000281 2: 00000000_0000943C  3: 00000000_00009999
D 672 00000281 4: 00000000_000099CA  5: 00000000_000099F2
D 672 00000281 6: 00000000_00000004  7: 00000000_0000917C
D 672 00000281 8: 00000000_0009144  9: 00000000_0009210
D 672 00000281 A: 00000000_0009706  B: 00000000_0008707
D 672 00000281 C: 00000000_80007708  D: 00000000_00000000
D 672 00000281 E: 00000000_80007BFE  F: 00000004_807BFE278
E 672 00000281 END OF SYMPTOM DUMP
```
To create a Rule where MLWTO minor = ALL

1 Fill in the Selection Criteria - MSG panel as follows:

<table>
<thead>
<tr>
<th>Issuer Identification:</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Job name</td>
<td>PROD</td>
<td>Name of issuer</td>
</tr>
<tr>
<td>Type</td>
<td>(JOB, STC, or TSO)</td>
<td></td>
</tr>
<tr>
<td>Jobclass</td>
<td></td>
<td>Job class of issuer</td>
</tr>
<tr>
<td>Acct Info</td>
<td></td>
<td>Job accounting information</td>
</tr>
<tr>
<td>RACF User</td>
<td>RACF Userid</td>
<td></td>
</tr>
<tr>
<td>RACF Group</td>
<td>RACF Group name</td>
<td></td>
</tr>
<tr>
<td>Route Codes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Desc Codes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Console Name</td>
<td>Destination Console Name</td>
<td></td>
</tr>
<tr>
<td>MLWTO Minor</td>
<td>ALL</td>
<td>Single/All</td>
</tr>
</tbody>
</table>

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

When MLWTO Minor ==> ALL is specified, all the lines of multiline WTO message are processed by the Rule as a single entity. BMC Software recommends that you use the variables &LINEx_WORDn variables on the Variable Dependencies panel so that the Rule can fire based on specific content of the MLWTO.

Note

When creating a Rule for a MLWTO message, the Rule compares the selection criteria values to the content of only the first line (Major line) of the MLWTO.

With the &LINEx_WORDn, x is set for every line of the MLWTO and n is set for every word in a line.

2 Press Enter the Variable Dependencies - MSG panel is displayed as shown in the following figure.

Figure 89: Variable Dependencies for MLWTO Minor=ALL

<table>
<thead>
<tr>
<th>Variable-name Op Variable-Value AND</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>LINE2_WORD3:6:3_________________ EQ 0C4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LINE5_WORD1_____________________ EQ NAME=ACCTPAY</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

--> ATTENTION: Use ASV command to add ASV criteria to the rule <---
Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes
3 Press Enter until the Alert Action(s) I - MSG panel is displayed as shown in the following figure.

The following entry on the Alert Action(s) II - MSG panel specify that when the Rule fires, a MainView AutoOPERATOR ALERT will be created.

**Figure 90: Alert Action(s) I - MSG for MLWTO Minor=ALL**

```
BMC Software -------------- Alert Action(s) I - MSG ------------- AutoOPERATOR
COMMAND ===>                                                       TGT --- A0AO
Rule-set === RUL01       Rule-id === DA010
Function ===> ADD (ADD, DELETE, DELETEQ)
Key ===> PROD&IMFOJOB
Text ===> PRODUCTION JOB &IMFOJOB ABENDED WITH AN &LINE1_WORD3:6:3 IN
PROGRAM &LINE5_WORD1:6:8
Alert Queue Name(s) === PRODQUE
Alert Queue Name(s) === PRODQUE
Priority ===> (CRIT,MAJ,MIN,WARN,INFO,CLEAR)
Color ===> RED,PINK,YEL,DKBL,LTBL,GRE,WHI
PCMD ===> System
Exec ===> Follow-up EXEC
Help ===> Associated HELP Panel
Targets ===> Target System
Udata ===> User Data
Origin ===> Origin
User ===> Userid
Alarm ===> Sound Alarm (YES/NO)
Publish ===> Alert Publishing Mode
* Enter a question mark(?) and blank for more options.
Press ENTER to continue, END return to Detail Control, CANCEL to cancel change
```

4 Press Enter and finish completing the Rule.

**Example 2: creating a Rule where MLWTO minor = SINGLE**

In this example, a MLWTO message is processed by the Rule where each minor line of the MLWTO can cause the Rule to fire because **MLWTO minor ==> SINGLE** is specified on the Selection Criteria panel.

This example Rule fires based on the message ID of the major and additional information from the minor messages. The following MLWTO message is the same as the message used for “Example 1: creating a Rule where MLWTO minor = ALL” on page 143.

```
M 05109 02:28:39.43 STC07239 00000281  IEA995I SYMPTOM DUMP OUTPUT 672
D 672 00000281 SYSTEM COMPLETION CODE=0C4
D 672 00000281 TIME=02.28.39  SEQ=00112  CPU=0000  ASID=005A
D 672 00000281 PSW AT TIME OF ERROR 07801000 00079544 ILC 2  INT 01
D 672 00000281 ACTIVE LOAD MODULE   ADDRESS=00007708  OFFSET=0000024C
D 672 00000281 NAME=ACCTPAY
D 672 00000281 DATA AT PSW  0000794E - 00011311 0A019140  B7F14770
D 672 00000281 GR 0: 00000000_00000001 1: 00000000_FFFF6F70
D 672 00000281 2: 00000000_00009998 3: 00000000_00009998
D 672 00000281 4: 00000000_00009998 5: 00000000_00009998
D 672 00000281 6: 00000000_00009998 7: 00000000_00009998
D 672 00000281 8: 00000000_00009998 9: 00000000_00009998
D 672 00000281 A: 00000000_00009998 B: 00000000_00009998
D 672 00000281 C: 00000000_00009998 D: 00000000_00009998
D 672 00000281 E: 00000000_00009998 F: 00000000_00009998
D 672 00000281 END OF SYMPTOM DUMP
```
To create this Rule

1 Fill in the Selection Criteria - MSG panel as follows:

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Selection Criteria - MSG</th>
<th>MAINVIEW AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND</td>
<td>TGT</td>
<td>AOAO</td>
</tr>
</tbody>
</table>

| Rule-set     | RUL01                     | Rule-id                | SLIPX33E               |
|--------------|---------------------------|------------------------|
| Text Description: | Ignore leading plus:     |
| Text ID      | IEA9951                   |                        |
| Text String (Enter Below): |                        |

Issuer Identification:

<table>
<thead>
<tr>
<th>Job name</th>
<th>PROD</th>
<th>Name of issuer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>(JOB, STC, or TSO)</td>
<td></td>
</tr>
<tr>
<td>Jobclass</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Acct Info</td>
<td></td>
<td>Job accounting information</td>
</tr>
<tr>
<td>RACF User</td>
<td></td>
<td>RACF Userid</td>
</tr>
<tr>
<td>RACF Group</td>
<td></td>
<td>RACF Group name</td>
</tr>
<tr>
<td>Route Codes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Desc Codes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Console Name</td>
<td></td>
<td>Destination Console Name</td>
</tr>
<tr>
<td>MLWTO Minor</td>
<td>SINGLE</td>
<td>Single/All</td>
</tr>
</tbody>
</table>

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

When MLWTO Minor==>SINGLE is specified, each individual line of multiline WTO message are processed by the Rule as separate events. BMC Software recommends that you use the variables &LINE2_WORDn variables on the Variable Dependencies panel so that the Rule can fire based on specific content of the MLWTO.

---

Note

When creating a Rule for a MLWTO message, the Rule compares the selection criteria values to the content of only the first line (Major line) of the MLWTO.

With the &LINE2_WORDn variable, LINE2 resolves to each and every line of the MLWTO and a WORDn variable resolves for each word of each line of the MLWTO.

2 Press Enter; the Variable Dependencies - MSG panel is displayed as shown in the following figure.

Figure 91: Variable Dependencies for MLWTO Minor=SINGLE

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Variable Dependencies - MSG</th>
<th>AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND</td>
<td>TGT</td>
<td>AOAO</td>
</tr>
</tbody>
</table>

| Rule-set     | RUL01                       | Rule-id      | ABNDPROD |
|--------------|-----------------------------|--------------|
| Variable-name | Op       | Variable-Value | AND       |
| LINE2_WORD3:6:3 | EQ | 0C4               |            |
|              |               |                |            |
|              |               |                |            |
|              |               |                |            |
|              |               |                |            |
|              |               |                |            |
|              |               |                |            |

---

Creating a Rule for MLWTO messages

MainView AutoOPERATOR Basic Automation Guide Volume 1: Using Rules
In this example, when you specify the variable LINE2_WORD3:6:3, the Rule Processor checks the third word in each and every line of the MLWTO message and checks the value of the last 3 characters starting at position 6.

When these characters resolve to 0C4 on any line of the MLWTO, the Rule fires. In this case, the Rule fires because on line 2 of the MLWTO, in the third word, the last 3 characters starting at position 6 contain 0C4.

You should contrast this example with the one in “Example 1: creating a Rule where MLWTO minor = ALL” on page 143 because the same MLWTO is used in both examples but based on the MLWTO Minor setting and the variables used, the Rule processes the major and minor lines of the MLWTO differently.

Creating a Rule for IMS MLSEG messages

The Selection Criteria panel for IMS events allows you to create Rules that fire for MLSEG messages.

Based on the specification in the field MLSEG Minor, an IMS Rule processes IMS MLSEG messages either as a single entity (specify MLSEG Minor ===> ALL) or each minor segment of the message is treated as a separate entity (specify MLSEG Minor ===> SINGLE).

The following two examples show how to use variables on the Variable Dependencies panel in a Rule to refine the selection criteria and process the segments of an IMS MLSEG message.

Example 1: creating a Rule with MLSEG Minor = ALL

In this example, an IMS MLSEG message is processed by the Rule as a complete entity because MLWTO Minor ===> ALL is specified on the Selection Criteria panel.
The Rule in this example uses the Variable Dependencies panel and the variables &LINEx_WORDn. The Rule fires based on content within the MLSEG such as the message ID of the major segment, the job name, and some additional information in the third minor segment of the message.

When all of the selection and variable dependency criteria is met, this example Rule creates an ALERT with the appropriate text.

The IMS MLSEG message is DFS2503W:

```
M 09048 13:49:03.44 STC06054 00000090  DFS2503W DYNAMIC ALLOCATION FAILED FOR 97
D                        978 00000090  (DFS2503W) DATASET NAME AAO.IMS111.DE2PCUST
E                        978 00000090  (DFS2503W) DATABASE BE2PCUST REASON CODE 1708
```

1 Fill in the Selection Criteria - IMS panel as follows:

```
BMC Software  -------------- Selection Criteria - IMS  -------------- AutoOPERATOR
COMMAND ===>                                                 TGT  --- NS7A
Rule-set === AAOUL01           Rule-id  === DFS2503W
Text Description: First word of message
Text ID     ===> DFS2503W
Text String (Enter Below):

Issuer Identification:
Job name    ===> IMS11PRD                          Jobname of IMS
Type        ===> (JOB, STC, or TSO)               Job class of issuer
Acct Info   ===>                                   Job accounting information
RACF User   ===>                                   RACF Userid
RACF Group  ===>                                   RACF Group name
MLSEG Minor ===> ALL                              All/Single

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

When MLSEG Minor == ALL is specified all the lines of IMS multisegment message are processed by the Rule as a single entity. BMC Software recommends that you use the variables &LINEx_WORDn on the Variable Dependencies panel so that the Rule can fire based on specific content of the MLSEG.

With the &LINEx_WORDn variables, x is set for every segment of the MLSEG and n is set for every word in a segment.

2 Press Enter; the Variable Dependencies - IMS panel is displayed as shown in the following figure.

**Figure 92: Variable Dependencies for MLSEG Minor=ALL**

```
BMC Software  -------------- Variable Dependencies - IMS  -------------- AutoOPERATOR
COMMAND ===>                                                 TGT  --- NS7A
```

---

**Note**

When creating a Rule for an IMS MLSEG message, the Rule compares the selection criteria values to the content of only the first (major) segment of the MLSEG.
When you specify variable &LINE3_WORD3:1:4, the Rule Processor checks the value of the first four characters of the third word, in segment three of the IMS MLSEG message. If this value resolves to characters BE2P, and if the sixth word of the same segment (&LINE3_WORD6) resolves to 1708, and the value of the fourth word of the second segment starts with 'AAO', the Rule fires.

3 Press Enter until the Alert Action(s) I - IMS panel is displayed as shown in the following figure.

The following entry on the Alert Action(s) II - IMS panel specifies that when the Rule fires, a MainView AutoOPERATOR ALERT will be created.

**Figure 93: Alert Action(s) I - IMS for MLSEG Minor=ALL**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Alert Action(s) I - IMS</th>
<th>AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rule-set === AAORUL01</td>
<td>Rule-id  === DFS2503W</td>
<td></td>
</tr>
<tr>
<td>Function ===&gt; ADD</td>
<td>(ADD, DELETE, DELETEQ)</td>
<td></td>
</tr>
<tr>
<td>Key     ===</td>
<td>PROD&amp;IMFOJOB</td>
<td></td>
</tr>
<tr>
<td>Text     ===&gt; PRODUCTION IMS &amp;IMFOJOB - ALLOCATION FAILED FOR DATASET: &amp;LINE2_WORD4, DATABASE: &amp;LINE3_WORD3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Queues ===&gt; PRODQUE</td>
<td>Alert Queue Name(s)</td>
<td></td>
</tr>
<tr>
<td>Priority ===&gt; WARN</td>
<td>(CRIT, MAJ, MIN, WARN, INFO, CLEAR)</td>
<td></td>
</tr>
<tr>
<td>Color   ===&gt; RED, PINK, YEL, DKBL, LTBL, GRE, WHI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCMD     ===&gt;</td>
<td>Return to target after PCMD</td>
<td></td>
</tr>
<tr>
<td>System   ===&gt;</td>
<td>Alert Publishing Mode</td>
<td></td>
</tr>
<tr>
<td>* Exec   ===&gt;</td>
<td>Follow-up EXEC</td>
<td></td>
</tr>
<tr>
<td>Help     ===&gt;</td>
<td>Associated HELP Panel</td>
<td></td>
</tr>
<tr>
<td>Targets ===&gt;</td>
<td>Target System</td>
<td></td>
</tr>
<tr>
<td>Udata    ===&gt;</td>
<td>User Data</td>
<td></td>
</tr>
<tr>
<td>User     ===&gt;</td>
<td>Origin</td>
<td></td>
</tr>
<tr>
<td>Alarm    ===&gt;</td>
<td>Sound Alarm (YES/NO)</td>
<td></td>
</tr>
<tr>
<td>Publish  ===&gt;</td>
<td>Alert Publishing Mode</td>
<td></td>
</tr>
</tbody>
</table>

* Enter a question mark(?) and blank for more options. Press ENTER to continue. END return to Detail Control, CANCEL to cancel change.

4 Press Enter and finish completing the Rule.
Example 2: creating a Rule where MLSEG Minor = SINGLE

In this example, an IMS MLSEG message is processed by the Rule where each MLSEG segment can cause the Rule to fire because **MLSEG Minor ===> SINGLE** is specified on the Selection Criteria panel.

This example Rule fires based on the message ID of the major segment and additional information from the minor messages. The following IMS MLSEG message is the same as the message used for “Example 1: creating a Rule with MLSEG Minor = ALL” on page 147.

```
M 09048 13:49:03.44 STC06054 00000000 DFS2503W DYNAMIC ALLOCATION FAILED FOR 97
E 978 00000000 (DFS2503W) DATASET NAME AAO.IMS111.DE2PCUST
E 978 00000000 (DFS2503W) DATABASE BE2PCUST REASON CODE 1708
```

To create a Rule where MLSEG Minor = SINGLE

1 Fill in the Selection Criteria - IMS panel as follows:

```
BMC Software ------------ Selection Criteria - IMS ------------ AutoOPERATOR
COMMAND ===>                                                 TGT --- NS7A

Rule-set === AAORUL01          Rule-id === DFS2503W

Text Description:                                                      First word of message
Text ID =====> DFS2503W

Issuer Identification:
Job name ===> IMS11PRD                     Jobname of IMS
Type ===> (JOB, STC, or TSO)            Job class of issuer
Acct Info ===>                                   Job accounting information
RACF User ===>                                   RACF Userid
RACF Group ===>                                   RACF Group name
MLSEG Minor ===> SINGLE             All/Single
```

When **MLSEG Minor ===> SINGLE** is specified, each individual line of an IMS multisegment message is processed by the Rule as separate events. BMC Software recommends that you use the variables &LINE2_WORDn variables on the Variable Dependencies panel so that the Rule can fire based on specific content of the MLSEG.

**Note**

When creating a Rule for a IMS MLSEG message, the Rule compares the selection criteria values to the content of only the first line (major segment) of the MLSEG.

With the &LINE2_WORDn variable, LINE2 resolves to each and every segment of the MLSEG and a WORDn variable resolves for each word of each line of the MLSEG.
Creating a Rule for CICS multiline messages

The Selection Criteria panel for CICS events allows you to create Rules that fire for multiline transient data queue (MLTDQ) messages.

Based on the specification in the field **MLTDQ**, a CICS Rule can process a CICS MLTDQ as a single entity (specify **MLTDQ ===> ALL**), or it can process only the first line of the message.

The following example shows how to use variables on the Variable Dependencies panel in a Rule to refine the selection criteria and process the lines of a CICS MLTDQ message, DFHXS1111. Line 1 of the message is:

```
DFHXS1111 08/30/2011 23:21:26 CI10 CEMT Security violation by user CI10 for resource CI10.TASK in class CCICSCMD. SAF
```

Line 2 of the message is:

```
codes are (X'00000008',X'00000000'). ESM codes are (X'00000008',X'00000000').
```

1. **Creating a rule**

   **Complete the Selection Criteria - CICS panel as follows:**

   ```
   BMC Software  ----------------- Selection Criteria - CICS ---------------- AutoOPERATOR
   COMMAND ===>                                                 TGT  --- JBAD
   Rule-set === AAORUL00             Rule-id  === DFHXS1111
   Text Description:
   Text ID     ===> DFHXS1111                         First word of TDQ msg/cmd
   ```
Because this panel specifies \textbf{MLTDQ \implies ALL}, the Rule will process all lines of the CICS multiline message as a single entity.

\textbf{Note}

For a CICS multiline message, the Rule compares the selection criteria values to the content of only the first (major) line of the MLTDQ.

2 Press \textbf{Enter} to continue to the Variable Dependencies - CICS panel as shown in the following figure.

A Rule can fire based on selected content within the MLTDQ (such as the message ID of the major line, user ID, resource name, SAF codes, and so on). In this example, \texttt{LINE1\_WORD10} tells the Rule Processor to check the value of the tenth word in line 1 of the message; the processor determines whether that word resolves to CI10 (as specified under \textbf{Variable-Value}). Also, \texttt{LINE2\_WORD7:4:8} tells the processor to check the seventh word in line 2, starting in position 4 with a length of 8, to see if that value resolves to 00000008. If both variables (based on the panel's AND entry) resolve, the Rule will fire.

\texttt{Figure 95: Variable Dependencies for MLTDQ \implies ALL}

<table>
<thead>
<tr>
<th>Variable-name</th>
<th>Op</th>
<th>Variable-Value</th>
<th>OR/AND</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINE1_WORD10</td>
<td>EQ</td>
<td>CI10</td>
<td>AND</td>
</tr>
<tr>
<td>LINE2_WORD7:4:8</td>
<td>EQ</td>
<td>00000008</td>
<td></td>
</tr>
</tbody>
</table>

--> \textbf{ATTENTION: Use ASV command to add ASV criteria to the rule} <--
3 Press **Enter** until the Alert Action(s) I - CICS panel is displayed as shown in the following figure.

The entries in the following figure specify that when the Rule fires, a MainView AutoOPERATOR ALERT is created, containing the information from the CICS MLTDQ message DFHXS111.

**Figure 96: Alert Action(s) I - CICS for MLTDQ ===> ALL**

- **BMC Software**
- **Command**
- **Rule-set**
- **Rule-id**
- **Function**
- **Key**
- **Text**
- **Queues**
- **Priority**
- **Color**
- **PCMD**
- **System**
- **Exec**
- **Help**
- **Targets**
- **Udata**
- **Origin**
- **User**
- **Alarm**
- **Publish**

*Enter a question mark(?) and blank for more options.*

4 Press **Enter** to continue through the remaining panels and save the Rule.
Creating more powerful Rules

This chapter provides topics about creating more powerful Rules.

Over of creating more powerful Rules

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 applicability to only very specific situations. However, by using variables, combinations of literals and variables, or pattern matching, you can create more powerful and flexible Rules that apply to more general situations. By using these techniques, you can accomplish an automation task with fewer 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 155
- “Using variables in the Rule Processor” on page 160

Using pattern matching on Selection Criteria panels

On the selection criteria panels for all events, you can use pattern matching.
You can use wildcard characters on most of the fields in the various Selection Criteria panels. The wildcard characters are

- the plus sign (+) stands for any 1 character
- the asterisk (*) stands for any, all, or no characters
- the not sign (¬) stands for a negative logic test
  For example, ¬ABCD matches with everything except ABCD.
- the single quotation mark (‘) acts as a toggle switch
  The first single quotation mark turns pattern matching off, the next turns it on again. If a text string contains a single quotation mark, it must be preceded by a single quotation mark. For example: This text is 'quoted' matches This text is ''quoted''.

If a + or * is embedded within the pattern string and the pattern string is specified within quotation marks, 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:

```
Text Description:
Text-ID      ==> DFH3461I                   First word of event
```
Example 2

In this example, any message whose first word:

- is 7 characters long and starts with INIT
- has three positions containing characters (not blanks)
- has the word INACTIVE followed anywhere in the character string by the word MISSION

is selected.

### Text Description:
- **Text ID**: INIT+++  
- **Text String**: (Enter Below): *INACTIVE*MISSION*

---

Example 3

In this example, any message whose first word

- starts with INIT
- has three positions containing characters (not blanks)
- has the word INACTIVE followed at the end of the character string by the word MISSION

is selected.

### Text Description:
- **Text ID**: INIT+++  
- **Text String**: (Enter Below): *INACTIVE*MISSION

---

Note

This text string definition matches 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, you must code the text string with an asterisk after the word MISSION as in “Example 2” on page 157.

---

Example 4

For better performance, use wildcard characters with care.
For example, using asterisks around the word STARTED causes Rule Processing to search every message for the word STARTED:

<table>
<thead>
<tr>
<th>Text Description:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text-ID ==&gt; *</td>
</tr>
<tr>
<td>Text String ==&gt; (Enter Below)</td>
</tr>
<tr>
<td><em>STARTED</em></td>
</tr>
</tbody>
</table>

It would be more efficient if you did the following search which searches only IM9161I messages for the word STARTED.

<table>
<thead>
<tr>
<th>Text Description:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text-ID ==&gt; IM9161I</td>
</tr>
<tr>
<td>Text String ==&gt; (Enter Below)</td>
</tr>
<tr>
<td><em>STARTED</em></td>
</tr>
</tbody>
</table>

### 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.

The following table shows examples of how you might try to fill in the pattern matching fields and which variations would create matching results.

<table>
<thead>
<tr>
<th>Character</th>
<th>String pattern</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3912,PROD,AAO)</td>
<td>*</td>
<td>Match</td>
</tr>
<tr>
<td>(3912,PROD,AAO)</td>
<td>*, PROD</td>
<td>Match</td>
</tr>
<tr>
<td>(3912,PROD,AAO)</td>
<td>*, PROD, *</td>
<td>Match</td>
</tr>
<tr>
<td>(3912,PROD,AAO)</td>
<td>3912,*,AAO</td>
<td>Match</td>
</tr>
<tr>
<td>(3912,PROD,AAO)</td>
<td>3*,P*,A*</td>
<td>Match</td>
</tr>
<tr>
<td>(3912,PROD,AAO)</td>
<td>PROD</td>
<td>No Match</td>
</tr>
</tbody>
</table>

The following table shows some additional examples of pattern matches.
### Table 12: Example of pattern matches

<table>
<thead>
<tr>
<th>Character string</th>
<th>Pattern</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABCDEF</td>
<td>ABC++F</td>
<td>Match</td>
</tr>
<tr>
<td>ABCDEF</td>
<td>ABC+F</td>
<td>No match</td>
</tr>
<tr>
<td>ABCDEF</td>
<td>A*</td>
<td>Match</td>
</tr>
<tr>
<td>ABCDEF</td>
<td>*D+F</td>
<td>Match</td>
</tr>
<tr>
<td>ABCDEF</td>
<td>A+C*</td>
<td>Match</td>
</tr>
<tr>
<td>ABCDEF</td>
<td>A*F</td>
<td>Match</td>
</tr>
<tr>
<td>ABCDEF</td>
<td>A*+F</td>
<td>Match</td>
</tr>
<tr>
<td>ABCDEF</td>
<td>A+FF</td>
<td>No match</td>
</tr>
<tr>
<td>ABCDEF</td>
<td>A*BCDEF</td>
<td>Match</td>
</tr>
<tr>
<td>ABCDEF</td>
<td>A+B*</td>
<td>No match</td>
</tr>
<tr>
<td>ABCDEF</td>
<td>'A*BCDEF'</td>
<td>No match</td>
</tr>
<tr>
<td>ABCDEF</td>
<td>¬XYZ</td>
<td>Match</td>
</tr>
<tr>
<td>ABCDEF</td>
<td>¬<em>BCD</em></td>
<td>No match</td>
</tr>
</tbody>
</table>

### Additional notes

The single quotation mark is used as a toggle switch for pattern matching and cannot be coded in the **Text String** field of the Selection Criteria panel or Variable Dependencies panel.

Pattern matching must be used in place of single quotation marks.

For example, on the Variable Dependencies panel, coding

`WORD3 EQ +IS MESSAGE`

results in a match with

`IJK5868 THIS 'IS A' TEST MESSAGE`

In the **Text String** field, coding

*IS A*

or

*TEST MESSAGE*
also results 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.

Two types of variables 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)
  
  — MainView AutoOPERATOR-supplied variables

See “List of MainView AutoOPERATOR-supplied EVENT variables” on page 169 for a list of EVENT variables.
Restrictions for long variables

Long variables cannot be used in Rules processing.

None of the Rule event types, including the VAR event type, support the use of long variables and the Rule Processor panels do not support long variables in any field.

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 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:

  1. BBI-SS PAS SHARED variables

  2. EVENT variables (WORD.xxx variables and MainView 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.

**WARNING**

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 MainView 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 symbols:

<table>
<thead>
<tr>
<th>End of variable name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blank</td>
<td></td>
</tr>
<tr>
<td>Period</td>
<td>.</td>
</tr>
<tr>
<td>Comma</td>
<td>,</td>
</tr>
<tr>
<td>Ampersand</td>
<td>&amp;</td>
</tr>
<tr>
<td>Apostrophe</td>
<td>'</td>
</tr>
<tr>
<td>Hyphen</td>
<td>-</td>
</tr>
<tr>
<td>Asterisk</td>
<td>*</td>
</tr>
<tr>
<td>Plus sign</td>
<td>+</td>
</tr>
<tr>
<td>Forward slash</td>
<td>/</td>
</tr>
<tr>
<td>Backslash</td>
<td>\</td>
</tr>
<tr>
<td>Colon</td>
<td>:</td>
</tr>
</tbody>
</table>

For example, the variable &MSGID contains the characters IEF1234I and the variable &TEST contains the characters ABCD and 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:

<table>
<thead>
<tr>
<th>End of variable name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blank</td>
<td></td>
</tr>
<tr>
<td>Comma</td>
<td>,</td>
</tr>
<tr>
<td>Ampersand</td>
<td>&amp;</td>
</tr>
<tr>
<td>Apostrophe</td>
<td>'</td>
</tr>
<tr>
<td>Hyphen</td>
<td>-</td>
</tr>
<tr>
<td>Asterisk</td>
<td>*</td>
</tr>
<tr>
<td>Plus sign</td>
<td>+</td>
</tr>
</tbody>
</table>
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

\[ &!SHIFT1.CICSTAT = SCHEDULED \]
\[ CICSTAT = \text{not defined} \]

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 Action: Set Variable action specification panel. On this panel, you do not need to use the special characters for either simple or compound variables because any value entered in this field is treated as a variable.
Setting variables

The action specification panel Action: Set Variable (see following figure) can be used for all Rule event types and on the panel, you can perform the following actions:

- Set one or more variables

- Increase or decrease a variable’s numerical value

To increase or decrease the numerical value, specify +n or -n, where n is a number.

- Use variable substringing

Figure 97: Action: Set Variables - MSG panel

To set simple variables, enter the variable name in the first field and the value in the second field.

Example

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:

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.

Substringing and Trimming 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
- “Variable trimming” on page 167

Variable substringing

To access only a part (a substring) of a variable, specify a starting position and an optional length of the substring.

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 cannot be longer than the length of the variable.

Optionally, another colon may follow a substring specification that can designate the total length of the character string to evaluate.

Defaults

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.

WARNING

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
```

<table>
<thead>
<tr>
<th>If you specify</th>
<th>The result is</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;ABC:11</td>
<td>TEST</td>
</tr>
<tr>
<td>&amp;ABC:1:4</td>
<td>THIS</td>
</tr>
<tr>
<td>&amp;ABC:11:38</td>
<td>TEST</td>
</tr>
<tr>
<td>&amp;ABC:6:NOT</td>
<td>IS A TEST:NOT</td>
</tr>
</tbody>
</table>

This example shows that you can add more characters that are appended to the value returned from the substringing operation.

---

**Variable 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. You 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 job name 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 155 for information about pattern matching.

---

**Using special-use variables on the Variable Dependencies panel**

MainView AutoOPERATOR provides special-use SHARED variables that you can use for variable comparisons on the Variable Dependencies panel for when a variable might have one of the following three conditions:

- Variable does not exist: The SHARED variable does not actually exist because it was never created or it was deleted.
- Variable is blank: A SHARED variable was created with a blank value.
- Variable is null: A SHARED variable was created with a null value such as a zero length.

The following describes the SHARED variables:

**Table 13: Special-use variables for comparisons on the Variable Dependencies panel**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QBLANK</td>
<td>A single blank character</td>
</tr>
<tr>
<td>QNOEXIST</td>
<td>A special variable to be used for comparing for non-existent variables</td>
</tr>
<tr>
<td>QNULL</td>
<td>Null variable, such as zero length data</td>
</tr>
<tr>
<td>QZ</td>
<td>A special variable that can be used to test for all three conditions</td>
</tr>
</tbody>
</table>

The following table shows whether a Rule matches based on the outcome of a variable value being compared to the special-use variable on the Variable Dependencies panel where YES indicates a Rule matches and NO indicates a Rule would not match:
Table 14: Matrix of Rule matching outcomes for special-use variables

<table>
<thead>
<tr>
<th>When a variable value is</th>
<th>Compared with &amp;QBLANK</th>
<th>Compared with &amp;QNOEXIST</th>
<th>Compared with &amp;QNULL</th>
<th>Compared with &amp;QZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>A single blank (X’40’)</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Multiple blanks</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Null (Zero length)</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Does not exist</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Valid value</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>

For example, the variable named TAKEACTION is entered on the Variable Dependencies panel to compare if the variable is set to NO, or to the &QNOEXIST variable, which represents the situation that the TAKEACTION variable does not exist.

---

**List of MainView 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

- MainView AutoOPERATOR-supplied variables
- WORDxxx variables, where xxx represents a number (for example, &WORD4)

Many of these variables are also be available for use by MainView AutoOPERATOR EXECs.
For more information about variables and MainView AutoOPERATOR EXECs, refer to the variables discussion in the *MainView AutoOPERATOR Advanced Automation Guide*.

When creating Rules for the ALRT event type, additional variables are available. Refer to Table 15 on page 170 for a list.

The following table shows all the EVENT variables supplied by MainView AutoOPERATOR.

### Table 15: MainView AutoOPERATOR-supplied EVENT variables

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;IMFACCTG</td>
<td>contains all accounting fields for a particular event. The accounting field values are separated by blanks. Maximum length is 142.</td>
</tr>
<tr>
<td>&amp;IMFALGID</td>
<td>contains a 16-character alphanumeric Alarm Group ID created by MainView Alarm Manager</td>
</tr>
<tr>
<td>&amp;IMFALID</td>
<td>contains the alarm ID associated with an alarm created by MainView Alarm Manager</td>
</tr>
<tr>
<td>&amp;IMFALPRI</td>
<td>contains the user-assigned priority of the alarm. Possible values are: 1 – Critical, 2 – Major, 3 – Minor, 4 – Warning, 5 – Informational, 6 – Clearing</td>
</tr>
<tr>
<td>&amp;IMFALQID</td>
<td>contains the name of the queue to which the alarm was assigned</td>
</tr>
<tr>
<td>&amp;IMFALRM</td>
<td>contains either Y (sound an alarm) or N (do not sound alarm)</td>
</tr>
<tr>
<td>&amp;IMFAUTH</td>
<td>contains Y if the message was issued from an authorized program</td>
</tr>
<tr>
<td>&amp;IMFCART</td>
<td>contains an 8-character Command and Response Token (CART) associated with an MVS command</td>
</tr>
<tr>
<td>&amp;IMFCARTX</td>
<td>contains the &amp;IMFCART variable in 16-character printable EBCDIC format</td>
</tr>
<tr>
<td>Variable name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>&amp;IMFCIJOB</td>
<td>contains the values for the &amp;IMFCIJOB variable for different event types are as follows:</td>
</tr>
<tr>
<td></td>
<td>■ ALRM: contains the same value as the IMFTHTGT variable, which is the name of the address space that causes the MainView Alarm to be generated</td>
</tr>
<tr>
<td></td>
<td>■ ALRT: contains the name of the address space that issues the MainView AutoOPERATOR ALERT (which is also the address space where the Rule fires).</td>
</tr>
<tr>
<td></td>
<td>■ CICS: contains the CICS address space name</td>
</tr>
<tr>
<td></td>
<td>■ CMD: contains the address space that issues the MVS command</td>
</tr>
<tr>
<td></td>
<td>■ DB2: contains the name of the address space issuing the WTOR</td>
</tr>
<tr>
<td></td>
<td>■ EXT: contains the name of the address space where the Rule fires</td>
</tr>
<tr>
<td></td>
<td>■ HWTO: contains the early or interim WTOs</td>
</tr>
<tr>
<td></td>
<td>■ IMP: contains the IMS OM address space name</td>
</tr>
<tr>
<td></td>
<td>■ IMS: contains the IMS address space name</td>
</tr>
<tr>
<td></td>
<td>■ JRNL: contains the name of the MainView AutoOPERATOR PAS address space</td>
</tr>
<tr>
<td></td>
<td>■ MQS: contains the 4-character Queue Manager name followed by MSTR</td>
</tr>
<tr>
<td></td>
<td>■ MSG: contains the name of the address space issuing the WTOR</td>
</tr>
<tr>
<td></td>
<td>■ NVC: contains the NetView address space name</td>
</tr>
<tr>
<td></td>
<td>■ NVM: contains the NetView address space name</td>
</tr>
<tr>
<td></td>
<td>■ Time: contains the name of the MainView AutoOPERATOR PAS address space</td>
</tr>
<tr>
<td></td>
<td>■ VAR: contains the name of the MainView AutoOPERATOR PAS address space</td>
</tr>
<tr>
<td>Variable name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>&amp;IMFCLOCK</td>
<td>contains the number derived from the first word of the MVS Time-of-Day clock. The increment value is approximately once every 1.05 seconds. Each Rule that fires has its own copy of the &amp;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 Rule Processor, specify: Key ===&gt; &amp;IMFRLID::&amp;IMFCLOCK</td>
</tr>
<tr>
<td>&amp;IMFCNTXT</td>
<td>contains the name of the context of the alarm</td>
</tr>
<tr>
<td>&amp;IMFCOLOR</td>
<td>contains the value of the message color as it was modified by the (Message Processing Facility (MPF) or a subsystem in the subsystem interface (SSI); valid values are as follows: RED, BLUE, PINK, GREEN, TURQUOISE, YELLOW, WHITE If the message color is not modified, the value is blank.</td>
</tr>
<tr>
<td>&amp;IMFCONNM</td>
<td>contains the console name to which the WTO was issued Valid only for MVS SP4 and above. &amp;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</td>
</tr>
<tr>
<td>&amp;IMFCSMLU</td>
<td>contains the LU name for SMCS consoles</td>
</tr>
<tr>
<td>&amp;IMFSSOW</td>
<td>contains the eight-character owner ID for subsystem consoles</td>
</tr>
<tr>
<td>&amp;IMFCSTYP</td>
<td>contains the console subtype If the Console Type resolves to Special (IMFCTYPE=P), IMFCSTYP resolves to one of the following values:  ■ INTERNAL  ■ INSTREAM  ■ UNKNOWN  ■ JES3</td>
</tr>
<tr>
<td>&amp;IMFCTYPE</td>
<td>contains the type of console; one of the following possible values: M: Command or WTO was issued from or to an MCS console S: SMCS console E: EMCS console U: Subsystem console P: Special (see definition for the IMFCSTYP variable)</td>
</tr>
<tr>
<td>Variable name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>&amp;IMFDATE</td>
<td>contains the date when the message was issued in the format that is specified on the IMFDATE parameter in BBPARM member AAOPRM00. The following list shows valid formats and an example of each:</td>
</tr>
<tr>
<td></td>
<td>- <code>YYYY/MM/DD</code>; 2050/10/23</td>
</tr>
<tr>
<td></td>
<td>- <code>YYYYMMDD</code>; 205001023</td>
</tr>
<tr>
<td></td>
<td>- <code>DD/MM/YYYY</code>; 23/10/2050</td>
</tr>
<tr>
<td></td>
<td>- <code>DDMMYYYY</code>; 23102050</td>
</tr>
<tr>
<td></td>
<td>- <code>MMDDYYYY</code>; 10232050</td>
</tr>
<tr>
<td></td>
<td>- <code>MM/DD/YYYY</code>; 10/23/2050</td>
</tr>
<tr>
<td></td>
<td>- <code>MMMDDYYYY</code>; OCT232050</td>
</tr>
<tr>
<td></td>
<td>- <code>DDMMMYYYY</code>; 23OCT2050</td>
</tr>
<tr>
<td>&amp;IMFDAY</td>
<td>contains the three-character day of the week: MON, TUE, WED, THU, FRI, SAT, SUN</td>
</tr>
<tr>
<td>&amp;IMFDDNAM</td>
<td>contains the DDNAME specified by the user to generate an external events (EXT event type) EX9T events are generated by using the SUBSYS= parameter on a DD statement in JCL. For more information about EXT events, see “EXT” on page 53.</td>
</tr>
<tr>
<td>&amp;IMFDOMID</td>
<td>contains the DOM ID associated with a WTO that caused a Rule to fire</td>
</tr>
<tr>
<td>&amp;IMFEVFRD</td>
<td>contains the number of Rules that have fired for a specific event</td>
</tr>
<tr>
<td>Variable name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>&amp;IMFETYPE</td>
<td>contains the event type that caused the Rule to fire. If a Rule schedules an EXEC, IMFETYPE contains the value from the Rule that scheduled the EXEC. Possible values for IMFETYPE are: ALRM, ALRT, BRC, CICS, CMD, DB2, EOM, EOS, EXT, HWTO, IMP, IMS, JRNL, MQS, MSG, NVC, NVM, TIME, TLM, VAR. For more information about these event types, refer to “Describing events” on page 35.</td>
</tr>
<tr>
<td>&amp;IMFGROUP</td>
<td>contains the IBM RACF group ID for the address space that issued the event. The group ID is taken from the GROUP= parameter on the job card. For commands issued from the Console address space, the value for IMFGROUP is derived from the RACF RUTOKEN.</td>
</tr>
<tr>
<td>&amp;IMFHILIT</td>
<td>contains the value of the message highlight level as it was modified by MPF or a subsystem in the SSI; valid values are as follows: NONE, BLINKING, REVERSE, UNDERLINE. If the message highlight level is not modified, the value is blank. A value NONE means the message was issued with no highlighting.</td>
</tr>
<tr>
<td>Variable name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>&amp;IMFIMTYP</td>
<td>contains a two-character value describing the type of IMS event. Valid value for this variable is RE; RE means that the event is a command response to an IMS command issued from an EXEC with response. This value can be used to prevent automatic logging of the IMS command response to the Journal by the IMS Generic (or similar) Rule. Specifying <strong>IMFIMTYP NE RE</strong> on the Variable Dependencies panel of an IMS event type Rule causes the Rule not to fire for responses to IMS command that are returned to and processed by an EXEC. Eliminating unnecessary logging to the Journal will reduce the possibility of out-of-storage problems and loss of automation during periods when MainView AutoOPERATOR is processing very large number of IMS response segments.</td>
</tr>
<tr>
<td>&amp;IMFINTEN</td>
<td>contains the value of the message brightness as it was modified by MPF or a subsystem in SSI; valid values are as follows: HIGH or NORMAL. If the message brightness is not modified, the value is blank.</td>
</tr>
<tr>
<td>&amp;IMFJBCOR</td>
<td>contains the 0 to 64-character job correlator token. This variable is valid only when running on IBM z/OS Version 2.1 or later, where its length will be 32 to 64 characters. For versions earlier than 2.1, this variable contains a NULL value (length = 0).</td>
</tr>
<tr>
<td>&amp;IMFJCAN</td>
<td>indicates if the job is canceled: Y means the job is canceled, N means the job is not canceled. The variable is valid only for the EOJ event.</td>
</tr>
<tr>
<td>&amp;IMFJCLAS</td>
<td>contains the job class from the job card of the batch job that has generated the message.</td>
</tr>
<tr>
<td>&amp;IMFJNUM</td>
<td>contains 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. The default setting is IMFJNUM=V. When IMFJNUM=V (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).</td>
</tr>
<tr>
<td>&amp;IMFJSACT</td>
<td>contains accounting information from the EXEC JCL statement.</td>
</tr>
<tr>
<td>&amp;IMFJSCC</td>
<td>contains the condition code. Values can be S0xxx (system abend), Unnnn (user abend), FLUSH (steps were flushed), nnnn (numeric condition code).</td>
</tr>
<tr>
<td>&amp;IMFJSCPU</td>
<td>contains the CPU time used by this job step, in hundredths of seconds. Value is zero (0) for EOJ.</td>
</tr>
<tr>
<td>&amp;IMFJSNUM</td>
<td>contains sequential number of this job step. Value is zero (0) for EOJ.</td>
</tr>
<tr>
<td>Variable name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| &IMFJSTEP | contains the job step name  
Value is blank for EOJ. |
| &IMFJTCPU | contains the total amount of CPU time used by this job, in hundredths of seconds |
| &IMFJTYPE | contains the type of job issuing message:  
J – Batch Job  
T – TSO User  
S – Started Task |
| &IMFLPROD | contains the name of the product associated with the alarm |
| &IMFLTYP | contains the literal value associated with the alarm; possible values can be  
START or STOP |
| &IMFLUSER | contains the user-specified user ID associated with the alarm |
| &IMFMFAU | contains the value of the AUTO parameter in SYS1.PARMLIB member MPFLSTxx  
The value could be YES, NO, or an 8-character token set by the user in SYS1.PARMLIB  
Refer to the chapter "Implementing support for MPF" in the MainView AutoOPERATOR Customization Guide for more information about MPF. |
| &IMFMPFSP | contains the value of the SUP parameter in SYS1.PARMLIB member MPFLSTxx  
The value could be either YES or NO.  
Refer to the chapter "Implementing support for MPF" in the MainView AutoOPERATOR Customization Guide for more information about MPF. |
| &IMFMSGID | contains the first word of the message (up to 16 characters) with any leading plus sign (+) that might be present stripped out of the word  
For example, for the following message:  
+DFHSI1500 TYSPCICS CICS startup is in progress  
The value of the &IMFMSGID variable is DFHSI1500  
Valid only for the MSG and DB2 event types. |
| &IMFMSTYP | contains the two-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 multiline WTO (MLWTO)  
Valid values for the second character can be  
C: command  
R: command response |
<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;IMFNOL</td>
<td>contains the number of LINE variables in the LOCAL pool. Output from multiline messages is placed in LOCAL variables, LINE1 through LINEnnnn.</td>
</tr>
<tr>
<td>&amp;IMFNUSER</td>
<td>contains the NetView user ID assigned to this event</td>
</tr>
<tr>
<td>&amp;IMFOASID</td>
<td>contains the originating Address Space ID (ASID) of the message. For IMFEOM, it is set to the ASID that is being terminated.</td>
</tr>
<tr>
<td>&amp;IMFODATE</td>
<td>contains the date when the message was issued in Julian date format</td>
</tr>
<tr>
<td>&amp;IMFOJOB</td>
<td>for WTOs: contains the job or Started Task that issued the WTO. For CICS messages: 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: 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:</td>
</tr>
<tr>
<td></td>
<td>■ IMS job name for IMS MTO messages</td>
</tr>
<tr>
<td></td>
<td>■ IMS job name for commands (and their responses) entered from MainView AutoOPERATOR</td>
</tr>
<tr>
<td></td>
<td>■ originating LTERM for commands (and their responses) entered from an IMS LTERM</td>
</tr>
<tr>
<td></td>
<td>for Journal messages issued by an EXEC: user ID of the person who scheduled the EXEC</td>
</tr>
<tr>
<td>&amp;IMFOMCLT</td>
<td>contains the name of the client (IMSplex member) where IMS Type-2 command originated. Valid only for IMP events.</td>
</tr>
<tr>
<td>&amp;IMFOMRTE</td>
<td>contains up to 30 characters of the ROUTE list of the IMS Type-2 command. Valid only for IMP events. The ROUTE list is a list of client names separated by commas. The ROUTE list can contain a single asterisk as a client name, which routes to all clients.</td>
</tr>
<tr>
<td>&amp;IMFOMUSR</td>
<td>contains user ID of application where the IMS Type-2 command originated. Valid only for IMP events.</td>
</tr>
<tr>
<td>&amp;IMFOQID</td>
<td>contains the CICS transient data queue name if source of message is CICSTD</td>
</tr>
<tr>
<td>Variable name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>&amp;IMFORGN</td>
<td>contains the possible event types and their descriptions are:</td>
</tr>
<tr>
<td></td>
<td>■ ALRM: name of issuer</td>
</tr>
<tr>
<td></td>
<td>■ ALRT: MainView AutoOPERATOR job name issuing the ALERT</td>
</tr>
<tr>
<td></td>
<td>■ BRC: PAS subsystem ID</td>
</tr>
<tr>
<td></td>
<td>■ CICS: CICS address space job name</td>
</tr>
<tr>
<td></td>
<td>■ CMD: job name of issuer</td>
</tr>
<tr>
<td></td>
<td>■ DB2: DB2 address space job name</td>
</tr>
<tr>
<td></td>
<td>■ EXT: job name of issuer</td>
</tr>
<tr>
<td></td>
<td>■ HWTO: job name of issuer</td>
</tr>
<tr>
<td></td>
<td>■ IMP: contains the IMS OM address space name</td>
</tr>
<tr>
<td></td>
<td>■ IMS CMD: IMS LTERM name</td>
</tr>
<tr>
<td></td>
<td>■ IMS MSG: IMS region name</td>
</tr>
<tr>
<td></td>
<td>■ JRNL: job name on whose behalf the Journal message was issued</td>
</tr>
<tr>
<td></td>
<td>■ MSG: job name of issuer</td>
</tr>
<tr>
<td></td>
<td>■ NVC: job name of issuer</td>
</tr>
<tr>
<td></td>
<td>■ NVM: job name of issuer</td>
</tr>
<tr>
<td></td>
<td>■ TIME: MainView AutoOPERATOR job name</td>
</tr>
<tr>
<td>&amp;IMFORGSS</td>
<td>contains the BBI Subsystem ID of the BBI-SS PAS</td>
</tr>
<tr>
<td>&amp;IMFOTIME</td>
<td>contains the time when the message was issued</td>
</tr>
<tr>
<td></td>
<td>Valid only for messages captured through the Rule Processor.</td>
</tr>
<tr>
<td></td>
<td>The format is</td>
</tr>
<tr>
<td></td>
<td>HH.MM.SS</td>
</tr>
<tr>
<td></td>
<td>for MSG event types. For all other event types, the format is</td>
</tr>
<tr>
<td></td>
<td>HH:MM:SS</td>
</tr>
<tr>
<td>&amp;IMFPCMD</td>
<td>contains the PCMD associated with the alarm</td>
</tr>
<tr>
<td>&amp;IMFPLUS</td>
<td>contains Y if the message ID text begins with a plus sign (+)</td>
</tr>
<tr>
<td>&amp;IMFREPLY</td>
<td>contains the Reply ID of the WTOR message</td>
</tr>
<tr>
<td>&amp;IMFRETAIN</td>
<td>contains Y if the message is marked as retain (and it is not automatically</td>
</tr>
<tr>
<td></td>
<td>scrolled off the MVS Console) and is retained in the Automatic Message</td>
</tr>
<tr>
<td></td>
<td>Retention Facility (AMRF)</td>
</tr>
<tr>
<td>&amp;IMFRLFRD</td>
<td>contains the number of times a Rule was fired</td>
</tr>
<tr>
<td>&amp;IMFRLID</td>
<td>contains the Rule identifier</td>
</tr>
<tr>
<td>&amp;IMFRLMAT</td>
<td>contains the number of times the Rules search criteria was matched</td>
</tr>
<tr>
<td>&amp;IMFRLSET</td>
<td>contains the name of the Rule Set the Rule belongs to</td>
</tr>
<tr>
<td>&amp;IMFRLSTA</td>
<td>contains the Rule status:</td>
</tr>
<tr>
<td></td>
<td>TEST: indicates the Rule is in a TEST state.</td>
</tr>
<tr>
<td></td>
<td>ACTIVE: indicates the Rule is in ACTIVE state.</td>
</tr>
<tr>
<td>Variable name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>&amp;IMFRUSER</td>
<td>contains the user ID from the USER= parameter on the job card. For commands issued from the Console address space, the value for IMFRUSER is derived from the RACF RUTOKEN.</td>
</tr>
<tr>
<td>&amp;IMFSCOPE</td>
<td>contains the name of the scope associated with the alarm</td>
</tr>
<tr>
<td>&amp;IMFSRC</td>
<td>contains the name of the NetView exit that produced the event</td>
</tr>
<tr>
<td>&amp;IMFSTEP</td>
<td>contains the step name that triggered the Rule</td>
</tr>
<tr>
<td>&amp;IMFSTOKN</td>
<td>contains the Address Space STOKEN name. This name is unique for the life of the IPL.</td>
</tr>
<tr>
<td>&amp;IMFSUPR</td>
<td>contains Y if the message text is suppressed from the console</td>
</tr>
</tbody>
</table>
| &IMFSYSID     | contains the possible event types and their descriptions are:  
  - ALRM: name of issuer  
  - ALRT: MainView AutoOPERATOR job name issuing the ALERT  
  - BRC: jobname of PAS  
  - CICS: CICS address space job name  
  - CMD: job name of issuer  
  - DB2: DB2 address space job name  
  - EXT: job name of issuer  
  - HWTO: job name of issuer  
  - IMP: contains the IMS OM address space name  
  - IMS: when a Rule matches an IMC command issued from LTERM, IMS LTERM name  
  - IMS: when a Rule matches an IMC message, IMS region name  
  - JRLN: job name on whose behalf the Journal message was issued  
  - MSG: job name of issuer  
  - NVC: job name of issuer  
  - NVM: job name of issuer  
  - TIME: MainView AutoOPERATOR job name |
<p>| &amp;IMFTCPCT     | contains the number of times this address space has already been extended by a Rule for CPU expired |
| &amp;IMFTCPSC     | contains the number of CPU seconds this address has already been extended by a Rule |
| &amp;IMFTERMT     | contains the type of address space termination; valid values are NORMAL or ABNORMAL |
| &amp;IMFTEXT      | message text that caused the EXEC to be scheduled. For CICS TD events, the maximum length for IMFTEXT can be up to 512 characters. For all other message types, IMFTEXT is 125 characters. |
| &amp;IMFTHTGT      | resolves to the address space name that causes the MainView alarm to be generated |</p>
<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;IMFTTIME</td>
<td>contains the time when the message was issued in HH:MM:SS format</td>
</tr>
<tr>
<td>&amp;IMFTOKEN</td>
<td>contains the token ID of the message. Same as hardcopy ID Used to attach multiline WTO Minor/Major Lines.</td>
</tr>
</tbody>
</table>
| &IMFTSNUM     | contains the sequential step number  
**Note:** The IMFSTEP variable is also populated. |
| &IMFTSSID     | contains the subsystem ID (SSID) that selected the job  
Possible values are as follows: JES2, JES3, ASCH (APPC), OMVS, STC, or TSO. The value can also be the jobname if none of these apply the jobname is less than 4 characters long. |
| &IMFTTLM      | contains the type of expired time limit; valid values are as follows:  
- JCPU: job processor time limit expired  
- SCPU: step processor time limit expired  
- WAIT: Continuous Wait Time expired |
| &IMFTWTCT     | contains the number of times this address space has already been extended by a Rule for Wait limit expired |
| &IMFTWTSC     | contains the number of seconds this address has already been extended by a Rule for WAIT limit expired |
| &IMFVALUE     | contains the value of a SHARED variable when an event type VAR Rule fires |
| &IMFVIEW      | contains the name of the view associated with the alarm |
| &IMFXDESC     | contains a two-character value that consists of the exact hexadecimal contents of the descriptor code field of the WQE (WQEDESCD) for a WTO  
This variable allows the testing of specific bits in the descriptor code, by using the FN and FO operators on the Selection Criteria panel or the BITON() function on the Advanced Variable Dependency panel, to determine eligibility of the Rule to fire.  
For example, if you want to Rule to fire when descriptor code 12 is used you could specify the following value on the Variables Dependency panel:  
IMFXDESC_______________________ FO  
0010___________________________  
Or, you could specify the following value on the Advanced Variables Dependency panel:  
BITON(&IMFXDESC,0010)  
Either method causes the Rule to fire. |
### Event variables and slot names for BRC events

The following table describes the event variables used by BRC-initiated Rules.

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;IMFXOJOB</td>
<td>contains the name of the original job or Started Task that requested the WTO to be issued by another address space. The contents of IMFXOJOB are only meaningful if the WTO is issued by another address space, otherwise its contents are identical to IMFOJOB. IMFXOJOB is valid only for MSG and JRNL type Rules. When a JRNL event originates from a MainView Total Object Manager (TOM) log message, the &amp;IMFXOJOB variable contains the TOMID of the TOM PAS that originated the message. To determine if a JRNL event originated from a local TOM PAS, use the Variable Dependencies (SV) panel in the Rules Processor to compare the IMFXOJOB variable to IMFXOJOB (IMFOJOB EQ IMFXOJOB). If the values are equal, the message originated from the local TOM PAS.</td>
</tr>
<tr>
<td>&amp;IMFXROUT</td>
<td>contains a 16-character value that consists of the exact hexadecimal contents of the routing code field of the WQE (WQEERC) for a WTO. This variable allows the testing of specific bits in the route code, by using the FN and FO operators on the Selection Criteria panel or the BITON() function on the Advanced Variable Dependency panel, to determine eligibility of the Rule to fire. For example, if you want to Rule to fire when route codes 2 and 11 are used you could specify the following value on the Variables Dependency panel: IMFXROUT___________________ FO 40200000000000000000000000000000___ Or, you could specify the following value on the Advanced Variables Dependency panel: BITON(&amp;IMFXROUT,4020) Either method causes the Rule to fire.</td>
</tr>
<tr>
<td>&amp;SYSDATE</td>
<td>contains the current date in the form month/day/year</td>
</tr>
<tr>
<td>&amp;SYSTIME</td>
<td>contains the current time of day expressed as hh:mm:ss</td>
</tr>
<tr>
<td>WORDxxx</td>
<td>represents a number (for example, &amp;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 &amp;WORD3 should resolve to the third word of the text of the event. Word variables are delimited by blanks and commas in the text. The message text used to return WORD variables is a maximum length of 512 for CICS TD messages and 255 for all others.</td>
</tr>
</tbody>
</table>
### Table 16: Event variables for BRC initiated Rules

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Variable value</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMFSEVERITY</td>
<td>severity</td>
<td>16</td>
</tr>
<tr>
<td>IMFCELLNAME</td>
<td>cell name</td>
<td>48</td>
</tr>
<tr>
<td>IMFCELLTYPE</td>
<td>cell type</td>
<td>8</td>
</tr>
<tr>
<td>IMFTARGET</td>
<td>target</td>
<td>70</td>
</tr>
<tr>
<td>IMFHOSTNAME</td>
<td>host name</td>
<td>48</td>
</tr>
<tr>
<td>IMFHOSTADDR</td>
<td>host IP address</td>
<td>48</td>
</tr>
<tr>
<td>IMFHOSTCLASS</td>
<td>host class</td>
<td>33</td>
</tr>
<tr>
<td>IMFTOOLNAME</td>
<td>tool name</td>
<td>48</td>
</tr>
<tr>
<td>IMFTOOLSEV</td>
<td>tool severity</td>
<td>16</td>
</tr>
<tr>
<td>IMF_</td>
<td>see “Converting slots to event variables” on page 182 for more information</td>
<td>32</td>
</tr>
</tbody>
</table>

### Converting slots to event variables

MainView AutoOPERATOR converts slots in the incoming BRC event to event variables. Rule-initiated EXECs can also use these variables.

Event variables consist of the following parts:

- **variable name**: MainView AutoOPERATOR converts the slot name to the variable name. The slot name is prefixed by the characters IMF_. The variable uses uppercase characters and is truncated at 32 characters.

  For example, MainView AutoOPERATOR converts a slot name of `mv_sample='Test'` to `IMF_MV_SAMPLE="Test"`.

  In another example, `mv_a_very_very_very_long_slot_name='Long slotname'` would be converted to `IMF_MV_A_VERY_VERY_VERY_LONG_SLO="Long slotname"`.

- **variable content**: MainView AutoOPERATOR uses the slot value as the variable value. The variable values are not translated to uppercase characters and contain
the values as they are sent from the creator of the BRC event. The length of the value is truncated at 255 characters.

If the slot contains an array, only the first entry in the array is stored in the variable. For example, the array `mv_array=['abc','def','ghi']` would be converted to `IMF_MV_ARRAY="abc"`

You should ensure that there is enough information in the event to determine such things as where the event originated (such as `mc_host`). MainView AutoOPERATOR does not validate any fields.

### List of variables for ALRT-initiated Rules

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Contents</th>
<th>Length/Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMFKEY</td>
<td>key of the ALERT</td>
<td>1-64 / character</td>
</tr>
<tr>
<td>AMFTEXT</td>
<td>text of the ALERT</td>
<td>0-255 / character</td>
</tr>
<tr>
<td>AMFALARM</td>
<td>alarm value of the alert</td>
<td>1 / Y (YES) or N (NO)</td>
</tr>
<tr>
<td>AMFCOLOR</td>
<td>color of ALERT</td>
<td>6 / as specified by COLOR parameter</td>
</tr>
<tr>
<td>AMFCURIN</td>
<td>current interval number when the AMEVENT variable value is ESC or FINAL This variable is not created for other ALERT event sub-types (such as ADD, DELETE, and so on).</td>
<td>1 / character 1-5 (normal escalation interval) or &quot;F&quot; (final escalation)</td>
</tr>
<tr>
<td>AMFEDIR</td>
<td>increase or decrease the priority of the ALERT when it is escalated</td>
<td>1 / character (U or D)</td>
</tr>
<tr>
<td>AMFEDISP</td>
<td>keep or delete the ALERT at the final escalation level</td>
<td>1 / character (K or D)</td>
</tr>
<tr>
<td>AMFEEEXEC</td>
<td>name of EXEC and EXEC parameters scheduled at final escalation priority</td>
<td>0 to 255 / character</td>
</tr>
<tr>
<td>AMFEINT1 AMFEINT2 AMFEINT3 AMFEINT4 AMFEINT5 AMFEINT6</td>
<td>number (in minutes) from 0 to 9999</td>
<td>4 / numeric (or null)</td>
</tr>
<tr>
<td>AMFEXEC</td>
<td>EXEC and EXEC parameters associated with the ALERT</td>
<td>0-256 / character</td>
</tr>
<tr>
<td>Name</td>
<td>Contents</td>
<td>Length/Format</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>AMFEVENT</td>
<td>type of ALERT event; for example: DELETEQ</td>
<td>3-7 / ADD, DELETE, DELETEQ, ESC, FINAL</td>
</tr>
<tr>
<td>AMFHELP</td>
<td>extended Alert member name</td>
<td>8 / character</td>
</tr>
<tr>
<td>AMFDATE</td>
<td>date ALERT was issued</td>
<td>9 / dd-mmm-yy</td>
</tr>
<tr>
<td>AMFITIME</td>
<td>time ALERT was issued</td>
<td>8 / hh:mm:ss</td>
</tr>
<tr>
<td>AMFORGN</td>
<td>origin of ALERT</td>
<td>1-8 / character</td>
</tr>
<tr>
<td>AMFPCMD</td>
<td>primary command specified in ALERT</td>
<td>0-256 / character</td>
</tr>
<tr>
<td>AMFPRIOR</td>
<td>priority of ALERT</td>
<td>13 / as specified in PRIORITY parameter</td>
</tr>
<tr>
<td>AMFPSYS</td>
<td>value for SYSTEM keyword (could be either YES or NO)</td>
<td>1 / character (Y or null)</td>
</tr>
<tr>
<td>AMFPTG1</td>
<td>names of the cells in the TGTPUB keyword, or the TARGETS field from the Rule panel</td>
<td>1 - 8 / character</td>
</tr>
<tr>
<td>AMFPTG2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMFPTG3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMFPTG4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMFPTG5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMFPTG6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMFPUB</td>
<td>value of the Publish keyword when an ALERT is created</td>
<td>2-7/ADD, REPLACE, or NO</td>
</tr>
<tr>
<td>AMFQUEUE</td>
<td>name of queue for ALERT</td>
<td>8 / character</td>
</tr>
<tr>
<td>AMFRTAIN</td>
<td>specifies whether or not to retain an ALERT across BBI-SS PAS warm and cold starts</td>
<td>1 / character (Y or N)</td>
</tr>
<tr>
<td>AMFSSID</td>
<td>system from which ALERT was issued</td>
<td>8 / character</td>
</tr>
<tr>
<td>AMFTGT</td>
<td>target to which ALERT was issued</td>
<td>1-8 / character</td>
</tr>
<tr>
<td>AMFUDATA</td>
<td>user data string</td>
<td>0-256 / character</td>
</tr>
<tr>
<td>AMFUSER</td>
<td>name of the user ID that the ALERT is addressed to</td>
<td>8 / character</td>
</tr>
</tbody>
</table>

**List of variables for HWTO-initiated Rules**

The following table shows the variables available when you create Rules for the HWTO event type.
Using Rules functions

The following sections describe general syntax rules for using Rules functions on the Selection Criteria panel (S1), the Advanced Variable Dependencies (ASV) panel, and the Advanced Set Variables (AAV) panel and how to use operators, comparison operators, connector operators, and built-in Rules functions.

Understanding syntax

The following rules apply when coding and evaluating strings or equations:

Table 18: Variables available for creating Rules for HWTO events

<table>
<thead>
<tr>
<th>Name</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMFRLID</td>
<td>Rule name</td>
</tr>
<tr>
<td>IMFRLSET</td>
<td>Rule Set name</td>
</tr>
<tr>
<td>IMFTEXT</td>
<td>Message text</td>
</tr>
<tr>
<td>IMFXOJOB</td>
<td>Job name</td>
</tr>
<tr>
<td>IMFORGSS</td>
<td>Name of the BBI-SS PAS</td>
</tr>
<tr>
<td>IMFCONNMM</td>
<td>Console name</td>
</tr>
<tr>
<td>IMFETYPE</td>
<td>Event type (which will be HWTO)</td>
</tr>
<tr>
<td>IMFIMPFAU</td>
<td>MPF token</td>
</tr>
<tr>
<td>IMFMSTYP</td>
<td>WTO type</td>
</tr>
<tr>
<td>IMFODATE</td>
<td>Date in YYDDD format</td>
</tr>
<tr>
<td>IMFOTIME</td>
<td>Time in hh:mm:ss format</td>
</tr>
<tr>
<td>IMF4DATE</td>
<td>Date in YYYYDDD format</td>
</tr>
<tr>
<td>IMFTIME</td>
<td>hh:mm:ss</td>
</tr>
<tr>
<td>IMFDATE</td>
<td>yyyyymmdd</td>
</tr>
<tr>
<td>IMFODESC</td>
<td>Descriptor codes</td>
</tr>
<tr>
<td>IMFRLFRD</td>
<td>Rule fired count</td>
</tr>
<tr>
<td>IMFRLSTA</td>
<td>Rule status (active or test)</td>
</tr>
</tbody>
</table>
- White space (blanks) is permitted between operands, operators, and function arguments to improve readability.

- Use single quotation marks before and after any string that you code that contains one or more reserved words, operators, or comparator symbols to have them treated as a literal. Literals specified in this way will be used as is and are not interpreted (resolved). White space can be embedded in literals that use single quotation marks as well. For example:

  - 'THIS IS A TEST MESSAGE'
  - 'SUBSTR() SHOULD ALWAYS BE = 12'

- Single quotation marks should not be used anywhere in the arguments to a function except when using the VALUE() function. If single quotation marks are needed as the strip character (argument #3) for the STRIP() function, you should use the X() function of X(7D) to do this.

- Side-by-side single quotation marks (no data between them) may be used to specify a null value. The NULL() function can also be used for this purpose. For example, you can test a variable for a null value setting in two ways:

  - &MVYVAR<>NULL()
  - &MVVAR<"'

- Double quotation marks are not supported. If a double quotation mark is needed, use the X() function of X(7F) instead.

- Built-in Rules functions can be nested. For example, SUBSTR(&WORD1,POS(&WORD1,A),3) is a permitted expression.

- Built-in Rules functions can use a variable as any argument of the function as needed. For example, SUBSTR(&WORD1,&WORD2,4).

- All variable names must be preceded with an ampersand (&) to be considered a variable.

- Built-in Rules function arguments are always separated with a comma (,). The use of a semicolon is not supported.

- All comparisons are done by using like types. You must ensure that each side of the equation is in like type by using built-in Rules functions if necessary; for example:
— If both operands are numeric (characters 0-9), a numeric comparison is done. For example, if the equation resolves to 23>10, a numeric comparison is calculated.

The maximum value that can be numerically compared is 2,147,483,647. For values larger than this, the two operands are compared as character strings.

— If one operand contains a real number (has a decimal point and digits to the right of it) and the other operand is numeric, a real number numerical comparison is done.

— If either operand contains non-numeric characters, a simple string comparison is done. Strings of different lengths will have the shorter string padded with blanks to the right to match the longer length operand.

**Operators**

This topic provides a table that shows the valid operators that can be used and the data type required to use them.

The following restrictions apply when using these operators:

- Math operators can only be used on integer values.
- Math that uses real numbers (those with a decimal point in them) is not supported.
- Math with *null* operands is unpredictable and therefore returns a numeric result of 0.

**Table 19: Supported operators**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Operation</th>
<th>Required operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Add</td>
<td>2 numeric integer operands</td>
</tr>
<tr>
<td>-</td>
<td>Subtract</td>
<td>2 numeric integer operands</td>
</tr>
<tr>
<td>*</td>
<td>Multiply</td>
<td>2 numeric integer operands</td>
</tr>
<tr>
<td>/</td>
<td>Divide</td>
<td>2 numeric integer operands</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>()</td>
<td>Grouping</td>
<td>Parentheses may be used to indicate precedence as needed</td>
</tr>
</tbody>
</table>

**Comparison operators**

This topic provides a table that shows the valid operators that can be used.
Comparisons are done using the most precise, like data type possible that can be determined at run time. For example, two numeric integer operands will have a true numeric comparison done. Two numeric real-number operands will have a real-number numeric compare done. If one or both of the operands is non-numeric, a simple character compare is done with the shorter operand padded with blanks to the right if needed.

Table 20: Supported comparison operators

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>Equal to</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less than</td>
</tr>
<tr>
<td>&lt;=, =&lt;</td>
<td>Less than or equal</td>
</tr>
<tr>
<td>&gt;=, =&gt;</td>
<td>Greater than or equal</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>Not equal</td>
</tr>
<tr>
<td>!&gt;</td>
<td>Not greater than</td>
</tr>
<tr>
<td>!&lt;</td>
<td>Not less than</td>
</tr>
</tbody>
</table>

Connector operators

This topic provides a table that shows the list of supported operators.

Table 21: Supported connector operators

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND</td>
<td>Logical AND</td>
<td>Both sides of the connector must be true for the end condition to be TRUE otherwise the end condition is FALSE.</td>
</tr>
<tr>
<td>OR</td>
<td>Logical OR</td>
<td>If either side of the connector is TRUE then the end condition is TRUE otherwise the end condition is FALSE.</td>
</tr>
</tbody>
</table>

Reserved words

To use a reserved word in any formula you must enclose the reserved word in single quotation marks.

All built-in Rules function names, operators, comparison operators, and connector operators are reserved words. Additionally, the exclamation point (!) is also a...
reserved word. It can still be used to construct a compound variable however. BMC recommends that the REXX() built-in be used on the ASV panel to construct compound variables instead of the exclamation point. The REXX() function is more intuitive and flexible and will result in fewer coding problems in your rules. An exclamation point should always be enclosed in single quotation marks unless it is used in a compound variable or as a comparison operator ( != !< !> ).

List of built-in Rules functions

This topic describes built-in Rules functions and how each can be used on the Advanced Set Variables panel.

**AVG**

This function returns an integer that represents the average of all the values provided as arguments. This function uses the following syntax:

```
AVG(value1, value2, value3, ..., valueX)
```

Arguments to this function can be any variable or function result. The value must be a numeric integer. Invalid values are ignored and not considered in computing the average.

No rounding of the result is done. Any number of arguments can be provided. If no arguments are provided or all arguments are non-numeric, the function returns zero (0).

**Example**

```
(&WORD3 = 11 and &WORD7 = 14)
AVG(&WORD3, &WORD7) returns 12
```

**BITON**

This function tests for one or more bits in a binary value and you can use it in when you need to determine whether bit flags are on or off in a flag byte. This function uses the following syntax:

```
BITON(string1, string2)
```

The `string1` value can be any variable or function result. No data conversion is automatically performed on `string1`.

The `string2` value is coded as a string of hexadecimal digit pairs with two characters being a single hexadecimal byte. The length of `string2` must be an even number of
characters. The specified string2 is converted to hexadecimal internally and is used to test the matching bits in string1. The string2 value is padded to the right if needed to match the length of string1.

---

**Example**

(&WORD3 = x'02C344')

BITON(&WORD3, 0003) returns 1 (all bits in string2 are on in string1)

BITON(&WORD3, 80) returns 0 (no bits in string2 are on in string1)

---

**C2D**

This function converts a character string to its decimal equivalent. This function uses the following syntax:

C2D(string)

---

**Example**

C2D(B) returns 194

---

**C2X**

This function converts a character string to its hexadecimal string equivalent. This function uses the following syntax:

C2X(string)

---

**Example**

C2X(A123) returns 'C1F1F2F3'

---

**CNVTUOM | CUOM | CU**

This function returns a string that represents the value of the selected string that has had the scaling unit-of-measure converted to single units. The maximum value is 9,223,372,036,854,775,807 (one byte less than 8192PI).

If you specify a value that is too large, the function returns the original string and no conversion will be done.

---

**Note**

Floating point numbers are not supported. For example, 23.87K is not a valid value for this function.

---

This function uses the following syntax:
CUOM(value|variable)

where value|variable is any constant or variable

The following lists the supported suffixes:

- K: value * 1000
- Ki: value * 1024
- M: value * 1,000,000
- Mi: value * 1,048,576
- G: value * 1,000,000,000
- Gi: value * 1,073,741,824
- T: value * 1,000,000,000,000
- Ti: value * 1,099,511,627,776
- P: value * 1,000,000,000,000,000
- Pi: value * 1,125,899,906,842,624

---

**Note**

The values above are shown with commas for clarity. Converted values do not include commas.

If the supplied value does not have a suffix or the suffix is not recognized, the function returns the original string and without any conversion.

Suffix values are case-insensitive.

---

**Example**

The Rule variable WORD5 contains a string of 80Ki. To test the value for greater than 50,000, enter the following function:

```
CUOM(&WORD5) > CUOM(50K)
```

The first CUOM function converts 80Ki to 81920 (80*1024) and returns 81920 for evaluation in the statement.

The second CUOM function converts 50K to 50 * 1000 and returns 50000. After resolving the values, the comparison statement is evaluated for true or false.

---

**DATE**

This function returns a character string representing the current date in the format specified. This function uses the following syntax:

```
DATE(format)
```

where format can be one of the following values:

- J: Julian date; formatted as YYYY/DDD
- JU: Julian date; formatted as YYYYDDD (no slashes)
- U: United States date; formatted as MM/DD/YYYY
- UU: United States date; formatted as MMDDYYYY (no slashes)
- US: United States date (short); formatted as MM/DD/YY
- USU: United States date (short); formatted as MMDDYY (no slashes)
- E: European date; formatted as DD/MM/YYYY
- EU: European date; formatted as DDMMYYYY (no slashes)
- ES: European date (short); formatted as DD/MM/YY
- ESU: European date (short); formatted as DDMMYY (no slashes)
- S: Sortable date; formatted as YYYY/MM/DD
- SU: Sortable date; formatted as YYYYMMDD (no slashes)
- SS: Sortable date (short); formatted as YY/MMDD
- SSU: Sortable date (short); formatted as YYMMDD (no slashes)
- M: Military date; formatted as DD-MMM-YYYY
- MU: Military date; formatted as DDMMYYYY (no dashes)
- MS: Military date (short); formatted as DD-MMM-YY
- MSU: Military date (short); formatted as DDMMMYY (no dashes)

The format code is optional and, if omitted, the U format is used as the default.

**Example**

\[
\text{DATE(SU)} \quad \text{returns} \quad 20081208 \\
\text{DATE(M)} \quad \text{returns} \quad 12-DEC-2008
\]

---

**D2C**

This function converts the supplied numeric value to the EBCDIC equivalent character string. Only a single numeric value can be supplied. The value must be in the range of 0-256. Not all values in this range will result in a printable character.

This function uses the following syntax:

\[
\text{D2C}(\text{string})
\]

**Example**

\[
\text{D2C}(194) \quad \text{returns} \quad B
\]

---

**D2X**

This function converts the supplied numeric value to the 2-character hexadecimal equivalent character string. Only a single numeric value can be used and the value must be in the range of 0-256.
This function uses the following syntax:

\[ \text{D2X}(\text{string}) \]

**Example**

\[ \text{D2X}(194) \] returns \( C2 \)

### EVDATE or ED

This function returns a character string representing the date of the event being handled in the format specified by format. The format code is optional and, if omitted, uses the "U" format as the default.

This function uses the following syntax:

\[ \text{EVDATE}() \]

where \( \text{format} \) can be:

- J: Julian date; formatted as \( YYYY/DDD \)
- JU: Julian date; formatted as \( YYYYDDD \) (no slashes)
- U: United States date; formatted as \( MM/DD/YYYY \)
- UU: United States date; formatted as \( MMDDYY \) (no slashes)
- US: United States date (short); formatted as \( MM/DD/YY \)
- USU: United States date (short); formatted as \( MMDDYY \) (no slashes)
- E: European date; formatted as \( DD/MM/YYYY \)
- EU: European date; formatted as \( DDMMYYYY \) (no slashes)
- ES: European date (short); formatted as \( DD/MM/YY \)
- ESU: European date (short); formatted as \( DDMMYY \) (no slashes)
- S: Sortable date; formatted as \( YYYY/MM/DD \)
- SU: Sortable date; formatted as \( YYYYMMDD \) (no slashes)
- SS: Sortable date (short); formatted as \( YY/MMDD \)
- SSU: Sortable date (short); formatted as \( YYMMDD \) (no slashes)
- M: Military date; formatted as \( DD-MMM-YYYY \)
- MU: Military date; formatted as \( DDMMYYYY \) (no dashes)
- MS: Military date (short); formatted as \( DD-MMM-YY \)
- MSU: Military date (short); formatted as \( DDMMMYY \) (no dashes)

**Example**

\[ \text{EVDATE}(SU) \] returns \( 20081208 \)

\[ \text{EVDATE}(M) \] returns \( 12-DEC-2008 \)
**EVTIME or ET**

This function returns a character string that represents the time of the event, formatted according to type. The type code is optional and, if omitted, the event time is formatted with a type of N.

This function uses the following syntax:

```
EVTIME(type)
```

where `type` can be:

- N: normal; formatted as `HH:MM:SS`
- NU: normal undelimited; formatted as `HHMMSS`

**Example**

```
EVTIME(N) returns 08:15:22 (event time)
ET() returns 08:15:22 (event time)
ET(NU) returns 081522 (event time unformatted)
```

**EXTRACT | EX**

This function returns a variable-length string extracted from another string for evaluation based on a beginning and ending delimiter in the source string.

This function uses the following syntax:

```
EXTRACT(value|variable, delim1, delim2[,start])
```

where:

- `value|variable`: any constant or variable that is the source for the extraction
- `delim1`: any single character that identifies the beginning of the string to be extracted.
  - If not specified, an open parenthesis “(“ is assumed by default.
- `delim2`: any single character that identifies the end of the string to be extracted.
  - If not specified, a closed parenthesis ”)” is assumed by default.
- `start`: an optional value for the start position in the source where the extraction will begin. The default is 1.

If `delim1` cannot be located within the source string, the function returns a single blank for a value. If `delim2` cannot be located, the function returns whatever value
was found (up to the first blank) while looking for delim2. This is useful when delim2 is missing (due to an intervening blank space).

**Note**

If the beginning of the start location has more than one occurrence of delim1, the EXTRACT function returns the value from the first occurrence of delim1 and delim2. Rule processing ignores all other occurrences.

If the start value specifies a value beyond the end of the value or variable, the EXTRACT function uses a value of 1 and proceeds with the evaluation.

---

**Example**

The Rule variable WORD3 contains a value of INIT(24). To access the value 24, enter the following function:

```
EX(&WORD3)
```

The Rule variable WORD3 contains a value of USER(“JOHN DOE”). To access the value JOHN DOE, enter the following function:

```
EX(&WORD3,X(7F),X(7F))
```

In this example, the X() function represents the double quotation marks because double quotation marks is a reserved symbol.

The Rule variable &IMFTEXT has the following value:

```
MSG0001 USER(abcde) ASSIGNED WORK FROM USER(ghijkl)
```

If you enter the function:

```
EX(&IMFTEXT,'(','')
```

The function returns data from the first occurrence which is “abcde”.

### INRGRPL

This function tests whether the user ID, associated with the issuer of the Rule event, belongs to a RACF Group that is referenced by the ACEEFCGP table.

This function accepts one argument. The argument can be up to 8 characters long and must be enclosed in single quotes; asterisk (*) and plus sign (+) are also supported. If you specify an asterisk, it must as the last character.

This function uses the following syntax:

```
INRGRPL(groupName)
```

If the argument is found in the RACF Group list, the return code is 1; otherwise, the return code is 0.

The specified argument is compared against the list of RACF Group names associated with the event origin address space, as indicated in the ACEEFCGP field of the ACEE.
Example

\[(\text{INRGRPL('DEV1')} \text{ OR INRGRPL('DEV2')} ) \text{ AND INRGRPL('PROD12')}\]
\[\text{INRGRPL('DEV+')=1 AND INRGRPL('PROD*')=1}\]
\[\text{INRGRPL('DEV*')}\]

**LC**

This function converts the character string to all lowercase letters. This function uses the following syntax:

\[\text{LC(string)}\]

**Example**

\&WORD1 = SystemB
\[\text{LC(&WORD1) returns systemb}\]

**LEFT**

This function returns \textit{len} characters from string starting at the left end. This function uses the following syntax:

\[\text{LEFT(string, len)}\]

**Example**

\&WORD1 = SYSTEMB
\[\text{LEFT(&WORD1,3) returns SYS}\]

**LEN**

This function returns the length of the character string. This function uses the following syntax:

\[\text{LEN(string)}\]

**Example**

\&WORD1 = SYSTEMB
\[\text{LEN(&WORD1) returns 7}\]
**LIKE | LK**

This function compares the string in argument 1 to a pattern in argument 2. The pattern can include wildcard characters (+) to identify a location to be skipped or a generic character (*) to identify a number of characters to be skipped.

The following lists the possible returned values:

- 1: TRUE indicating that the pattern is found in the string
- 0: FALSE indicating that the pattern is not found in the string

This function uses the following syntax:

\[
\text{LIKE(value|variable, pattern)}
\]

where:

- *value|variable* is any constant or variable that is the source string for the matching operation.
  - If the value includes blanks spaces, you must enclose it with single quotation marks. You do not need to use quotation marks if the value is a variable name,
- *pattern* is any constant or variable that contains the pattern to be matched.

### Example

The Rule variable IMFTEXT contains the following string:

```plaintext
$HASP375 JOBAINV2 ESTIMATE EXCEEDED BY 8,000 LINES
```

To pattern match &WORD2 (JOBAINV2) for any value that starts with the characters JOB, has any letter or number in the 4th position followed by INV, enter the following function:

\[
\text{LIKE(&WORD2,'JOB+INV*')}
\]

This function matches values such as JOBAINV2, JOBXINV9, JOBQINV, etc.

### Example

To pattern match for a string without specifying where it occurs in the message, use a leading and trailing asterisk (*). For example, using the same IMFTEXT variable as above, enter the following function:

\[
\text{LIKE(&IMFTEXT,'*JOB+INV*')}
\]

This statement matches values such as JOBAINV2, JOBXINV9, and JOBQINV regardless of where the value occurs in the message text.
**MAX**

This function returns the integer value of the argument that is the largest. Any number of arguments may be provided. If no arguments are provided or all arguments are non-numeric, the function returns zero (0).

This function uses the following syntax:

\[
\text{MAX}(\text{value}_1, \text{value}_2, \text{value}_3, \ldots, \text{value}_X)
\]

Arguments to this function can be any variable or function result. The value must be a numeric integer. Invalid values are ignored.

---

**Example**

&WORD3 = 11 and &WORD7 = 14
\[\text{MAX}(&\text{WORD3}, &\text{WORD7}) \text{ returns 14}\]

---

**MIN**

This function returns the integer value of the argument that is the smallest. Any number of arguments may be provided. If no arguments are provided or all arguments are non-numeric, the function returns zero (0).

This function uses the following syntax:

\[
\text{MIN}(\text{value}_1, \text{value}_2, \text{value}_3, \ldots, \text{value}_X)
\]

Arguments to this function can be any variable or function result. The value must be a numeric integer. Invalid values are ignored.

---

**Example**

&WORD3 = 11 ,&WORD7 = 14, &WORD9 = 'abc'
\[\text{MIN}(&\text{WORD3}, &\text{WORD7}) \text{ returns 11}\]
\[\text{MIN}(&\text{WORD7}, &\text{WORD9}) \text{ returns 14}\]

---

**NULL**

This function returns a null string and you can use it to determine whether another value has a value assigned it. This function takes no arguments. You can also enter two single quotation marks (" ) with no intervening data or space) can also be used to indicate a null string.

This function uses the following syntax:
&MYVAR NULL() shows how to create a Rule to fire if variable &MYVAR has a non-null value assigned to it.

PARSE | PA

This function parses the specified string into words and returns the specified word from the source string. Words are numbered starting at 1 and delimited by white space.

Syntax: PARSE(value | variable, number)

- \textit{value | variable} is any constant or variable that is the source string for the parsing operation.
  
  If the value includes blanks spaces, you must enclose it with single quotation marks. You do not need to use quotation marks if the value is a variable name,

- \textit{number} is the relative number of the desired word.
  
  Words within the string are numbered starting at 1. If the value for number exceeds that number of words in the string, the function returns a single blank.

Example

The Rule variable IMFTEXT contains the following string:

\$HASP375 JOBPROD1 ESTIMATE EXCEEDED BY 8,000 LINES

To access the value 8,000 within the string as a numeric value, enter the following function:

PARSE(STRIP(&IMFTEXT,A,','),6)

The function strips any of the commas from the text and parses the string into words. The function returns the value of 8000 from the 6th word in IMFTEXT.

POS

This function returns the position of \textit{string2} in \textit{string1}. This function uses the following syntax:

POS(string1,string2)

This function returns 0 if \textit{string2} cannot be found in \textit{string1}. The search always begins with the leftmost character of \textit{string1}.
Example

&WORD3 = ABCD123DEF
POS(&WORD3,123) returns 5
SUBSTR(&WORD3,POS(&WORD3,1),3) returns 123

REXX | RX

This function creates a compound variable name that can then be used to obtain a value for inclusion in the equation. This function uses the following syntax:

REXX(string)

Example

(&WORD1=SJSD, &WORD2=CICS, &SJSD.SYSTEM=RUNNING, &SJ.SYSTEM.CICS=DEAD)
RX(&WORD1||.SYSTEM) returns RUNNING
RX(SUBSTR(&WORD1,1,2)||.SYSTEM.||&WORD2) returns DEAD

Note that the REXX() function supports variable substitution for the first node of the compound variable but this function is not supported in an EXEC or CLIST because the REXX() built-in function cannot be used in those environments. In an EXEC or CLIST, the first node of the compound variable must be a static constant.

RIGHT

This function extracts len characters from string starting at the right end. This function uses the following syntax:

RIGHT(string,len)

Example

(&WORD1 = SYSTEMB)
RIGHT(&WORD1,5) returns STEMB

STRIP | STP

This function removes a character from the string as indicated by the options.

If the option and char values are omitted, blanks are stripped from the lead and tail of the string. There are three formats to the function. All arguments must be specified for the chosen format; that is, you cannot skip an argument by using a double comma such as STRIP(&WORD1,,!). The char value must not be any of the operator or comparator symbols (+, -, =, and so on) or single or double quotation
marks. If double quotation marks are necessary, use the X() function to specify the reserved character.

This function uses the following syntax:

\[
\text{STRIP}(\text{value}, \text{option, char})
\]

\[
\text{STRIP}(\text{value, option})
\]

\[
\text{STRIP}(\text{value})
\]

The following options are valid:

- B: remove leading and trailing char from the string
- L: remove leading char from string
- T: remove trailing char from string
- A: remove char from anywhere it is found in the string

**Example**

\[
&\text{WORD5} = "\text{MYSTRING}\" \text{then STRIP}(&\text{WORD5}, B, X(7F)) \text{returns MYSTRING}
\]

\[
&\text{WORD5} = \text{MYSTRING?} \text{then STRIP}(&\text{WORD5}, T, ?) \text{returns MYSTRING}
\]

\[
&\text{WORD5} = ???\text{MYSTRING}!!! \text{then STRIP}(\text{STRIP}(&\text{WORD5}, L, ?), T, !) \text{returns MYSTRING}
\]

\[
&\text{WORD5} = '\text{MYSTRING}' \text{then STRIP}(&\text{WORD5}, B, X(7D)) \text{returns MYSTRING}
\]

\[
&\text{WORD5} \text{has imbedded periods such as } \ldots\text{MY}\ldots\text{STRING}\ldots\text{DATA}\ldots \text{then STRIP}(&\text{WORD5}, A, .) \text{returns MYSTRINGDATA}
\]

**SUBSTR | SB**

This function returns a portion of the string beginning at the start position for the specified length.

Commas are used to separate the arguments of the function. All arguments must be specified for the chosen format; that is, you cannot skip an argument by using a double comma (for example, SUBSTR(&WORD1, 5) is not valid). By default, if the length value is omitted (second format shown), the function returns the rest of the string based on the start position value.

This function uses the following syntax:

\[
\text{SUBSTR}(\text{value}, \text{startPositionLength})
\]

\[
\text{SUBSTR}(\text{value, startPosition})
\]
&WORD1 = MYSTRING &MYVAR = 3
SUBSTR(&WORD1,2,4) returns YSTR
SUBSTR(&WORD1,5) returns RING
SUBSTR(&WORD1,&MYVAR,5) returns STRING
SUBSTR(&WORD1,POS(&WORD1,R)) returns RING

**SYMBOL | SYM**

This function returns a value indicating the status of variable name as a valid variable that currently is allocated in the variable pool. This function uses the following syntax:

```
SYMBOL(\text{variableName})
```

**Example**

SYM(variableName)

Returns:

- LIT: \text{variablename} is not recognized as a variable and is treated as a literal
- VAR: \text{variablename} is a valid variable and has a value

**Example**

&MYVAR1 = ABCDEF and &YOURVAR does not exist

SYMBOL(&MYVAR1) returns VAR

SYMBOL(&YOURVAR) returns LIT

In another example, enter this on the Advanced Variable Dependencies panel

SYMBOL(&MYVAR1)=VAR AND SYMBOL(&YOURVAR)=LIT
to code a Rule that fires when &MYVAR1 exists and &YOURVAR does not exist.

**TIME**

This function returns a character string that represents the current time-of-day formatted according to type. This function uses the following syntax:

```
TIME(\text{type})
```

where \text{type} can be one of the following values:

- N: normal; formatted as \textit{HH:MM:SS}
- NU: normal undelimited; formatted as \textit{HHMMSS}
- L: long form; formatted as \textit{HH:MM:SS.THM} (to milliseconds)
- LU: long form undelimited; formatted as \textit{HHMMSSTHM}
- X: extra long form; formatted as $HH:MM:SS.THMIJU$ (to microsecond)
- XU: extra long undelimited; formatted as $HHMMSSTHMIJU$
- Example: \texttt{TIME(X)} returns 08:15:22.354208 (current time of day)

\textbf{UC}

This function returns character string \textit{string1} in all uppercase letters. This function uses the following syntax:

\texttt{UC(string1)}

\textbf{Example}

\texttt{&WORD1 = SystemB}
\texttt{UC(&WORD1) returns SYSTEMB}

\textbf{VALUE}

This function returns the contents of symbol after resolving it as a variable (similar to the REXX VALUE function). This function uses the following syntax:

\texttt{VALUE(symbol)}

\textbf{Example}

\texttt{&VAR1=ASM and &ASM=FAVORITE}
\texttt{VALUE('&VAR1') returns ASM}
\texttt{VALUE(&VAR1) returns FAVORITE}

\textbf{X}

This function changes the specified \textit{string1} of hexadecimal characters into an internal hexadecimal value.

The value \textit{string1} must contain an even number of characters consisting only of numbers 0 through 9 and characters A through F.

This function uses the following syntax:

\texttt{X(string1)}
**Example**  
X(04C327) returns a 3-byte result of x'04C327'

---

**X2C**

This function returns the character string \textit{string1} of hexadecimal characters as its EBCDIC equivalent character string.

An even number of characters must be supplied in \textit{string1}. Note that not all hexadecimal string values will return printable characters.

This function uses the following syntax:

\[
\text{X2C(string1)}
\]

**Example**  
X2C(F1C1C2C3F8F9) returns 1ABC89

---

**X2D**

This function returns the character string \textit{string1} of hexadecimal characters as its decimal equivalent character string.

The maximum is 8 characters (4 hexadecimal bytes) and you must specify an even number of characters.

This function uses the following syntax:

\[
\text{X2D(string1)}
\]

**Example**  
X2D(012C6F27) returns 19689255  
X2D(00000010) returns 16

---

**X2B**

This function returns the character string \textit{string1} of hexadecimal characters into its binary bit character string. This function uses the following syntax:

\[
\text{X2B(string1)}
\]

The maximum is 8 characters (4 hexadecimal bytes) and you must specify an even number of characters.
Example

\(X2B(\text{C1F9})\) returns 1100000111111001

\(X2B(03)\) returns 00000011

Fields that support built-in functions

The following table shows the selection criteria fields for each event type that support the use of built-in Rules functions. Any of the Rules functions are supported.

<table>
<thead>
<tr>
<th>Event Type</th>
<th>Fields that support use of built-in Rules functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALRM</td>
<td>Context, Scope, Queue</td>
</tr>
<tr>
<td>ALRT</td>
<td>Key, Queue</td>
</tr>
<tr>
<td>CICS</td>
<td>Job Name, Acct Info, RACF User, RACF Group, CICS TDQ</td>
</tr>
<tr>
<td>CMD</td>
<td>Job Name, Acct Info, RACF User, RACF Group</td>
</tr>
<tr>
<td>DB2</td>
<td>none</td>
</tr>
<tr>
<td>EOM</td>
<td>Job Name</td>
</tr>
<tr>
<td>EOS</td>
<td>Job Name, RACF User, RACF Group</td>
</tr>
<tr>
<td>EXT</td>
<td>Job Name, Acct Info, RACF User, RACF Group</td>
</tr>
<tr>
<td>IIZ</td>
<td>Job Name, Acct Info, RACF User, RACF Group, Class</td>
</tr>
<tr>
<td>IMP</td>
<td>Job Name, Client, Userld, ROUTE list</td>
</tr>
<tr>
<td>IMS</td>
<td>Job Name, Acct Info, RACF User, RACF Group</td>
</tr>
<tr>
<td>JRNL</td>
<td>Job Name</td>
</tr>
<tr>
<td>MQS</td>
<td>Job Name, Queue Id</td>
</tr>
<tr>
<td>MSG</td>
<td>Job Name, Acct Info, RACF User, RACF Group</td>
</tr>
<tr>
<td>NVC</td>
<td>Job Name</td>
</tr>
<tr>
<td>NVM</td>
<td>Job Name</td>
</tr>
<tr>
<td>TIME</td>
<td>none</td>
</tr>
<tr>
<td>TLM</td>
<td>Job Name, Acct Info, RACF User, RACF Group</td>
</tr>
<tr>
<td>VAR</td>
<td>none</td>
</tr>
</tbody>
</table>
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.

What the Automation Control panel is

The Automation Control panel, shown in the following figure, has four 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 MainView 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
a scrollable list of Rule Sets in the Automation library, including the Rule Set description and the version of MainView AutoOPERATOR that last saved the Rule Set.

Display this portion of the panel by using the EXPAND primary command.

**Figure 98: Automation Control panel**

<table>
<thead>
<tr>
<th>LC</th>
<th>Rule-Set Status</th>
<th>Rules</th>
<th>Rule-Set-Description</th>
<th>Vers</th>
</tr>
</thead>
<tbody>
<tr>
<td>___</td>
<td>----------------</td>
<td>-------</td>
<td>----------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>___</td>
<td>ENABLED</td>
<td>33</td>
<td>0 42,672 04-MAR-11 06:00:40 ALL</td>
<td></td>
</tr>
<tr>
<td>___</td>
<td>ENABLED</td>
<td>88</td>
<td>55 42,672 04-MAR-11 06:00:41 FIRST</td>
<td></td>
</tr>
<tr>
<td>___</td>
<td>ENABLED</td>
<td>5</td>
<td>0 42,672 04-MAR-11 06:00:41 FIRST</td>
<td></td>
</tr>
<tr>
<td>___</td>
<td>ENABLED</td>
<td>44</td>
<td>0 42,672 04-MAR-11 06:00:42 ALL</td>
<td></td>
</tr>
<tr>
<td>___</td>
<td>ENABLED</td>
<td>18</td>
<td>1 42,672 04-MAR-11 06:00:43 FIRST</td>
<td></td>
</tr>
<tr>
<td>___</td>
<td>ENABLED</td>
<td>48</td>
<td>1 42,672 04-MAR-11 06:00:43 FIRST</td>
<td></td>
</tr>
<tr>
<td>___</td>
<td>ENABLED</td>
<td>15</td>
<td>1 42,672 04-MAR-11 06:00:44 FIRST</td>
<td></td>
</tr>
<tr>
<td>___</td>
<td>ENABLED</td>
<td>10</td>
<td>0 42,672 04-MAR-11 06:00:45 FIRST</td>
<td></td>
</tr>
<tr>
<td>___</td>
<td>ENABLED</td>
<td>15</td>
<td>0 42,672 04-MAR-11 06:00:45 FIRST</td>
<td></td>
</tr>
</tbody>
</table>

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 Strategy**
- **manage groups of Rules, called Rule Sets, to provide simpler management of your basic automation**

This function includes using Rule Set filtering that 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 219.

- **see the Rule Set description and the version of MainView AutoOPERATOR that last saved the Rule Set (seen by issuing the EXPAND primary command, as indicated in the following figure).**

**Figure 99: Automation Control panel: EXPAND**

<table>
<thead>
<tr>
<th>LC</th>
<th>Rule-Set Status</th>
<th>Rules</th>
<th>Rule-Set-Description</th>
<th>Vers</th>
</tr>
</thead>
<tbody>
<tr>
<td>___</td>
<td>----------------</td>
<td>-------</td>
<td>----------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>___</td>
<td>ENABLED</td>
<td>33</td>
<td>0 42,672 04-MAR-11 06:00:40 ALL</td>
<td></td>
</tr>
<tr>
<td>___</td>
<td>ENABLED</td>
<td>88</td>
<td>55 42,672 04-MAR-11 06:00:41 FIRST</td>
<td></td>
</tr>
<tr>
<td>___</td>
<td>ENABLED</td>
<td>5</td>
<td>0 42,672 04-MAR-11 06:00:41 FIRST</td>
<td></td>
</tr>
<tr>
<td>___</td>
<td>ENABLED</td>
<td>44</td>
<td>0 42,672 04-MAR-11 06:00:42 ALL</td>
<td></td>
</tr>
<tr>
<td>___</td>
<td>ENABLED</td>
<td>18</td>
<td>1 42,672 04-MAR-11 06:00:43 FIRST</td>
<td></td>
</tr>
<tr>
<td>___</td>
<td>ENABLED</td>
<td>48</td>
<td>1 42,672 04-MAR-11 06:00:43 FIRST</td>
<td></td>
</tr>
<tr>
<td>___</td>
<td>ENABLED</td>
<td>15</td>
<td>1 42,672 04-MAR-11 06:00:44 FIRST</td>
<td></td>
</tr>
<tr>
<td>___</td>
<td>ENABLED</td>
<td>10</td>
<td>0 42,672 04-MAR-11 06:00:45 FIRST</td>
<td></td>
</tr>
<tr>
<td>___</td>
<td>ENABLED</td>
<td>15</td>
<td>0 42,672 04-MAR-11 06:00:45 FIRST</td>
<td></td>
</tr>
</tbody>
</table>
### Setting automation status/strategy (upper portion)

The upper portion of the Automation Control panel controls whether automation is active and, if so, which Rules will be fired.

**Figure 100: Automation Control panel: upper portion**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Automation Control</th>
<th>COMMAND</th>
<th>TGT</th>
<th>MAINVIEW AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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**
- **INDIVIDUAL**

<table>
<thead>
<tr>
<th>LC</th>
<th>Rule-Set Status</th>
<th>Rules</th>
<th>Rule-Set-Description</th>
<th>Vers</th>
</tr>
</thead>
<tbody>
<tr>
<td>___</td>
<td>RULALL</td>
<td>ENABLED</td>
<td>33</td>
<td>MSG EVENTS AND JRNL EVENTS</td>
</tr>
<tr>
<td>___</td>
<td>RULA073</td>
<td>ENABLED</td>
<td>88</td>
<td>AUTOOPERATOR 7.3.00 RULES</td>
</tr>
<tr>
<td>___</td>
<td>RULEXCS</td>
<td>ENABLED</td>
<td>5</td>
<td>MANAGE EXECS AND DIAGNOSE EXEC ISSUES</td>
</tr>
<tr>
<td>___</td>
<td>RULASV</td>
<td>ENABLED</td>
<td>44</td>
<td>TEST ALL ASV FUNCTIONS</td>
</tr>
<tr>
<td>___</td>
<td>RULRPLAY</td>
<td>ENABLED</td>
<td>18</td>
<td>TEST ALL REPLAY FUNCTIONS</td>
</tr>
<tr>
<td>___</td>
<td>RULDAVEO</td>
<td>ENABLED</td>
<td>48</td>
<td>GENERAL RULES FOR MANAGING WORKLOADS</td>
</tr>
<tr>
<td>___</td>
<td>RULDAVEI</td>
<td>ENABLED</td>
<td>15</td>
<td>GENERAL RULES FOR MANAGING IMS</td>
</tr>
<tr>
<td>___</td>
<td>AAORULD1</td>
<td>DISABLED</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>
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 218 for a complete discussion about how to set Automation Strategy.

### Using primary commands

You can enter the following primary commands on the **COMMAND** line of the Automation Control panel.

<table>
<thead>
<tr>
<th>Command</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td>xx</td>
<td>RULxxxxxx</td>
</tr>
<tr>
<td>Command</td>
<td>Parameters</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| 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. |
<table>
<thead>
<tr>
<th>Command</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARMSHOW</td>
<td>[ON</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enter on the <strong>COMMAND</strong> line:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ PARMSHOW OFF: hides the following automation fields from the Automation Control panel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Automation Status     ===&gt; ACTIVE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Automation Strategy   ===&gt; INDIVIDUAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Honor MPF Suppression ===&gt; NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ PARMSHOW ON: shows the automation fields from the Automation Control panel in NOEDIT mode (the specified settings in the Automation fields cannot be modified)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ PARMSHOW: toggles the automation fields display in the following order: OFF---NOEDIT--&gt;EDIT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ PARMSHOW NOEDIT: (Default setting) prevents you from accidentally or unintentionally modifying the automation settings in these fields that affect overall automation by making the fields unmodifiable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ PARMSHOW EDIT: allows you to modify the automation settings</td>
</tr>
<tr>
<td>LOCATE</td>
<td>xxxx</td>
<td>Locates a Rule Set that is listed in the Automation Control Rule Set display. You can specify a full or partial Rule Set name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enter on the <strong>COMMAND</strong> line:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ LOC xxxx: locates a specific Rule Set with the name xxxx</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ L xxx*: locates the first Rule Set name that begins with xxx but might also have additional characters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You can also enter: L, LOC, or LOCATE.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the Rule Set name is not found, the last entry in the Automation Control Rule Set list is displayed.</td>
</tr>
</tbody>
</table>
Describing fields

The following list describes the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automation Status</td>
<td>Indicates whether the Rule Processor itself is ACTIVE or INACTIVE. This field can be modified by typing over the field.</td>
</tr>
<tr>
<td>Automation Strategy</td>
<td>Controls which Rules will be fired when an event meets the selection criteria for more than one Rule. Valid values are FIRST, ALL, and INDIVIDUAL. This field can be modified by typing over the field. To permanently change the Automation Strategy, you must edit BBPARM member AAOPRMxx and enter the automation strategy on the RULESCAN parameter. Rules are organized by Rules Sets and the search order for a Rule depends on the order in which the Rule Sets appear in “What the Automation Control panel is” on page 207. Refer to “Choosing an automation strategy” on page 218 for information about selecting an Automation Strategy.</td>
</tr>
<tr>
<td>Honor MPF Suppression</td>
<td>Provides support for the IBM Message Processing Facility (MPF). Note: This field is supported only if you have the MainView AutoOPERATOR for z/OS 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.</td>
</tr>
</tbody>
</table>

Viewing automation statistics (middle portion)

All of the automation statistics shown in the middle portion of the Automation Control panel are statistics accumulated since the last MainView AutoOPERATOR cold start or since the last time the statistics were reset by using the .RESET STATS command.

**Figure 101: Automation Control panel: middle portion**

<table>
<thead>
<tr>
<th>Automation Statistics</th>
<th>Total Events</th>
<th>Display suppressed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>624</td>
<td>0</td>
</tr>
</tbody>
</table>
Fields in the Automation Statistics portion of the Automation Control panel are described in the following table.

**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 210 for more information.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Events</td>
<td>Is the total number of events that have been presented to the Rule Processor as potential candidates for automation.</td>
</tr>
<tr>
<td>Display Suppressed</td>
<td>Is the total number of times the Rule Processor suppressed the display of an event.</td>
</tr>
<tr>
<td>Events Handled</td>
<td>Is the total number of times a Rule was fired on a matched event.</td>
</tr>
<tr>
<td>Hardcopy Suppressed</td>
<td>Is the total number of times the Rule Processor has suppressed messages destined for SYSLOG or the Journal.</td>
</tr>
<tr>
<td>Current Arrival Rate</td>
<td>Is the rate at which events are presented to the Rule Processor during the last 10 seconds.</td>
</tr>
<tr>
<td>Rule generated Alerts</td>
<td>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.</td>
</tr>
<tr>
<td>Peak arrival rate</td>
<td>Is the highest arrival rate of events passed to the Rule Processor since activation.</td>
</tr>
<tr>
<td>Rule invoked EXECs</td>
<td>Is the total number of times the Rules processor has scheduled an EXEC.</td>
</tr>
</tbody>
</table>

**Grouping Rules into Rule Sets (bottom portion)**

Use the bottom portion of the Automation Control panel to see all the Rule Sets, their status, and how many Rules have been fired.

**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 (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:

<table>
<thead>
<tr>
<th>LC</th>
<th>Rule-Set Status</th>
<th>Rules</th>
<th>Fired</th>
<th>Filtered</th>
<th>Date</th>
<th>Time</th>
<th>Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>___</td>
<td>RULQUAL1 ENABLED</td>
<td>6</td>
<td>46</td>
<td>624d</td>
<td>DD-MM-YY</td>
<td>13:18:50</td>
<td>FIRST</td>
</tr>
<tr>
<td>___</td>
<td>RULJRNL1 ENABLED</td>
<td>108</td>
<td>93</td>
<td>183</td>
<td>DD-MM-YY</td>
<td>13:18:49</td>
<td>FIRST</td>
</tr>
<tr>
<td>___</td>
<td>RULJRNL2 ENABLED</td>
<td>53</td>
<td>6</td>
<td>8</td>
<td>DD-MM-YY</td>
<td>13:18:49</td>
<td>FIRST</td>
</tr>
<tr>
<td>___</td>
<td>RULJRNL3 ENABLED</td>
<td>59</td>
<td>46</td>
<td>32</td>
<td>DD-MM-YY</td>
<td>13:18:49</td>
<td>ALL</td>
</tr>
<tr>
<td>___</td>
<td>RULJRNL4 ENABLED</td>
<td>56</td>
<td>9</td>
<td>17</td>
<td>DD-MM-YY</td>
<td>13:18:49</td>
<td>FIRST</td>
</tr>
<tr>
<td>___</td>
<td>RULJRNL5 ENABLED</td>
<td>37</td>
<td>346</td>
<td>46</td>
<td>DD-MM-YY</td>
<td>13:18:49</td>
<td>ALL</td>
</tr>
<tr>
<td>___</td>
<td>RUL911 ENABLED</td>
<td>6</td>
<td>46</td>
<td>46</td>
<td>DD-MM-YY</td>
<td>13:18:49</td>
<td>FIRST</td>
</tr>
<tr>
<td>___</td>
<td>AAORULCC DISABLED</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>___</td>
<td>AAORULJB DISABLED</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

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 219.

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 “Using the Rule Set Overview panel to manage Rules in Rule Sets” on page 229 for more information about the Rule Set Overview panel.

The fields in the Automation Library portion of the Automation Control panel include

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule-Set</td>
<td>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 MainView AutoOPERATOR use only.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Status</td>
<td>Is the state of the Rule Set. Valid values are ENABLED, DISABLED, TEST, and LOADED.</td>
</tr>
<tr>
<td></td>
<td>■ 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 AAOPRM00 are ENABLED automatically after a cold start.</td>
</tr>
<tr>
<td></td>
<td>■ DISABLED means that the Rules within a Rule Set are not available to process events. Rule Sets that are not specified with the parameter RULESET= in BBPARM member AAOPRM00 are DISABLED after a cold start.</td>
</tr>
<tr>
<td></td>
<td>■ TEST means that the Rules within a Rule Set are fired in test mode, which means that the only action to be performed is to schedule the appropriate EXEC with the variable IMFRLSTA set to TEST.</td>
</tr>
<tr>
<td></td>
<td>■ LOADED is a status that appears on the Automation Control panel when Rules within a Rule Set have been loaded into ECSA storage but the Rules are not enabled and will not fire. The Rules are not written to disk until you SAVE the Rules. Enabling a Rule Set with LOADED status will enable the Rules as they are currently defined in storage and not as they are defined on disk. Disabling a Rule Set with LOADED status purges the Rule Set from storage and all changes that you made to the Rule Set are lost unless you have previous issued a SAVE command.</td>
</tr>
<tr>
<td>Rules</td>
<td>Is the number of Rules contained in the Rule Set.</td>
</tr>
<tr>
<td>Fired</td>
<td>Is the number of times Rules within the Rule Set were fired.</td>
</tr>
<tr>
<td>Filtered</td>
<td>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 219.</td>
</tr>
<tr>
<td></td>
<td>In the space immediately following the Filtered column, is one of three values:</td>
</tr>
<tr>
<td></td>
<td>■ d: Indicates Rule Set filtering is disabled for a Rule Set.</td>
</tr>
<tr>
<td></td>
<td>■ e: Indicates Rule Set filtering is enabled for a Rule Set.</td>
</tr>
<tr>
<td></td>
<td>■ blank: Indicates no Rule Set filtering has been set for a Rule Set.</td>
</tr>
<tr>
<td>Date</td>
<td>Is the date on which the Rule Set was last activated in the format DD-MMM-YY.</td>
</tr>
<tr>
<td>Time</td>
<td>Is the time when the Rule Set was last activated.</td>
</tr>
<tr>
<td>Strategy</td>
<td>Shows a value of: FIRST, 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 218.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Rule Set Description</td>
<td>Shows up to 45 characters of the Rule Set description. This field is displayed when you enter the EXPAND primary command on the COMMAND line.</td>
</tr>
<tr>
<td>Vers</td>
<td>Shows the first 4 characters of the version of MainView AutoOPERATOR that last saved this Rule Set to the BBPARM data set. This field is displayed when you enter the EXPAND primary command on the COMMAND line.</td>
</tr>
</tbody>
</table>

**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 210 for more information.

Use the following line commands in the LC column:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select</td>
<td>S</td>
</tr>
<tr>
<td>Enable</td>
<td>E</td>
</tr>
<tr>
<td>Disable</td>
<td>D</td>
</tr>
<tr>
<td>Test</td>
<td>T</td>
</tr>
<tr>
<td>SAve</td>
<td>SA</td>
</tr>
<tr>
<td>Move</td>
<td>M</td>
</tr>
<tr>
<td>Before</td>
<td>B</td>
</tr>
<tr>
<td>After</td>
<td>A</td>
</tr>
</tbody>
</table>
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:

- **ALL**
- **FIRST**
- **INDIVIDUAL**

If you choose an automation strategy of **ALL** or **FIRST**, all Rule Sets are searched for Rules to match a certain event. You can specify **ALL** or **FIRST** 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 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 Rule can fire to handle a single event.

**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 first.
Selecting INDIVIDUAL

If you select the INDIVIDUAL automation strategy, you can specify that each Rule Set has its own strategy of FIRST or ALL. 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.

---

**Note**

When you are creating Rules for event type MSG for multiline WTOs (MLWTOs) by specifying `MLWTO Minor = ALL` or `MLWTO Minor = SINGLE`, the strategy of ALL and FIRST works differently.

For these Rules, even when you specify FIRST, multiple Rules might fire because when the Rule Processor searches for Rules that match selection criteria you have specified, the Rule Sets are scanned for a Rule for a single WTO (where no values is specified in the `MLWTO Minor` field), a MLWTO Rule with `MLWTO Minor = ALL` specified, and a MLWTO Rule with `MLWTO Minor = SINGLE` specified.

When you add Rules for MLWTO messages, you should try to ensure that there might be more than one Rule that matches the selection criteria you choose. In addition, if you place a Rule with `MLWTO Minor = ALL` specified before a Rule for the same message with `MLWTO Minor = SINGLE` does not prevent the second Rule from firing.

---

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 the following figure, note the column Filtered. This value represents the number of events that the Rule Set was searched for matching Rules.

![Figure 103: Automation Control panel: Rule Set filtering](image)

<table>
<thead>
<tr>
<th>Rule Set Status</th>
<th>Rules</th>
<th>Fired</th>
<th>Filtered</th>
<th>Date</th>
<th>Time</th>
<th>Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>RULQUAL1 ENABLED</td>
<td>6</td>
<td>46</td>
<td>624d</td>
<td>DD-MM-YY 13:18:50</td>
<td>FIRST</td>
<td></td>
</tr>
<tr>
<td>RULJRNL1 ENABLED</td>
<td>108</td>
<td>93</td>
<td>183</td>
<td>DD-MM-YY 13:18:49</td>
<td>FIRST</td>
<td></td>
</tr>
<tr>
<td>RULJRNL2 ENABLED</td>
<td>53</td>
<td>6</td>
<td>8</td>
<td>DD-MM-YY 13:18:49</td>
<td>FIRST</td>
<td></td>
</tr>
<tr>
<td>RULJRNL3 ENABLED</td>
<td>59</td>
<td>46</td>
<td>32</td>
<td>DD-MM-YY 13:18:49</td>
<td>ALL</td>
<td></td>
</tr>
<tr>
<td>RULJRNL4 ENABLED</td>
<td>56</td>
<td>9</td>
<td>17</td>
<td>DD-MM-YY 13:18:49</td>
<td>FIRST</td>
<td></td>
</tr>
<tr>
<td>RULJRNL5 ENABLED</td>
<td>37</td>
<td>346</td>
<td>46</td>
<td>DD-MM-YY 13:18:49</td>
<td>ALL</td>
<td></td>
</tr>
<tr>
<td>AAORULCC DISABLED</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>AAORULJB DISABLED</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Therefore, the information for Rule Set RULJRNL1 looks like the following panel:

![Automation Statistics](image)

<table>
<thead>
<tr>
<th>Rule Set Status</th>
<th>Rules</th>
<th>Fired</th>
<th>Filtered</th>
<th>Date</th>
<th>Time</th>
<th>Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>RULQUAL1 ENABLED</td>
<td>6</td>
<td>46</td>
<td>624d</td>
<td>DD-MM-YY 13:18:50</td>
<td>FIRST</td>
<td></td>
</tr>
<tr>
<td>RULJRNL1 ENABLED</td>
<td>108</td>
<td>93</td>
<td>183</td>
<td>DD-MM-YY 13:18:49</td>
<td>FIRST</td>
<td></td>
</tr>
</tbody>
</table>

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 filtering and Rule Set match rate for Rule Sets

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. From this panel, after making the filter changes, the
  Confirm Rule Set Filter Modification panel is displayed and you can specify that
  you want to save, or not save, the Rule Set filter changes that you made. The
  following figure shows an example of the panel.

- 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 (RULJ911) with filtering enabled
lessens the number of events seen by the Rule Set and makes automation with Rules
more efficient.

To enable filtering

1. From the Automation Control panel, type F in the LC column next to the Rule Set.
   (Figure 105 on page 221)
Figure 106 on page 222 is displayed.

**Figure 106**: Filter Criteria panel: viewing the filter Criteria

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 specifies that all events with a Text-ID that begins with IM911 and are JRNL events will be processed by the Rules in this Rule Set.

The following table describes the fields on this panel:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text ID</td>
<td>Fill in the <strong>Text-ID</strong> of the event that you want to be seen by this Rule Set. This maximum length of the <strong>Text-ID</strong> is 16 characters. To have the Rule Set select events with more information, use the <strong>Text String</strong> field.</td>
</tr>
<tr>
<td>Text String</td>
<td>Specify message/event text to be matched in addition to the Text-ID. The maximum length of the Text-ID is 75 characters.</td>
</tr>
<tr>
<td>Ignore leading plus</td>
<td>Specify YES so that the filter criteria for this Rule Set ignores a leading plus sign (+) in the text-IDs of write-to-operator messages (WTOs) and allows the Rules within this Rule Set to be search for matching Rules.</td>
</tr>
<tr>
<td>Field name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| Event Type | Specify the event type for the events that will be seen by the Rule Set. You can specify ALRT, ALRM, BRC, CICS, CMD, DB2, EOM, EOS, EXT, HWTO, IMP, IMS, JRNL, MQS, MSG, NVC, NVM, TIME, TLM, VAR. For the MSG event type,  
  ■ 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. |
in the Rule Processor” on page 160 for more information about using variable dependencies for selection criteria.

3 Press Enter; Figure 108 on page 224 is displayed.

**Figure 108: Rule Set default values panel**

Use this panel to define the default Criteria match rate to be used for each Rule in this Rule Set that does not have a value specified.

---

**Note**

Refer to “Using the Rule Processor Detail Control panel” on page 63 for details on Criteria match rates within an individual Rule.

You can also specify a default setting for the Rule Set where the all the Rules for MSG events will ignore the leading plus sign (+) that occurs in message text unless the individual Rule has specified NO. Refer to “Selection Criteria fields” on page 279 where the Ignore leading plus field for individual Rules is documented for more information.

The following table describes the fields on the Rule Set default values panel:
### Table 23: Default Rule Set settings: Criteria match rate and ignore leading plus

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>If matched</td>
<td>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 <strong>in seconds</strong> and <strong>then status</strong> 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.</td>
</tr>
<tr>
<td>in seconds</td>
<td>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 <strong>If matched</strong> field within the time interval set in the <strong>in seconds</strong> field, the action specified in the <strong>then status</strong> field is taken.</td>
</tr>
<tr>
<td>then status</td>
<td>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 as follows: <strong>SUSPEND</strong>: 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. <strong>DISABLE</strong>: 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. <strong>NOACTION</strong>: 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.</td>
</tr>
</tbody>
</table>
## Default ignore leading plus value for MSG rules

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Default ignore leading plus value for MSG rules</strong></td>
<td>Specify <strong>YES</strong> so that every MSG and HWTO initiated Rule in this Rule Set, by default, ignores the leading plus sign (+) that often precedes write-to-operator (WTO) messages. The default is <strong>NO</strong>. This setting affects only MSG and HWTO initiated Rules in this Rule Set that do not have a value specified for the <strong>Ignore leading plus</strong> selection criteria field. Refer to the description of the <strong>Ignore leading plus</strong> field in “Selection Criteria fields” on page 279 for more information about how the setting in this field affects MSG and HWTO initiated Rules.</td>
</tr>
</tbody>
</table>

4. **Press PF3** to return to the Automation Control Panel.

**Note**

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.

5. *(optional)* To view the Rules, from the Automation Control panel, type **S** in the **LC** column next to the Rule Set as shown in (Figure 109 on page 226.

### Figure 109: Automation Control panel: viewing a Rule Set’s Rules

<table>
<thead>
<tr>
<th>Status</th>
<th>Automation Status</th>
<th>Automation Strategy</th>
<th>Honor MPF Suppression</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Active</strong></td>
<td><strong>INDIVIDUAL</strong></td>
<td><strong>NO</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total Events</th>
<th>Events Handled</th>
<th>Current arrival rate</th>
<th>Peak arrival rate</th>
<th>Rule-Set Status</th>
<th>Rules</th>
<th>Fired</th>
<th>Filtered</th>
<th>Filtered</th>
<th>Date</th>
<th>Time</th>
<th>Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>624</td>
<td>49</td>
<td>3 / sec</td>
<td>10 / sec</td>
<td>ENABLED</td>
<td>6</td>
<td>46</td>
<td></td>
<td></td>
<td>DD-MM-YY 13:18:50</td>
<td>FIRST</td>
<td></td>
</tr>
<tr>
<td>RULQUAL1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DD-MM-YY 13:18:49</td>
<td>FIRST</td>
<td></td>
</tr>
<tr>
<td>RULJRN1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DD-MM-YY 13:18:49</td>
<td>FIRST</td>
<td></td>
</tr>
<tr>
<td>RULJRN2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DD-MM-YY 13:18:49</td>
<td>FIRST</td>
<td></td>
</tr>
<tr>
<td>RULJRN3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DD-MM-YY 13:18:49</td>
<td>ALL</td>
<td></td>
</tr>
<tr>
<td>RULJRN4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DD-MM-YY 13:18:49</td>
<td>FIRST</td>
<td></td>
</tr>
<tr>
<td>RULJRN5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DD-MM-YY 13:18:49</td>
<td>ALL</td>
<td></td>
</tr>
<tr>
<td>RULJ911</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DD-MM-YY 13:18:49</td>
<td>FIRST</td>
<td></td>
</tr>
<tr>
<td>AAORULCC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DD-MM-YY 13:18:49</td>
<td>ALL</td>
<td></td>
</tr>
<tr>
<td>AAORULJB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DD-MM-YY 13:18:49</td>
<td>FIRST</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 109 on page 226 is displayed.**

### Figure 110: Rule Set Overview panel: viewing a Rule Set’s Rules

- Rule Set ID: **RULJ911**
- Ruleset Strategy: **FIRST**
- Date: **YY/MM/DD**

**Figure 110 on page 226 is displayed.**
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.

6 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 on the COMMAND line from the Rule Set Overview or enter SAVE when prompted while exiting the Rule Set.
Chapter 9  Using the Rule Set Overview panel to manage Rules in Rule Sets

Using the Rule Set Overview panel to manage Rules in Rule Sets

This chapter explains how to copy or move Rules within a single Rule Set and between Rule Sets.

What the Rule Set Overview panel is

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 219.

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 111 on page 229 shows the Rule Set Overview panel when you access it from the Automation Control panel.

Figure 111: Rule Set Overview panel—view 1

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Rule Set Overview</th>
<th>AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Set ID: AAORUL00</td>
<td>Ruleset Strategy: FIRST</td>
<td>TGT: JO81</td>
</tr>
<tr>
<td>Primary commands: Add, Save, Sort, Unsort, Reset, Filter</td>
<td>DATE: YY/MM/DD</td>
<td>TIME: 14:14:49</td>
</tr>
<tr>
<td>LC CMDS: (S)elect, (BR)owse, (E)nable, (D)isable, (T)est, (DE)lete, (I)nsert</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(C)opy/(C)copy, (M)ove/(M)ove, (B)efore or (A)fter, (R)epeat, GO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rule Set Desc:</td>
<td>Scroll right/left</td>
<td></td>
</tr>
<tr>
<td>LC Rule-id Stat Text-id</td>
<td>Type</td>
<td>Fired EXEC</td>
</tr>
</tbody>
</table>
Scroll left to see the **Group**, **Function**, **Code**, **Author**, and **Description** columns as shown in Figure 112 on page 230. The fields on this view of the panel are as follows.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule-id</td>
<td>is the 1- to 8-character unique ID assigned to this Rule at its creation</td>
</tr>
<tr>
<td>Stat</td>
<td>is the Rule status, which can be one of the following: ENA, DIS, TST, or SUS</td>
</tr>
<tr>
<td>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 “Criteria match rate thresholds” on page 78.</td>
<td></td>
</tr>
<tr>
<td>Text-id</td>
<td>is the 1- to 16-character first word of the message or command defined for this Rule</td>
</tr>
<tr>
<td>Type</td>
<td>is the origin of the event, which can be one of the following types: ALRT, ALRM, CICS, CICa, CMD, DB2, EXT, HWTO, IMP, IMS, IMSa, IMSs, JRNL, MQS, MSG, MSGa, MSGs, NVC, NVM, TIME, or VAR</td>
</tr>
<tr>
<td>Fired</td>
<td>is the number of times the Rule was triggered</td>
</tr>
<tr>
<td>EXEC</td>
<td>is the name of the EXEC that will be scheduled if the Rule fires</td>
</tr>
<tr>
<td>If the EXEC name specified in the Rule is a variable, its length may be more than 8 charact</td>
<td></td>
</tr>
<tr>
<td>Changed</td>
<td>is the date (YY/MM/DD) and time (HH:MM:SS) the Rule was last modified</td>
</tr>
<tr>
<td>This data is stored in the Rule.</td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>is the user ID that made the most recent change to this Rule</td>
</tr>
<tr>
<td>This data is stored in the Rule.</td>
<td></td>
</tr>
</tbody>
</table>

Scrolling *left* shows the **Group**, **Function**, **Code**, **Author**, and **Description** fields (Figure 112 on page 230).

**Figure 112: Rule Set Overview panel—view 2**

<table>
<thead>
<tr>
<th>Rule Set ID: AAORULST</th>
<th>Ruleset Strategy ===FIRST===</th>
<th>DATE --- YY/MM/DD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary commands: Add, Save, Sort, Unsort, Reset, Filter</td>
<td>TIME --- 14:21:30</td>
<td></td>
</tr>
<tr>
<td>LC CMDS --- (S)elect, (BR)owse, (E)nable, (D)isable, (T)est, (DE)lete, (I)nsert</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(C)opy/(CC)opy, (M)ove/(MM)ove, (B)efore or (A)fter, (R)epeat, GO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rule Set Desc.</td>
<td>Sort Criterion: Filter ENABLED</td>
<td>Scroll right/left</td>
</tr>
<tr>
<td>LC Rule-id Status</td>
<td>Group</td>
<td>Function</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>---------</td>
</tr>
<tr>
<td>IS0834I DIS</td>
<td>STRTERM</td>
<td>DB2START</td>
</tr>
<tr>
<td>IS0836I DIS</td>
<td>STRTERM</td>
<td>DB2TERM</td>
</tr>
<tr>
<td>DFS9940I DIS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IM0001I DIS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMSCHCK ENA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RUL00001 ENA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The following table describes the fields on this view of the panel.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule-id</td>
<td>is the one- to eight-character unique ID assigned to this Rule at its creation</td>
</tr>
<tr>
<td>Stat</td>
<td>is the Rule status, which can be one of the following statuses: 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 “Criteria match rate thresholds” on page 78.</td>
</tr>
<tr>
<td>Group</td>
<td>is the one- to eight-character group name defined for this Rule</td>
</tr>
<tr>
<td>Function</td>
<td>is the one- to eight-character function name defined for this Rule</td>
</tr>
<tr>
<td>Code</td>
<td>is the two-character code ID defined for this Rule</td>
</tr>
<tr>
<td>Author</td>
<td>is the one- to eight-character name of the author defined for this Rule</td>
</tr>
<tr>
<td>Description</td>
<td>is the description defined for this Rule. This field can be up to 40 characters but on the Rule Set Overview panel, the display is limited to show only the first 25 characters of the description. The Rule Processor Detail Control panel shows the full 40 character description.</td>
</tr>
</tbody>
</table>

Scroll right once to display the Text-ID and Text-String fields as shown in Figure 113 on page 231.

**Figure 113: Rule Set Overview panel—view 3**

| Rule Set ID: AAORULST Ruleset Strategy ===> FIRST DATE --- YY/MM/DD |
| Primary commands: Add, Save, Sort, Unsort, Reset, Filter TIME --- 14:21:30 |
| LC CMDs --- (S)elect, (BR)owse, (E)nable, (D)isable, (T)est, (DE)lete, (I)nteract. (C)opy/(CC)opy, (M)ove/(MM)move, (B)efore or (A)fter, (R)epeat, GO |
| Rule Set Desc.: Filter ENABLED Scroll right/left |
| LC Rule-id Text-ID Text-String |
| __ IS0834I |
| __ IS0836I1 |
| __ DFS9940 |
| __ IM0001I |
| __ IMSCHCK |
| __ RUL00001 |
| __ IS0861I |

The following list describes the fields on this view of the panel.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule-id</td>
<td>is the 1- to 8-character unique ID assigned to this Rule at its creation</td>
</tr>
<tr>
<td>Text-ID</td>
<td>is the 1- to 16-character first word of the message or command defined for this Rule</td>
</tr>
</tbody>
</table>
For information about **Text-ID** and **Text-String**, see “Enabling filtering and Rule Set match rate for Rule Sets” on page 221.

Scroll *right* once displays **Stat**, **Start Time**, **Interval**, **Stop Time**, **Stop Count**, and **Fired** fields (Figure 116).

**Figure 114: Rule Set Overview panel—view 4**

---

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text-String</td>
<td>is the text string defined for this Rule; can be up to 26 characters and includes the Text ID</td>
</tr>
</tbody>
</table>

The fields on this view of the panel are as follows:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule-id</td>
<td>is the one- to eight-character unique ID assigned to this Rule at its creation</td>
</tr>
<tr>
<td>Stat</td>
<td>is the Rule status, which can be one of the following values: ENA, DIS, TST, or SUS</td>
</tr>
<tr>
<td></td>
<td>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 “Criteria match rate thresholds” on page 78.</td>
</tr>
<tr>
<td>Start Time</td>
<td>displays (in hours, minutes, and seconds) at what clock time a time-initiated Rule begins firing</td>
</tr>
<tr>
<td></td>
<td>This field contains data only for Rules whose Event type is TIME.</td>
</tr>
<tr>
<td>Interval</td>
<td>displays (in hours, minutes, and seconds) how often the Rule fires</td>
</tr>
<tr>
<td></td>
<td>For example, an interval of 00:05:00 means the Rule will fire at the Start Time and every five minutes thereafter until the Stop Time or Stop Count value is reached. The minimum interval is five seconds.</td>
</tr>
<tr>
<td>Stop Time</td>
<td>displays (in hours, minutes, and seconds) at what clock time the Rule should stop firing</td>
</tr>
<tr>
<td>Stop Count</td>
<td>specifies the number of times the Rule is to be fired from the Start Time</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>Fired</td>
<td>shows how many times the Rule was fired</td>
</tr>
</tbody>
</table>

Scroll right once displays **Job Name**, **Job Type**, **Job Class**, **RACF User-ID**, and **RACF Group** fields (Figure 115 on page 233).

**Figure 115: Rule Set Overview panel—view 5**

The following list describes the fields on this view of the panel.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule-id</td>
<td>is the one- to eight-character unique ID assigned to this Rule at its creation</td>
</tr>
<tr>
<td>Job Name</td>
<td>is the name of the originating batch job, TSO user ID, or Started Task defined for this Rule Blank means any.</td>
</tr>
<tr>
<td>Job Type</td>
<td>is the type of originator defined for this Rule: JOB, STC, or TSO</td>
</tr>
<tr>
<td>Job Class</td>
<td>is the job class of the job defined for this Rule</td>
</tr>
<tr>
<td>RACF User-ID</td>
<td>is the user ID defined for this Rule, if RACF or CA-TOPSECRET or ACF2 is installed</td>
</tr>
<tr>
<td>RACF GROUP</td>
<td>is the name of the RACF Group defined for this Rule</td>
</tr>
</tbody>
</table>

Scrolling right once displays **Accounting Information**, **Con Name**, **Route Code**, and **Desc Codes** fields (Figure 116 on page 233).

**Figure 116: Rule Set Overview panel—view 6**
The following list describes the fields on this view of the panel.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounting Information</td>
<td>is the one- to eight-character unique ID assigned to this Rule at its creation</td>
</tr>
<tr>
<td>Con Name</td>
<td>is the accounting entered on the Account Info field of the Selection Criteria panel</td>
</tr>
<tr>
<td>Rout Code</td>
<td>is the MVS console name for messages or commands</td>
</tr>
<tr>
<td>Job Class</td>
<td>is the MVS route code associated with the WTO or WTOR</td>
</tr>
<tr>
<td>Desc Code</td>
<td>is the MVS descriptor code associated with the WTO or WTOR</td>
</tr>
</tbody>
</table>

Scrolling right once displays BiiZ Send and BiiZ Class fields (Figure 117 on page 234).

**Figure 117: Rule Set Overview panel—view 7**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Rule Set Overview</th>
<th>AutoOPERATOR COMMAND</th>
<th>TGT: DLAO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Set ID: AAORUL00</td>
<td>Ruleset Strategy: FIRST</td>
<td>DATE: YY/MM/DD</td>
<td>TIME: 12:04:19</td>
</tr>
<tr>
<td>Primary commands: Add, Save, Sort, Unsort, Filter</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LC CMDS: (S)elect, (BR)owse, (E)nable, (D)isable, (T)est, (DE)lete, (I)ntersect, (C)opy/(CC)opy, (M)ove/(MM)ove, (B)efore or (A)fter, (R)epeat, GO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rule Set Desc.:</td>
<td>Scroll left</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BiiZ</td>
<td>BiiZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LC Rule-id</td>
<td>Send Class</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The following list describes the fields on this view of the panel.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule-id</td>
<td>is the one- to eight-character unique ID assigned to this Rule at its creation</td>
</tr>
<tr>
<td>BiiZ Send</td>
<td>shows whether events are to be sent to the BMC Impact Manager cell</td>
</tr>
<tr>
<td>BiiZ Send</td>
<td>shows the class name mask used to select Rules</td>
</tr>
</tbody>
</table>
# Using primary commands

You can enter the following primary commands on the **COMMAND** line of the Rule Set Overview panel.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td>displays the Rule Processor Detail Control panel</td>
</tr>
</tbody>
</table>
| 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 in which case the default is Descending | D. |
| UNSORT  | reverses 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 | I can be issued. |
| RESET   | enables you to cancel any pending move or copy line commands that you might have |
| Filter  | enables 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 219. |

# Using line commands

You can use the following line commands in the **LC** column.
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select</td>
<td>S</td>
</tr>
<tr>
<td>Browse</td>
<td>BR</td>
</tr>
<tr>
<td>Enable</td>
<td>E</td>
</tr>
<tr>
<td>Disable</td>
<td>D</td>
</tr>
<tr>
<td>Test</td>
<td>T</td>
</tr>
<tr>
<td>DElete</td>
<td>DE</td>
</tr>
<tr>
<td>Insert</td>
<td>I</td>
</tr>
<tr>
<td>Copy</td>
<td>C</td>
</tr>
<tr>
<td>Move</td>
<td>M</td>
</tr>
<tr>
<td>Before</td>
<td>B</td>
</tr>
<tr>
<td>After</td>
<td>A</td>
</tr>
<tr>
<td>Repeat</td>
<td>R</td>
</tr>
<tr>
<td>GO</td>
<td>starts (or restarts) a TIME-initiated Rule that has ATSTART specified as its start time. Note: The Rule must have a status of ENA or SUS. Use the GO line command when you need to immediately start an existing ATSTART TIME Rule and you cannot restart the BBI-SS PAS. You can also use GO to start a new ATSTART TIME Rule.</td>
</tr>
</tbody>
</table>

What the Rule Set Overview panel is
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.

When you are trying to duplicate a Rule within the same Rule Set, you must use the Repeat | R command. The Copy command will not work.

Copying or moving Rules within Rule Set

To use the copy or move commands within the same Rule Set, follow these steps:

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 Rules in the LC column.
4. Enter either the After | A or the Before | B line commands depending on where you want to copy or move the Rules 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 Rules 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 Rules from the first Rule Set to this Rule Set.

7. This Rule Set must have at least one Rule already in it where you can enter the After | A or Before | B line commands.

8. Press Enter.

The Rules are copied/moved to the new location.

9. Press PF3/END.

The Confirm Rule Set Modifications panel is displayed:

10. 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.

   - 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.

Deleting Rule Sets

After you have issued a SAVE command for a Rule, it is saved to the Rule Set that it was defined in and the Rule and the Rule Set are saved to common storage.
The saved Rule Set is a member of a partitioned data set (PDS) as the first data set of the BBIPARM DDNAME concatenation for the MainView AutoOPERATOR BBI-SS PAS. The Rule Set is saved with a member name that matches the Rule Set name that was used in the Rules Processor panels. The Rules Processor application does not provide an online facility that allows you to delete a Rule Set, even if a Rule Set does not contain any Rules.

If you have never issued the SAVE command for a Rule (or Rules) in a Rule Set, if you disable the Rule Set, it deletes the Rule Set and its contents are not retrievable.

After the Rules and Rule Set have been saved, the only way to delete a saved Rule Set is to disable the Rule Set from the Rules Processor application, edit the data set in ISPF (or another product) and delete the member (Rule Set).

Before you delete the Rule Set, you should ensure that other MainView AutoOPERATOR BBI-SS PASs are not sharing the data set. Also, note that when you delete a Rule Set (PDS member), and the Rule Set is enabled on any BBI-SS PAS, you are not deleting the Rules from the other BBI-SS PAS because the Rules are saved in common storage and continue to exist on that LPAR.

If you accidentally delete a Rule Set that was enabled in another MainView AutoOPERATOR BBI-SS PAS, you can copy the Rule Set to a MainView AutoOPERATOR BBI-SS PAS, enter the SAVE command, and restore the Rule Set.
This chapter describes terms and concepts you should be familiar with for implementing Rules Management. It also includes information about how to implement Rules Management for MainView AutoOPERATOR.

Benefits of using a Rules registry

Using a Rules registry with the Rules Management application allows you to have better control over Rules and eases administration tasks for implementing Rules throughout many LPARs and sysplexes.

The Rules registry is a Runtime Component System (RTCS) registry that is used as a repository for Rules-specific information.

A single registry can be used by a single MainView AutoOPERATOR PAS or, it can be shared by multiple MainView AutoOPERATOR PASes, which is also called an AOPLex.

For more information about RTCS, see the BMC Runtime Component System Configuration and Administration Guide.

Rules Management editor

The Rules Management application provides a windows-based editor that you can use to create, edit, and deploy Rules throughout your sysplex.

The Rules Management editor is easy to use and is also compatible with MainView Explorer. This means that you can also perform all of the Rules Management functions through the web.
Rules

With the Rules Management application, Rules are no longer stored by Rule Set where they are members of a data set as they are with classic Rules.

With Rules Management, you can associate a Rule with more than one Rule Set. This eases maintenance by eliminating unnecessary duplication.

Rules are now grouped into Rule pools that you can set up in any grouping that makes sense for your enterprise. This could be by geographical location, by application, by responsible department, by sysplex, or by anything else.

The same is true for Rule Sets. Rule Sets are now organized into setbases that can, like a Rule pool, represent any organization model. A Rule Set can be defined as a generic Rule Set when the Rule Set does not have individual Rules as members but instead, identifies an entire Rule pool as automatic members of the Rule Set. You can add new Rules to the Rule pool and deploy the Rules without having to add the Rule to a Rule Set. Because the Rule is a member of a Rule pool, it is automatically a member of any generic Rule Set that uses that Rule Pool.

In MainView AutoOPERATOR version 8.1 (and later), the windows mode Rules editor might contain additional or extended functionality that is not offered in the full-screen Rules editor. One example is that in MainView AutoOPERATOR version 8.1 you can use a variable name as the value of the DELAY( ) keyword. This makes it possible to set up standard delay values using variables and, alter those delays by changing the variable’s value instead of having to edit all of the Rules individually.

Deployment and AOplexes

You can define as many AOplexes in your system as you need. An AOplex might be one MainView AutoOPERATOR PAS or it might contain many PASes.

Regardless of the level of registry sharing in an AOplex, you have the ability to deploy Rules and Rule Sets to remote registries. This ability means that several PASes can share a single registry image and, at the same time, the registry image can also be used to house Rules and Rule Sets for other MainView AutoOPERATOR PASes that do not share the registry image.

The following is an example of how you might administer Rules across multiple sysplexes.
The level of registry sharing and how centralized you want to make your Rules management is up to you. With deployment, it can be as centralized or as decentralized as your needs dictate.

Note
An AOPlex cannot span across sysplex boundaries. This is a limitation of using the RTCS registry. However, if your CASes are connected by way of TCP/IP, you can access, edit, add, and deploy Rules on another sysplex. For more information about the AOPlex, refer to the MainView AutoOPERATOR Customization Guide.

Accessing the Rules Management application

You can access the Rules Management application in two different ways.

To access the Rules Management application from full-screen MainView AutoOPERATOR

1 Display the Automation Menu.

2 Select R for Rules Management via Rules Registry (Windows Mode) and press Enter.

Figure 118: Accessing the Rules Management application: full-screen mode
The following steps describe a second way that you can access the Rules Management application.

3 Display the AutoOPERATOR EZ Menu in windows mode.

4 Select Rules Management from the Automation Support list of options.

**Figure 119: EZAO menu**

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>SCROLL</th>
<th>CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURR WIN</td>
<td>ALT WIN</td>
<td></td>
</tr>
<tr>
<td>&gt;W1 =EZAO</td>
<td>ENTER</td>
<td></td>
</tr>
</tbody>
</table>

**Note**

The Rules Management option will be greyed out if the MainView AutoOPERATOR PAS is not using the Rules Registry.

5 The EZAORUL view is displayed.

**Figure 120: Rules Management application menu: EZAORUL**
Creating a PAS definition

A PAS definition (PASDef) specifies what Rule Sets a MainView AutoOPERATOR PAS will use and how the Rules Sets are installed (enabled, disabled, and so on). The PASDef also identifies which PASes will use the PASDef.

You can specify as many MainView AutoOPERATOR PASes to share a single PASDef as you like. So for example, if you need one PAS on each of 5 different LPARs and you want all the PASes to have the same Rules and Rule Sets, you can create one PASDef record and tell it the names of the 5 PASes that will use it.

To create a PAS Definition

1. Select the option PAS Definitions from the EZAORUL menu.

   **Figure 121: EZAORUL menu**

   The PASDEF view is displayed. For information about this view, place the cursor on the view name and press PF1.

2. Issue the ADD command in the COMMAND field to start creating a new PAS definition.

   **Figure 122: Adding a new PAS definition**

   The PASDEF view is displayed. For information about this view, place the cursor on the view name and press PF1.
The ADD PASDef dialog is displayed.

3 Enter values for the required fields on the ADD PASDef dialog.

**Figure 123: ADD PASDef dialog**

<table>
<thead>
<tr>
<th>COMMAND ====&gt;</th>
<th>ADD PASDef</th>
<th>SCROLL ====&gt; CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>New PASDef Information:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name.:</td>
<td>(Required)</td>
<td></td>
</tr>
<tr>
<td>Group:</td>
<td>BBPLEX01</td>
<td>(Required)</td>
</tr>
<tr>
<td>Desc.:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automation Strategy:</td>
<td>S Individual</td>
<td>First</td>
</tr>
<tr>
<td>Automation Status..:</td>
<td>S Active</td>
<td>Inactive</td>
</tr>
<tr>
<td>Honor MPF..........:</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Add Rsn:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

END to save the new PASDef
CANCEL to exit without saving
HELP to see dialog help

For more information about any of the required fields, place the cursor on the field and press **PF1**.

4 After filling the required information on the dialog, enter END to save the new PAS Definition

### Creating a valid PAS list

After creating the PASDef, you must associate the MainView AutoOPERATOR PAS or PASes with a PASDef.

**To associate a MainView AutoOPERATOR PAS with a PASDef**

1 Issue the VP line command next to the PAS definition on the PASDEF view.

**Figure 124: PASDEF view**

<table>
<thead>
<tr>
<th>CMD PASDef Name</th>
<th>Group</th>
<th>Description</th>
<th>Val PAS Ruleset</th>
<th>Ruleset Automatn Honor</th>
</tr>
</thead>
<tbody>
<tr>
<td>VP AAON81.PAS.Definition</td>
<td>BBPLEX01</td>
<td>Created by import</td>
<td>1</td>
<td>1 All</td>
</tr>
</tbody>
</table>

**Creating a valid PAS list**

246 *MainView AutoOPERATOR Basic Automation Guide Volume 1: Using Rules*
The VPL view is displayed. For information about this view, place the cursor on the view name and press PF1.

2 Issue the ADD command in the COMMAND field, to start creating a new valid PAS list (VPL) definition.

Figure 125: ADD VPL view

The ADD Valid PAS dialog is displayed.

3 Enter values for the required fields on the ADD PASDef dialog.

Figure 126: Add Valid PAS dialog

For more information about any of the required fields, place the cursor on the field and press PF1.

4 After filling the required information on the dialog, enter END to save the new VPL Definition

The MainView AutoOPERATOR PAS specified in the VPL will now use the selected PAS definition. The PAS definition identifies the list of Rule Sets that will be loaded into the PAS during a COLD start of the PAS.
Creating a setbase

A setbase is a collection of Rule Sets. You can decide the purpose of the collection. The collection of Rule Sets might be for a single application, a single geographical location, or a business unit of your company.

The setbase can represent anything that makes sense in your environment. You might decide that you do not have a need for a setbase. If you do not need a setbase, place all of your Rule Sets into the supplied DEFAULT setbase. You can always implement additional setbases at a later time if you decide you need to use them. Organizing your Rule Sets into setbases will not prevent or restrict how you use those Rule Sets in a PAS.

To create a setbase

1. Select Setbases from the EZAORUL view.

   **Figure 127: EZAORUL view**

   The SETBS view is displayed. For information about this view, place the cursor on the view name and press PF1.

   **Figure 128: SETBS view**

   The ADD Setbase Definition dialog is displayed.

2. Issue the ADD command in the COMMAND field to start creating a new setbase.
3 Enter values for the required fields on the ADD Setbase Definition dialog.

**Figure 129: ADD Setbase Definition dialog**

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>SCROLL</th>
<th>CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD Setbase Definition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Setbase Information:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name.:</td>
<td>(required)</td>
<td></td>
</tr>
<tr>
<td>Group:</td>
<td>(required)</td>
<td></td>
</tr>
<tr>
<td>Desc:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>END to save the setbase</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CANCEL to abandon the request</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HELP to see help for the dialog</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For more information about any of the required fields, place the cursor on the field and press **PF1**.

4 After filling the required information on the dialog, enter END to save the new setbase definition.

You can use the SETBS view to select which setbase is the default setbase. The view will remember the last setbase you accessed and use it when you launch the RULESET view the next time.

You can proceed to creating Rule Sets.

---

**Creating a Rule Set**

A Rule Set contains names of Rules and their sequence. Including a Rule Set in a PASDef means that the Rule Set is loaded into the PAS (or PASes) that use the PASDef during a COLD start of the PAS.

The Rules are not physically stored in a Rule Set. Rules can be included in more than one Rule Set so that you have only one copy of the Rule that you need to maintain.

**To create a Rule Set**

1 Select Rulesets from the EZAORUL view.

**Figure 130: EZAORUL view**

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>SCROLL</th>
<th>CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>ddmmyyyy ------ MAINVIEW WINDOW INTERFACE (V6.1.01) --------</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Chapter 10  Rules Management application  249
The RULESET view is displayed. For information about this view, place the
cursor on the view name and press **PF1**.

When the RULESET view is displayed, verify that the desired setbase and group
are shown at the top of the view. If they are not, use the SETBS view to select the
correct setbase and return to the RULESET view. Once a setbase is selected, it is
remembered until you change it using the SETBS view.

**Note**

You can also migrate Rules and Rule Sets from classic Rules to Rules
Management. For more information see "Migrating from classic Rules" in this
manual.

2

3 Issue the ADD command in the COMMAND field to start creating a new Rule
Set.

**Figure 131: RULESET view**

```
W1 =RULESET==========NS81========ddmmmyyyy==14:51:30====MVAO=====D====1

Setbase.....: NevenaTestSetbaseAO81Rules
Group.......: SCOFFICE

<table>
<thead>
<tr>
<th>CMD Name</th>
<th>RuleSet</th>
<th>Fltr</th>
<th>FltrMode</th>
<th>Member</th>
<th>Description</th>
<th>LastChg</th>
<th>LastChg</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAORUL01</td>
<td>All</td>
<td>No</td>
<td></td>
<td>64</td>
<td>BAONIS5 05MAY2016</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

The ADD Ruleset dialog is displayed.

4 Enter values for the required fields on the ADD Ruleset dialog.

**Figure 132: ADD Ruleset dialog**

```
ADD Ruleset

Setbase Information:
Setbase: DEFAULT
Group..: BBPLEX01
```

250 *MainView AutoOPERATOR Basic Automation Guide Volume 1: Using Rules*
New Ruleset Information:

Name.:  (Required)
Desc:
Strategy:  First  All
Add Rsn:

END to save the new ruleset
CANCEL to exit without saving
HELP to see dialog help

For more information about any of the required fields, place the cursor on the field and press PF1.

5 After filling the required information on the dialog, enter END to save the new Rule Set definition.

Creating a Rule Pool

All of the Rules belong to a Rule Pool. From the Rule Pool, a Rule can be assigned to different Rule Sets but the Rule is not duplicated.

To create a Rule Pool

1 Select Rule Pools and Rules from the EZAORUL view.

Figure 133: EZAORUL view

The RPOOL view is displayed. For information about this view, place the cursor on the view name and press PF1.
2 Issue the ADD command in the COMMAND field to start creating a new Rule Pool.

**Figure 134: RPOOL view**

<table>
<thead>
<tr>
<th>Rule Pool Name</th>
<th>Description</th>
<th>Last Chg</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAORUL01.AAONS81.Rule.Pool</td>
<td>Pool created by import of AAORUL01 from BAONIS5</td>
<td>10AUG2016</td>
</tr>
<tr>
<td>AAORUL02.AAONS81.Rule.Pool</td>
<td>Pool created by Repeat against AAORUL01 BAONIS5</td>
<td>11APR2016</td>
</tr>
</tbody>
</table>

The RPOOL view shows a list of Rule Pools that have been defined in the registry. Each Rule Pool can contain any number of Rules. You can use Rule Pools to group Rules. For example, Rule Pools may represent geographical locations, business divisions, products (CICS, DB2, IMS, and so on) or any other grouping or combination that makes sense in your organization.

The ADD (New) Rule Pool dialog is displayed.

3 Enter values for the required fields on the ADD (New) Rule Pool dialog.

**Figure 135: ADD (New) Rule Pool dialog**

For more information about any of the required fields, place the cursor on the field and press **PF1**.

4 After filling the required information on the dialog, enter END to save the new Rule Pool definition.

**Creating a Rule**

This section describes creating a new Rule. You can also import classic Rules. For more information, see the section "Migrating classic Rules to Rules Management Rules".

The first thing you have to do is select the Rule Pool where the Rule will exist.
To select the Rule Pool where the new Rule will be stored

1. Select Rule Pools and Rules from the EZAORUL view.

Figure 136: EZAORUL view

The RPOOL view is displayed. For information about this view, place the cursor on the view name and press PF1.

2. Issue the S line command in the CMD field next to the Rule Pool where you want to store the Rule.

Figure 137: RPOOL view

The RULES view is displayed. For information about this view, place the cursor on the view name and press PF1. The name of the selected Rule Pool is remembered by the interface so the next time, you can go directly to the RULES view without stopping at the RPOOL view.

3. Enter ADD on the COMMAND field.

Figure 138: RULES view

The name of the selected Rule Pool is remembered by the interface so the next time, you can go directly to the RULES view without stopping at the RPOOL view.
The Rule Processor Detail Control dialog is displayed.

4 Use the Rule Processor Detail Control dialog to enter information about the Rule.

Figure 139: Rule Processor Detail Control dialog

<table>
<thead>
<tr>
<th>Rule Processor Detail Control - Rule ADD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>COMMAND</strong> ===&gt; SCROLL ===&gt; CSR</td>
</tr>
<tr>
<td>MORE: +</td>
</tr>
</tbody>
</table>

- **Rule Identification:**
  - Rule Pool...: AAORUL01.AAONS81.Rule.Pool

- **Rule Name...:** (Required)
- **Event Type...:** (Required)
- **Initial Mode:** S Enabled Disabled Test

- **Criteria Match Rate Threshold**
  - If Matched.: (Max times matched within INTERVAL, 0-100)
  - In seconds.: (Interval length, 1-99999 seconds)
  - then status.: Suspend Disable Noaction

- **Application Information**
  - Group...:
  - Function:
  - Code....:
  - Author..:
  - Desc....:

- **Add**
- **Rsn:**

Next to set additional parameters
- **END** to exit and save changes
- **CANCEL** to exit without saving changes
- **HELP** to view related help
- **S1** to jump to Selection Criteria
- **SV** to jump to Variable Dependencies
- **A1** to jump to Event Actions
- **AV** to jump to Set Variables
- **AA** to jump to Alert Actions
- **BZ** to jump to BEIM Actions/Alerts

You can see that the rules editor dialog provides shortcuts to other parts of the dialog to help you skip the sections your Rule might not need. Navigate through the Rule creation dialogs, enter values in the fields for the Rule, and enter END to save the Rule.

5 On the RULES view, enter the AR command in the CMD field next to the newly created Rule to add the Rule to a Rule Set. This must be done so that the Rule can be loaded into a MainView AutoOPERATOR PAS.

Figure 140: RULES view

| ddmmmyyyy 11:22:54 ------ MAINVIEW WINDOW INTERFACE (V6.1.01) --- COMMAND ===> SCROLL ===> CSR |

Creating a Rule
### Rules List

#### Pool Name: AAORUL01.AAONS81.Rule.Pool

<table>
<thead>
<tr>
<th>RuleName</th>
<th>Initial</th>
<th>Text-id</th>
<th>EXEC</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAABBBC</td>
<td>EXT</td>
<td>Disabled</td>
<td>AAABBBC</td>
<td></td>
</tr>
<tr>
<td>AAOSSUP</td>
<td>JRNL</td>
<td>Enabled</td>
<td>PM0010I</td>
<td>STARTTOM</td>
</tr>
<tr>
<td>ALERTASV</td>
<td>ALRT</td>
<td>Disabled</td>
<td>NEVITEST</td>
<td>@AOTRACE@</td>
</tr>
<tr>
<td>AU1034E</td>
<td>JRNL</td>
<td>Enabled</td>
<td>AU103*</td>
<td></td>
</tr>
<tr>
<td>AU1034EM</td>
<td>MSG</td>
<td>Enabled</td>
<td>AU103*</td>
<td></td>
</tr>
<tr>
<td>BBSECSAF</td>
<td>JRNL</td>
<td>Enabled</td>
<td>AA32+++</td>
<td></td>
</tr>
<tr>
<td>BQO1127A</td>
<td>IMS</td>
<td>Disabled</td>
<td>A52</td>
<td>BP099743</td>
</tr>
<tr>
<td>CICSLONG</td>
<td>CICS</td>
<td>Disabled</td>
<td>WURD01</td>
<td></td>
</tr>
<tr>
<td>AR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CICSTEST</td>
<td>CICS</td>
<td>Disabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DAVELEVY</td>
<td>TIME</td>
<td>Disabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DFHZC596</td>
<td>CICS</td>
<td>Disabled</td>
<td>DFSZC5966</td>
<td></td>
</tr>
<tr>
<td>DFHZC693</td>
<td>CICS</td>
<td>Disabled</td>
<td>DFSZC6935</td>
<td>NEWCICS</td>
</tr>
<tr>
<td>DFS3466I</td>
<td>IMS</td>
<td>Disabled</td>
<td>DFS3466I</td>
<td></td>
</tr>
<tr>
<td>DFS4445I</td>
<td>IMS</td>
<td>Disabled</td>
<td>DFS4445I</td>
<td></td>
</tr>
</tbody>
</table>

The Add to Ruleset(s) dialog is displayed.

6. Enter the S line command next to the Rule Sets that you want the Rule to belong to and enter END.

**Figure 141: Rule Processor Detail Control dialog**

<table>
<thead>
<tr>
<th>Cmd</th>
<th>RuleSet</th>
<th>Setbase Name</th>
<th>Group</th>
<th>Desc (partial)</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>AAORUL01</td>
<td>NevenaTestSetbaseAO81Rules</td>
<td>SCOFFICE</td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>AORULIMS</td>
<td>DEFAULT</td>
<td>BBPLEX01</td>
<td>Ruleset for IMS and</td>
</tr>
<tr>
<td>s</td>
<td>AORULIMS</td>
<td>IMS.Production.Rules</td>
<td>SCOFFICE</td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>AORULIMS</td>
<td>IMS.Production.Rules</td>
<td>SCOFFICE</td>
<td>Ruleset for IMS and</td>
</tr>
</tbody>
</table>

The Rule is added to the Rule Sets you selected.
7 To associate the Rule Set with a PASDef where you want the Rule to be active, display the PASDEF view from the EZAORUL menu. Issue the RM command next to the PASDef.

**Figure 142: PASDEF view**

```
>W1 =PASDEF=------------------NS81=-------------------ddmmmyyyy=11:38:02=-----MVAO=-----D=-----2
CMD PASDef Name                      Group    Description              Val PAS
----- -------------------------------- Name---- ------------------------ --Count
RM  AAONS81.PAS.Definition           BBPLEX01 Created by import              1
   AAONS81.Second.PAS.Definition    SCOFFICE Testing alternate PASDef       0
```

The PASDSMBR view is displayed.

8 Enter EDIT in the **COMMAND** field on the PASDSMBR view to switch the mode from BROWSE to EDIT and allow you to make changes.

The mode is displayed on the top of the view.

9 To add a Rule Set, enter the INSERT command on the **COMMAND** line.

**Figure 143: PASDSMBR view**

```
>W1 =PASDSMBR=------------------NS81=-------------------ddmmmyyyy=11:41:39=-----MVAO=-----D=-----1
- PASDef Rulesets
  PASDef Name....: AAONS81.PAS.Definition
  Group Name.....: BBPLEX01
  Description....: Created by import
  Last Change....: (none)
  Change Reason.:  
  CMD Ruleset  SeqNo Initial  Ruleset  Fltr  Fltr       Rule Description
  --- -------- ----- Mode---- Strategy ---- Mode---- -Count ---------------------
       AAORUL01     0 Enabled  All      No                66
```

The List Rulesets dialog is displayed.

10 Enter the S line command next to the Rule Set you want to add to the PASDef and enter END on the **COMMAND** line.

**Figure 144: List Rulesets dialog**

```
List Rulesets

COMMAND ===> 
Instructions
Select one or more rulesets from the supplied list to be inserted into the PASDef. Use the rule filter to quickly locate the desired ruleset(s).
```

256 MainView AutoOPERATOR Basic Automation Guide Volume 1: Using Rules
The PASDSMBR view is displayed.

11 Enter SAVE on the COMMAND line to save your changes.

To begin using the Rule you just created, you must deploy and activate the PASDef.

12 Refer to “Deployment of Rules, Rule Sets, and PASDefs” on page 267 for more information about deploying the Rule after you have created it.

## Editing a Rule

This section describes editing a Rule that is already a member of an active Rule Set.

The first thing you have to do is select the Rule pool where the Rule exists.

### To select the Rule pool where the Rule is stored

1 Select the Rule Pools and Rules view from the EZAORUL view.

**Figure 145: EZAORUL view**

<table>
<thead>
<tr>
<th>Real-time Information</th>
<th>Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>.MVAO AOPlex Members</td>
<td>.List of Views</td>
</tr>
<tr>
<td>.AOPlex Member Checkup</td>
<td>.List of User Views</td>
</tr>
<tr>
<td>.Ruleset List</td>
<td>.List of Screens</td>
</tr>
<tr>
<td>.Local Options/Parms</td>
<td>.Select MVAO Target</td>
</tr>
<tr>
<td></td>
<td>.AGS Product Messages</td>
</tr>
<tr>
<td>Administration</td>
<td>.Select SSI Context</td>
</tr>
<tr>
<td>.PAS Definitions</td>
<td>.Select Product</td>
</tr>
<tr>
<td>.Valid PAS List</td>
<td>Tools</td>
</tr>
<tr>
<td>.Setbases</td>
<td>.Import Ruleset(s)</td>
</tr>
<tr>
<td>.Rulesets</td>
<td>.Deployment Queue</td>
</tr>
<tr>
<td>.Rule Pools and Rules</td>
<td></td>
</tr>
</tbody>
</table>

Chapter 10  Rules Management application  257
The RPOOL view is displayed. For information about this view, place the cursor on the view name and press PF1.

2 Issue the S line command in the CMD field next to the Rule pool where the Rule exists.

Figure 146: RPOOL view

<table>
<thead>
<tr>
<th>Rule Pool Name</th>
<th>Description</th>
<th>LastChg User</th>
<th>LastChg Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAORUL01.AAONS81.Rule.Pool</td>
<td>62 Pool created by import of AAORUL01 from BAONIS5</td>
<td>10AUG2016</td>
<td></td>
</tr>
<tr>
<td>AAORUL02.AAONS81.Rule.Pool</td>
<td>49 Pool created by Repeat against AAORUL01 BAONIS5 11APR2016</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The RULES view is displayed. For information about this view, place the cursor on the view name and press PF1.

3 Enter the E line command next to the Rule you want to edit under the CMD heading.

Figure 147: RULES view

<table>
<thead>
<tr>
<th>Rule Name</th>
<th>Initial</th>
<th>Text-id</th>
<th>EXEC</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAABBBC</td>
<td>EXT</td>
<td>Disabled</td>
<td>AAABBBC</td>
<td>STARTTOM</td>
</tr>
<tr>
<td>AAOSSUP</td>
<td>JRNL</td>
<td>Enabled</td>
<td>PM0010I</td>
<td>@AOTRACE@</td>
</tr>
<tr>
<td>ALRT</td>
<td>Disabled</td>
<td>NEVITEST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AU1034E</td>
<td>JRNL</td>
<td>Enabled</td>
<td>AU103*</td>
<td></td>
</tr>
<tr>
<td>AU1034EM</td>
<td>MSG</td>
<td>Enabled</td>
<td>AU103*</td>
<td></td>
</tr>
<tr>
<td>BBSECSAF</td>
<td>JRNL</td>
<td>Enabled</td>
<td>AA32+++</td>
<td></td>
</tr>
<tr>
<td>BQO1127A</td>
<td>IMS</td>
<td>Disabled</td>
<td>A52</td>
<td>BPO99743</td>
</tr>
<tr>
<td>CICSLONG</td>
<td>CICS</td>
<td>Disabled</td>
<td>WURD01</td>
<td></td>
</tr>
<tr>
<td>CICSTEST</td>
<td>CICS</td>
<td>Disabled</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>DAVELEY</td>
<td>TIME</td>
<td>Disabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DFHZC596</td>
<td>CICS</td>
<td>Disabled</td>
<td>DFHZC5966</td>
<td></td>
</tr>
<tr>
<td>DFHZC693</td>
<td>Cica</td>
<td>Disabled</td>
<td>DFHZC6935</td>
<td>NEWCICS</td>
</tr>
<tr>
<td>DFS3466I</td>
<td>IMS</td>
<td>Disabled</td>
<td>DFS3466I</td>
<td></td>
</tr>
<tr>
<td>DFS4445</td>
<td>IMS</td>
<td>Disabled</td>
<td>DFS4445I</td>
<td></td>
</tr>
<tr>
<td>DFS4445M</td>
<td>MSG</td>
<td>Disabled</td>
<td>DFS4445I</td>
<td></td>
</tr>
</tbody>
</table>

The Rule Processor Detail Control dialog is displayed.

4 Use the Rule Processor Detail Control dialog to edit information about the Rule.

Figure 148: Rule Processor Detail Control dialog

- Rule Identification:
  - Rule Pool: AAORUL01.AAONS81.Rule.Pool

- Rule Name: (Required)
- Event Type: (Required)
You can see that the dialog provides shortcuts to other parts of the dialog to help you skip the sections your Rule might not need.

5 Enter END on the last panel of the Rule Processor Detail Control dialog to save your changes to the Rule.

6 Refer to “Deployment of Rules, Rule Sets, and PASDefs” on page 267 for more information about deploying the Rule after you have updated it.

**Associating a Rule with a Rule Set**

This section describes associating a Rule with a Rule Set.

**To associate a Rule with a Rule Set**

1 Display the RULES view by entering RULES in the COMMAND line of the Rules Management application.

**Figure 149: RULES view**

<table>
<thead>
<tr>
<th>RuleName</th>
<th>Rule Initial</th>
<th>Text-id</th>
<th>EXEC</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAABBBC</td>
<td>EXT</td>
<td>Disabled</td>
<td></td>
<td>AAABBBC</td>
</tr>
<tr>
<td>AAOSSUP</td>
<td>JRNL</td>
<td>Enabled</td>
<td>PM0010I</td>
<td>STARTTOM</td>
</tr>
</tbody>
</table>
2 Enter the AR line command in the CMD field next to the Rule that you want to add to a Rule Set. The Add to Ruleset(s) dialog is displayed.

Figure 150: Add to Ruleset(s) dialog

3 Enter the S line command next to the Rule Set that you want to associate the Rule under the CMD heading and enter END at the COMMAND line. The RULES view is redisplayed.

4 To use the Rule, you must deploy and activate the Rule Set. Refer to “Deployment of Rules, Rule Sets, and PASDefs” on page 267 for more information about deploying the Rule.

Associating a Rule Set with a PASDef

This section describes associating a Rule Set with a PASDef.
To associate a Rule Set with a PASDef

1. Display the PASDEF view by entering PASDEF in the COMMAND line of the Rules Management application or select PAS Definitions from the EZAORUL menu.

![Figure 151: PASDEF view](image)

2. Enter the RM line command in the CMD field next to the PAS definition that you want to add to a Rule Set. The PASDSMBR view is displayed.

![Figure 152: PASDSMBR view](image)

3. Enter the EDIT command and then the INSERT command in the COMMAND line. The List Rulesets dialog is displayed.

4. Enter the S line command next to the Rule Set you want to add and enter END on the COMMAND line.

![Figure 153: List Rulesets dialog](image)
The PASDSMBR view is displayed.

5 Enter SAVE to save your changes.

6 To use the newly affected Rule Set, you must activate and deploy the PASDef. Refer to “Deployment of Rules, Rule Sets, and PASDefs” on page 267 for more information about deploying the Rule Set.

Administrative tasks

The section describes different administrative tasks you can do for Rules Management.

Registry management

It is important to define the correct registry size for your Rules because if the registry becomes full, MainView AutoOPERATOR cannot function correctly. For more information, see chapter Implementing the Rules Registry in the MainView AutoOPERATOR Customization Guide and the documentation for the RTCS registry in the MainView Customization Reference.

In BBPARM member AAOPLEXxx, the REGISTRY_WARNHWM parameter allows you to set a high-water mark percentage where a warning message is issued. The warning message AGSRY0014W is issued when either one of the following situations is true:

- The MainView AutoOPERATOR PAS is being initialized.
- The high-water mark has increased at least one percent since the last time the AGSRY0014W message was issued and one minute has passed since the last time the message was issued.

If the registry becomes full and unusable, MainView AutoOPERATOR will issue the following message:

AGSRY1221E Registry is full
Changing the registry size

If you find that you need to change the registry size, you should make a backup of the registry. The backup can be used to restore the registry. To change the size of the repository, use the following instructions.

1. Make a backup of the MainView AutoOPERATOR registry.
   Use the sample JCL found in BBSAMP member AAOAGSBK, which creates a flat-file backup of the registry. The JCL has a number of comments at the top to assist in customizing the JCL. Execute the JCL to create a flat-file backup of the MainView AutoOPERATOR registry.

2. Delete and reallocate the registry following the procedure "Allocating a Registry" in the MainView AutoOPERATOR Customization Guide.

3. Specify a larger registry.

4. Restore the old registry into the newly created registry.
   Use the sample JCL found in BBSAMP member AAOAGSRS, which loads a registry backup into the target registry.
   Note that the target registry can be a newly-created completely empty registry data set or it can be an already-populated registry data set. In the case of the already-populated registry, the restore will merge the flat-file backup into the registry, create new items as needed and replacing existing items as needed.

Migrating from classic Rules

Existing classic Rules are imported directly from the BBPARM data set(s) where they are currently stored. Classic Rules that are to be imported must be in data sets that are included in the BBIPARM DD concatenation of the PAS.

To migrate classic Rules into a registry, the MainView AutoOPERATOR PAS must be running in ADMIN or FULL mode so the registry is open for you to populate it with imported Rules.

In addition, you need to think about setbases. All of the Rule Sets are stored in the registry in setbases. You might decide to put all of your classic Rules and Rule Sets into a single setbase (such as the DEFAULT setbase) or you can import them into any number of setbases. A single setbase named DEFAULT is provided in the registry.
You must choose a default setbase, using the SETBS view, before invoking the Rules import. The Rules import utility, as well as the RULESET view itself, always assumes that you want to work with Rule Sets in the setbase that you have defined. Rule Sets from one or more setbases can be used in any PAS without restriction so you can group your Rule Sets in whatever way makes sense for your enterprise.

**To restart the MainView AutoOPERATOR PAS in ADMIN mode**

1. Specify USAGE_MODE_RULES=ADMIN in BBPARM member AAOPLXxx.

   If the keyword is set to OFF, the registry is not available to the PAS so the import utility will not function. The PAS must be running in ADMIN or FULL mode to use the import utility.

2. Restart the MainView AutoOPERATOR PAS.

   No automation from registry-based Rules will occur while the PAS is in ADMIN mode. Automation from classic Rules will function normally.

**To import classic Rules**

1. Select Import Ruleset(s) from the Tools portion of the EZAORUL view.

   **Figure 154: IMPTRSET view**

   ```plaintext
   ddmmmyyyy  13:11:50 ------ MAINVIEW WINDOW INTERFACE (V6.1.01)  ---------------
   COMMAND  ===>                                                 SCROLL ===> CSR
   CURR WIN ===> 1        ALT WIN ===>                      
   >W1 =IMPTRSET=DEFAULT=BBPLEX01=ddmmmyyyy=13:11:49-----MVAO-----D=37
   - Ruleset Import                                        Hyperlinks:............
   Target Setbase                                  Name........: DEFAULT
   Group.......: BBPLEX01         ...Review Import Messages
   CMD Ruleset  Suggestd Rule   Description                              Vers  Seri
   --- Name---- Status-- Count-- ---------------------------------------  ---- Numb
   AAORUL01 ENABLED      66                                          0801 0000
   AAORULBQ DISABLED     N/A  0000
   AAORULMQ DISABLED     N/A  0000
   AAORULBC DISABLED     N/A  0000
   AAORULBD DISABLED     N/A  0000
   AAORULBG DISABLED     N/A  0000
   AAORULBE DISABLED     N/A  0000
   AAORULBI DISABLED     N/A  0000
   AAORULBA DISABLED     N/A  0000
   AAORULBB DISABLED     N/A  0000
   AAORULBF DISABLED     N/A  0000
   AAORULBH DISABLED     N/A  0000
   AAORULBP DISABLED
   
   The IMPTRSET view is displayed. For information about this view, place the cursor on the view name and press **PF1**.

   This view displays a list of all the classic Rule Sets in the BBIPARM DD concatenation. The utility can import any Rule Set in the list no matter its status (ENABLED or DISABLED).
The Suggested Status shown for each Rule Set is based on the contents of the RULESET= keyword in BBPARM member AAOPRMxx that is used by the target PAS. As a result, the Suggested Status might not represent the Rule Sets that are currently active in the target PAS. It is recommended that the Suggested Status be used, however, since the goal of the import is to recreate the RULESET= keyword’s contents in the registry.

2 Enter the TAG line command in the CMD field next to the Rule Sets you want to import.

3 Enter IMPORT on the COMMAND field and press ENTER.

![Figure 155: Ruleset Import Options view](image)

The Rule Set Import options dialog is displayed. The dialog provides default specifications and suggests names and settings for items such as the Rule pool to house the Rules, the Rule Set’s name and setbase, a PASDef to use (or create), and so on.

4 Review the suggested options on the dialog, make any modifications if necessary, and issue END to initiate the import process.
You will need to repeat the END command for each Rule Set that is being imported.

**How to verify the imported Rules**

Once all of the classic Rules have been imported, you can use the following views to review them.

- **RPOOL view**: use this view to display a list of Rule pools. You can select a Rule pool to view the Rules within the pool.

- **RULESET view**: use the RULESET view to see and confirm the list of Rule Sets that were imported. On this view, you can use the RM line command against a Rule Set to see a list of Rules within a Rule Set.

- **PASDEF view**: use the PASDEF view to see a list of PASDefs, including the one you just created. Use the VP line command to see a list of which PASes are defined to the PASDef. Use the RM line command to see a list of Rule Set members of the PASDef.

As you review the imported elements, you can make decisions about on what names work well for you and which (if any) you might like to change. Then you can stop the PAS, redefine the registry data set, restart the PAS and use the import utility to import your Rule Sets using new Rule pool names, Rule Set names, and so on that meet your needs.

**Activating the imported Rules**

After you have reviewed the imported Rules and Rule Sets, you can deploy and activate the Rules, Rule Sets, and PASDefs in a PAS that is running in FULL mode. If the PAS you are using is already running in FULL mode, you can use the deployment feature to deploy the PASDef for the PAS to activate the PASDef or you can simply restart the PAS using a COLD start.

If the PAS that will be using the newly imported Rules and Rule Sets is running in ADMIN mode, specify `USAGE_MODE_RULES=FULL` in BBPARM member AAOPXLxxx and restart the MainView AutoOPERATOR PAS with a COLD start.

You can also specify `AUDIT_STARTUP_RULESETS=YES` in BBPARM member AAOPXLxxx and when the PAS is COLD started, the PAS issues the message AGSIA4050I with a list of the Rule Sets that the PAS will use.
Deployment of Rules, Rule Sets, and PASDefs

Regardless of the level of registry sharing in an AOPlex, you can deploy Rules, Rule Sets, and PASDefs to remote registries as needed and activate the object(s) in the remote PASes.

This means that several PASes can share a single registry image and, at the same time, the registry image can also be used to store Rules, Rule Sets, or PASDefs for other MainView AutoOPERATOR PASes that do not share the registry image. All the Rules, Rule Sets, and PASDefs reside in the central repository from which you can deploy them to remote registries for the MainView AutoOPERATOR PASes to use.

You can deploy based on levels: Rules, Rule Sets, and PASDefs. During deployment, you must set the following two options:

- Replace (YES | NO) objects in the target PAS
- Activate (YES | NO) objects in the target PAS.

You can choose to deploy without activating (ACTIVATE=NO) and wait for the next cold start of the PAS or use some other manual means to activate the Rules or Rule Sets. If you choose ACTIVATE=YES, the object(s) and all lower level object(s) are deployed and activated at one time.

To determine what deployment level is needed (Rule, Rule Set, or PASDef) you need know the highest level that was changed. The PASDef is at the highest level and the Rule is at the lowest level.

For example, if you create a new Rule and add it to an existing Rule Set, the highest level that was changed was the Rule Set level. Therefore you must deploy and activate the Rule Set. If, however, a new Rule Set was also created for the new Rule and you added the Rule Set to an existing PASDef, then you must deploy and activate the PASDef.

Deploy tagged items

The DEPLOY can be initiated on any of the following MainView AutoOPERATOR views:

- PASDEF
- RULESET
- RULES

Deployment can be used to store definitions in a different local MainView AutoOPERATOR registry or to send the definitions to a different AOPlex reachable within the current CASPlex.
The DEPLOY command copies the tagged Rules, Rule Sets or PASDefs to another system by using MainView CAS-to-CAS communications. It invokes a series of dialogs that sends the tagged items to another target (typically another AOplex) without requiring you to export and import the Rules registry to another CAS.

On any of the views, enter the TAG (T) line command next to the definitions you want to deploy, enter DEPLOY on the COMMAND line, and press ENTER. The Deployment dialog is displayed and you can enter values for the deployment.

**Manage and monitor the deployment process**

You can use the DPLYMNT view to monitor and manage the deployment process. The DPLYMNT view displays a list of deployment requests along with status information about each request. From the DPLYMNT view, you can perform the following actions:

- Stop, start, delete, and cancel deployment requests by using the line commands
- See details about a deployment request by using the Select line command, which displays additional information about the request
- See details about the status of a deployment request by hyperlinking from the status, which displays the LOGDEPLM view.

---

**Note**

When a deployment is finished and you have reviewed the results, be sure to clean-up the deployment request using the D line command on the DPLYMNT view. Failure to delete the accumulated requests might result in an out-of-space condition in the CAS at some point in the future.

**Deploying a Rule Set**

The following example shows how to deploy a Rule Set.

**To deploy a Rule Set**

1. Display the RULESET view.

---

### Figure 156: RULESET view

<table>
<thead>
<tr>
<th>CMD</th>
<th>RuleSet</th>
<th>Fltr</th>
<th>FltrMode</th>
<th>Member</th>
<th>Description</th>
<th>LastChg</th>
<th>LastChg</th>
</tr>
</thead>
<tbody>
<tr>
<td>AADRLUL01</td>
<td>All</td>
<td>No</td>
<td>66</td>
<td>Production ruleset</td>
<td>BAONIS5</td>
<td>19OCT2016</td>
<td></td>
</tr>
<tr>
<td>AADRLUL02</td>
<td>First</td>
<td>No</td>
<td>0</td>
<td>Ruleset for testing system</td>
<td>BAONIS5</td>
<td>19OCT2016</td>
<td></td>
</tr>
</tbody>
</table>
2. Enter the TAG line command in the CMD field next to the Rule Sets you want to deploy and press ENTER.

3. Enter DEPLOY on the COMMAND field and press ENTER. The Ruleset(s) Deployment (1 of 3) dialog is displayed.

Figure 157: Ruleset Deployment (1 of 3) dialog

Specify YES for the Replace Ruleset(s) and Activate Ruleset(s) fields and enter Next to go to the next step.

The Ruleset(s) Deployment (2 of 3) dialog is displayed.

4. Confirm the Ruleset(s) that you want to Deploy.

The S column is used to select or exclude a listed item for deployment. Type S to select an item. Leave the field blank to exclude an item. You can also customize the Replace and Activate options for each item individually.
In this example, the AAORUL01 Rule Set has been selected to be replaced and activated by the deployment.

Enter Next to proceed to the next step.

The Targets dialog is displayed.

Select the Targets for the deployment.

5 Select the Targets for the deployment.

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>SCROLL</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter Next to proceed to the next step.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The Targets dialog is displayed.

5 Select the Targets for the deployment.

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>SCROLL</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter Next to proceed to the next step.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The Targets dialog is displayed.

Select the Targets for the deployment.

| Context: | Product: MVAO |
|------------------|--------|------|
| S Target System Server |
| S NS81 MVIMSD NS81 |

Deploy to start deployment
Select to select a context
SELECTALL to select all listed targets of context
UNSELECTALL to unselect all listed targets of context
ENTER to refresh context
CAncel to exit without deploying

Initially, the Context field displays the context or target that was entered with the deploy command. This field is empty if context or target is not specified. If the field is empty or you want to change the deployment target(s), enter a full or generic context and press ENTER to see a list of contexts

To select a context, enter the S command (for Select) and choose a context from the list or, type a context name in the field and press ENTER.

6 Enter Deploy in the COMMAND field to start the deployment.

Deploy is initiated and the DPLYMNT view is displayed to provide the status.

Realtime views

The following sections describe some tasks you can do in realtime.
Determine which Rule Sets are active in a PAS

You can determine which Rule Sets are active in your PAS by performing the following steps:

1. Select the Ruleset List from the EZAORUL menu. The RSETR view is displayed.

2. Review the list of Rule Sets that are active on the PAS.

   These Rule Sets are what the PAS is currently using for automation. The Automation Status, Automation Strategy, Honor MPF Suppression, and other statistics about the Rule Set are displayed on this view.

The following is an example of the RSETR view.

```
>W1 =RSETR=============NS81=====*========ddmmmyyyy==15:31:41====MVAO=====D====1
  - Real-time Rulesets....
    Automation Control---
    Automation Status.... Active (Cmd: STATACT, STATINA )
    Automation Strategy.. All (Cmd: STRATI, STRATA, STRATF)
    Honor MPF Suppression No (Cmd: MPFYES, MPFNO )
    Automation Statistics
    Total Events.......... 1178028 Display Suppressed............ 1002
    Total AO Events...... 1178028 Hardcopy Suppressed............ 732
    Total Events Handled. 1416 Rule-generated Alerts............ 6
    Current Arrival Rate. 19 Rule-scheduled EXECs............ 253
    Peak Arrival Rate.... 154

    CMD Ruleset       Rule    Fired Filtered F Enabled Enabled Enabled
                      Name---- Status-- ---Count ---Count ---Count S Date----- Time---- Strategy
                      --- Name---- Status-- ---Count ---Count ---Count S Date----- Time---- Strategy
    AAORUL01 ENABLED  66      691   756032   11-OCT-16 10:07:24 N/A
```

Determine which Rules have fired in a PAS

You can determine which Rules have fired in your PAS by performing the following steps:

1. Select the Ruleset List from the EZAORUL menu. The RSETR view is displayed.

2. Enter the S line command next to a Rule Set listed on the view. The RULESR view is displayed.

   The RULESR view displays all of the Rules in the Rule Set that are available to the PAS.

```
>W1 =RULESR============NS81=====*========ddmmmyyyy==15:27:59====MVAO=====D====66
  - Real-time Rules..
    Ruleset Overview
    Name............ AAORUL01  Serial# 00000
    Version........ 0801      Setbase test1
```
Utilities

The following lists some utilities that you can use to help you using the Rules Management application.

Registry backup

Use the JCL for registry backup located in BBSAMP member AAOAGSBK. This JCL creates a flat file backup of the registry that can be restored when you need it. Review the comments in the beginning of the JCL to help you customize the JCL and submit the JCL to create a backup.

Registry restoration

In the event that you need to restore from a backup registry, use the JCL from BBSAMP member AAOAGSRS. This JCL will restore the registry with a backup that you made and load it into the target registry.

Note that the target registry can be a newly created and empty registry data set or it can be an already-populated registry data set. In the case of the already populated registry, the restore will merge the backup into the registry, create new items as needed, and replace existing items as needed.

Export Rules from a registry

An export utility is provided for use with a registry. You can use the export utility to make backup copies of Rule Sets instead of (or in addition to) making full registry backups.

The exported Rule Sets are stored in a partitioned data set (PDS) with the name of each Rule Set being the name of the PDS member. The exported Rule Sets can be
used in a PAS that does not use a registry (such as at a disaster recovery site) or they could be imported into the same or a different registry for use.

The export utility is accessible from any REXX EXEC that you execute in any PAS that has access to the registry with the Rule Set(s) you want to export. You can find a sample REXX EXEC in BBSAMP member QAOEXPRT. The sample EXEC contains documentation for you to customize the utility.

BBI command differences

The following lists some differences between classic Rules and Rules Management Rules and how they affect the BBI commands.

**RESET (.E)**

The RESET (.E) command, when used with the parameter (P) sub-parameter, supports Rule Sets names using the classic naming convention (RULxxxxx or AAORULxx).

If classic Rule Set names are being used in the registry, this command will continue to function as in the past. If Rule Set names other than RULxxxxx or AAORULxx are used in the registry, you cannot use this command to reset them.

*Note*

If non-classic names are used in the registry, you can replace the RESET (.E) command to reload a Rule Set with using two .T commands. Use the first .T command to disable (DI) the Rule Set and the second .T command to enable (EN) the Rule Set. These two commands, in effect, accomplish the same result as using the RESET command.

For example, replace the following command:

```
.E P MYRULSET
```

with the following two commands:

```
.T RULESET,DI,MYRULSET
.T RULESET,EN,MYRULSET
```

**SET (.T)**

The SET (.T) command with the RULESET parameter supports both classic and non-classic Rule Set names and performs the same functions as in the past.
DISPLAY RULESET ALL (.D RULESET ALL)

The .D RULESET ALL command response is enhanced to issue new message AU1507I when the PAS is using Rules registry:

AU1507I Rule Sets loaded from the Registry

Differences between classic Rules and Rules Management Rules

The following lists some differences between classic Rules and Rules Management Rules.

Aside from the most obvious difference (windows mode vs. fullscreen mode), one of the primary goals was to modernize and increase flexibility. To this end, there are a number of differences when editing Rules with the windows mode editor. One of the first is that there are fewer dialogs and the dialogs are scrollable forward and backward. And, because the editor works in windows mode, it is now available directly in MVE. The following sections outline some of the other differences.

DELAY() keyword

The DELAY() keyword that delays a Rule from scheduling an EXEC or the issuing a command is enhanced to support a SHARED variable as the delay value. This allows the flexibility to dynamically set the amount of delay for all commands or EXECs that use the SHARED variable.

For example, now you can specify the variable &MYVAR to delay scheduling an EXEC.

```
COMMAND ===>                                              SCROLL ===> PAGE
REXX EXEC or CLIST
EXEC andParms 1
STARTTOM

Optional Modifier Keywords
Priority:   Normal    High    Hot    First    Single    Xsingle
Delay....: &MYVAR

Note: Total length of all input values must be less than 255 characters

END to complete the request
CANCEL to abandon the request
HELP to view related help
```
Change reason

In Rules Management, Rules allow you to specify a change reason where you can document the reason for the change and retain the reason as part of the Rule's change history. A record of up to the last ten (10) changes is kept for review using a hyperlink.

For example, the following example text shows a change reason added at the bottom of the dialog.

----------------------- Rule Processor Detail Control ----------------------

Rule Identification:
Rule Pool...: AAORUL01.AAONS81.Rule.Pool
Rule Name...: AAOSSUP
Serial #....: 0000000   Updt Version: 0800

Event Type..: JRNL  (Required)
Initial Mode: S Enabled   Disabled   Test

Criteria Match Rate Threshold
If Matched..: 2     (Max times matched within INTERVAL, 0-100)
in seconds..: 10   (Interval length, 1-99999 seconds)
then status.:   Suspend S Disable   Noaction

Application Information
Group....:
Function:
Code....:
Author..: NIS1
Desc....:

Chg Rsn: Updating the Rule to add Delay for scheduling an EXEC

Last Changed by BAONIS5  on 16/03/22 12:11
Last Rsn: Imported rule from BBPARM

Next to set additional parameters
END to exit and save changes
CANcel to exit without saving changes
HELP to view related help
S1 to jump to Selection Criteria
SV to jump to Variable Dependencies
AI to jump to Event Actions
AV to jump to Set Variables
AA to jump to Alert Actions
BZ to jump to BEIM Actions/Alerts

Radio buttons

The use of radio buttons is only noticeable when using the MainView Explorer user interface. Required fields that force you to choose one from a list of values in Rules Management use radio buttons. Radio buttons provide an easy way to choose an option without having to deselect whatever option was previously selected. Selecting the new option automatically deselects any previous option.
Checkboxes (YES or NO)

Virtually all of the Rules options that required you to specify YES or NO have been converted to checkboxes. For example, instead of Display at Dest: ___ where you specify YES or NO (or leave it blank to get the default), the Rules Management editor uses a checkbox that looks like this: Display at Dest: _ Yes.

Normally, a checkbox that is left unchecked is the same as leaving a field blank in the full-screen editor. The default value is implied as being the opposite of the supplied prompt. That is, if the default is NO, then the checkbox will be checked to get the YES option.

Checkboxes (one of many choices)

Rules fields that contain a one of many values are now shown as checkboxes. This makes it obvious what the options are without needing to memorize any acronyms or abbreviations, as well as making it easy to select the desired option. You can see an example of this change with the Rule action of Command. In classic Rules, you had to either know what value to use for the command type or use Help to find out. Now you can see two rows of checkboxes. These checkboxes identify all the valid command types that you can select from.

Larger data entry fields

The Rules Management windows mode editor provides enough space for all fields to be entered in their entirety without the use of special helper dialogs. For example, on the classic Rules Actions dialog (A1), the EXEC/Parms field is limited to 126 characters due to screen size restriction. The Rules Management windows mode editor supplies the full 255-character field for data entry.

Journal option

Many Rule event types support the Journal option. This option echoes the event's message to the BBI-SS journal when selected. Many event types also offer the ability to reword the event's message or command. For these event types, the Journal option has a third variation. The third option (besides YES and NO) is REW. When REW was specified, the reworded message/command is used for the text sent to the BBI-SS Journal by the option. In the new Rules editor, you will now see a separate checkbox labeled Reworded associated with the Journal option when it is available for that Rule event type.

Issue WTO or Journal Message

The classic Rules action for Issue WTO (or Journal or Both) has been split. In the classic fullscreen editor, you had to choose what to issue (WTO, JRN, or BOTH) and enter the text to be issued. In the Rules Management windows mode editor,
these fields are now split apart. You no longer a need to specify what to issue separately from the text. If you want to issue a WTO, you can put the text in the Issue WTO text field.

This is also true for the Issue Journal Message text field. Splitting the field means that you can send different texts to WTO and the Journal in the same Rule.

**Mixed case entry**

In the classic full-screen Rules editor, the use of mixed-case values was severely restricted and required special coding allowances. Mixed-case, however, is much more common now than it was in the past.

The Rules Management windows mode editor has expanded the number of fields that support mixed-case data entry. These fields are documented in the online help. The primary fields affected by this change on the Actions dialog (A1) are the EXEC/Parms, Reword, Issue WTO, and Issue Journal Message fields. Additionally, on the Alert Action dialog (AA), the Alert Text and Alert Follow-Up EXEC/Parms also support mixed-case values. The Final Action field on the AA dialog supports mixed-case only when the EXEC option is selected. Otherwise, it is uppercase only.

**BiiZ and BEIM changes**

Several changes have been made to the Rules BEIM Actions dialog (BZ). Some of these were to improve usability and others where to allow the user full use of data fields that were previously restricted in size or number in classic Rules.

In classic Rules the BEIM Event Action Name/Value Pairs, these pairs were restricted both in number and in length. The classic Rules panel allowed you to enter a fixed number of eight pairs with the name length set to a maximum of 33 characters, and the slot value length is restricted to 38 characters.

These restrictions were implemented due to space considerations on the panel but for some customers, these limitations were too restrictive. In the Rules Management windows mode editor, the slots now support a variable number of slots. The slot name still has a maximum of 33 characters but you can enter a slot value up to 255 bytes. The total number of slots is limited only by the aggregate length of all the slot pairs. The maximum length is 600 bytes (including the 4 bytes of overhead for each name/value pair).

In classic Rules the BEIM Alert Action Name/Value pairs were also restricted to a fixed number of eight pairs, the maximum name length is set to 33 characters, and the slot value length is restricted to 38 characters. In the Rules Management windows mode editor, the number of pairs is still eight and each slot name is still limited to 33 characters. However, each slot value has now been expanded to a maximum of 255 bytes per slot value.
Prompter panels

The classic full screen Rules editor offers a series of dialogs that were collectively called the prompter panels. The prompter panel is invoked when you enter a question mark into a field followed by a blank and pressing enter. The resulting panel would provide help that you can select such as a command type or the panel would assist in the creation of any optional prefix keywords. This is especially useful for some command types that supported a large variety of prefix keywords.

The Rules Management windows-mode editor still supports a prompter system but it is now required to use them when editing any of the Rule fields that provided a prompter panel in classic Rules. In addition, the Rules Management editor uses the term modifier instead of prefix. The term modifier more accurately reflects what the keywords do to the command or EXEC. They modify or otherwise alter when and how the command is issued.

There are primarily two reasons for this change: better usability and improved accuracy. No longer will you be tempted to guess at what the modifier keywords are and which one(s) are required. No longer will you be tempted to stack EXECs when stacking is not allowed. This change makes the creation of Rules easier as well as making you aware of modifier keywords that you might not have known before.
Field descriptions for the Rule Processor

This appendix contains examples of panels from the Rule Processor and tables that describe the fields available.

Selection Criteria fields

Not every selection criteria field can be used for every event type.

The second column in the following table shows which event types can use a specific selection criteria field. For any event type for which you create a Rule, a selection criteria panel for that event type appears and contains a subset of the criteria listed in Table 24 on page 279.

The following table lists all of the available selection criteria fields for all event types.

That TIME-initiated Rules do not use any of the selection criteria listed that are here. To see the selection criteria for TIME-initiated Rules, refer to “Creating TIME-initiated Rules” on page 119.

Table 24: Selection Criteria fields and event types

<table>
<thead>
<tr>
<th>Selection Criteria field</th>
<th>Event type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acct Info</td>
<td>CICS, CMD, EXT, IMS, MSG, TLM</td>
<td>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.</td>
</tr>
<tr>
<td>Selection Criteria field</td>
<td>Event type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Alarm ID</td>
<td>ALRM</td>
<td>Enter the Alarm ID associated with the Alarm. SHARED variables, EVENT variables, and pattern matching can be used in this field.</td>
</tr>
<tr>
<td>Alarm Text</td>
<td>ALRM</td>
<td>Enter the text string that all ALARM messages are searched for.</td>
</tr>
<tr>
<td>Cell Name</td>
<td>BRC</td>
<td>Select all events that match the cell name. SHARED variables, EVENT variables, and pattern matching can be used in this field.</td>
</tr>
<tr>
<td>Cell Type</td>
<td>BRC</td>
<td>Select all events that match the cell type. SHARED variables, EVENT variables, and pattern matching can be used in this field.</td>
</tr>
<tr>
<td>CICS TDQ</td>
<td>CICS</td>
<td>Enter the one- to four-character name of the CICS Transient Data Queue where the event originated. If the queue is an indirect queue, specify the indirect queue name instead of the name the queue resolves to. The indirect queue name is the queue that MainView AutoOPERATOR compares to the destination queue of the incoming message. SHARED variables, EVENT variables, and pattern matching can be used in this field.</td>
</tr>
<tr>
<td>Class</td>
<td>BRC</td>
<td>Select all events that contain the class string. SHARED variables, EVENT variables, and pattern matching can be used in this field.</td>
</tr>
<tr>
<td>Client</td>
<td>IMP</td>
<td>Specify a 1- to 8-byte name of the client (IMSplex member) where the IMS Type-2 command originated. SHARED variables, EVENT variables, and pattern matching can be used in this field. <strong>Note:</strong> This selection criteria field is not valid for IMSplex commands that are issued from MainView AutoOPERATOR or MainView for IMS.</td>
</tr>
<tr>
<td>Console Name</td>
<td>CMD, MSG</td>
<td>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.</td>
</tr>
<tr>
<td>Context</td>
<td>ALRM</td>
<td>Enter the context associated with the Alarm. SHARED variables, EVENT variables, and pattern matching can be used in this field.</td>
</tr>
<tr>
<td>Count</td>
<td>TLM</td>
<td>Specify the maximum number of Time Limit Extensions are allowed to occur for this address space. Values can be from 0 to 9999. SHARED variables and EVENT variables can be used in this field.</td>
</tr>
<tr>
<td>Selection Criteria field</td>
<td>Event type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Desc codes</td>
<td>MSG</td>
<td>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.</td>
</tr>
<tr>
<td>Selection Criteria field</td>
<td>Event type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Event</td>
<td>ALRT</td>
<td>Specify when a Rule should fire when a matching ALERT is detected. Valid values are as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <strong>ADD</strong> (default): The Rule fires when a matching ALERT is added (created).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <strong>DELETE</strong>: The Rule fires when a matching ALERT is deleted manually or by way of automation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <strong>DELETEQ</strong>: The Rule fires when the matching ALERT queue is deleted manually or by way of automation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <strong>ANY</strong>: The Rule fires for any matching ADD, DELETE, or DELETEQ event.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specifying <strong>ANY</strong> is useful when you want the same action to occur when an ALERT is added and when it is deleted. When you specify <strong>ANY</strong>, a single Rule can accomplish automation for both cases.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When <strong>ANY</strong> is in effect, only the following fields are used to determine a match:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— <strong>ADD</strong>: TEXT ID, TEXT STRING, KEY, or QUEUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— <strong>DELETE</strong>: KEY or QUEUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— <strong>DELETEQ</strong>: QUEUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <strong>ESC</strong>: The Rule fires when a matching ALERT is being escalated. All ALERT variables will contain &quot;after&quot; values for the escalation. The following fields can be used to determine a match:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— TEXT ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— TEXT STRING</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— KEY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— QUEUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <strong>ALL</strong>: The Rule fires for any matching ADD, DELETE, DELETEQ, or ESC event. See <strong>ANY</strong> for details about how ALERTs are matched for only ADD, DELETE, and DELETEQ events.</td>
</tr>
<tr>
<td>Selection Criteria field</td>
<td>Event type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Host Address</td>
<td>BRC</td>
<td>Select all events that match the host IP address. SHARED variables, EVENT variables, and pattern matching can be used in this field.</td>
</tr>
<tr>
<td>Host Class</td>
<td>BRC</td>
<td>Select all events that match the host class. SHARED variables, EVENT variables, and pattern matching can be used in this field.</td>
</tr>
<tr>
<td>Host Name</td>
<td>BRC</td>
<td>Select all events that match the host name. SHARED variables, EVENT variables, and pattern matching can be used in this field.</td>
</tr>
</tbody>
</table>
| Ignore leading plus     | MSGDB2     | Specify YES to ignore a leading plus sign (+) in the text-ID of write-to-operator messages (WTOs).  
Note: Leave this field blank to use the default that is specified for the Rule Set on the Rule Set Default Values panel (refer to “Enabling filtering and Rule Set match rate for Rule Sets” on page 221 for more information. Specify NO to ignore the default that is specified in the Rule Set Default Values panel.  
Messages that are issued from a non-authorized program can have a plus sign (+) inserted at the beginning of the text-ID of the message. By specifying YES in this field, you can create a single Rule that fires for messages that are issued from both authorized and non-authorized programs because the Rule ignores the leading plus sign. When you specify YES, you need to 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.  
This setting does not impact the contents of the IMFTEXT or WORD1 event variables.  
For more information about how to set a default for all the MSG, HWTO, or DB2 initiated Rules within a Rule Set to ignore the leading plus sign, refer to the description of the Default ignore leading plus value for MSG rules field in “Enabling filtering and Rule Set match rate for Rule Sets” on page 221.  
You can use the STRIP built-in Rules function on the Advanced Variable Dependencies panel to ignore a leading plus signs in the &IMFTEXT variable during variable value comparisons: for example; STRIP(&IMFTEXT,L,'+')  
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. |
| Interval                | Time       | Specify (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. |
### Selection Criteria fields

<table>
<thead>
<tr>
<th>Selection Criteria field</th>
<th>Event type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Jobclass</strong></td>
<td>CICS, CMD, EXT, IMS, MSG, TLM</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Job name</strong></td>
<td>CMD, DB2, EOSEOM, EXT, HWTO, IMS, MSG, NVC, NVM, TLM</td>
<td>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 that is issuing the event. SHARED variables, EVENT variables, and pattern matching can be used in this field.</td>
</tr>
<tr>
<td><strong>Key</strong></td>
<td>ALRT</td>
<td>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 <em>MainView AutoOPERATOR Advanced Automation Guide</em>.</td>
</tr>
<tr>
<td>Selection Criteria field</td>
<td>Event type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| MLWTO Minor             | MSG        | Specify how a Rule processes the minor lines of a multiline write-to-operator (MLWTO) message. Valid values are ALL or SINGLE. Specifying ALL means that the Rule treats the entire MLWTO (the major message plus all the minor message lines) as a single entity. A Rule with ALL specified does not fire unless the entire MLWTO has been issued. In other words, it does not fire until the MLWTO is considered complete by z/OS or MainView AutoOPERATOR or the text line with a text type of E or DE has been received. For more information about message text type codes, refer to the IBM publication *MVS Programming: Assembler Services Reference, Volume 2*. When you specify ALL, BMC Software recommends that you use the variables &LINE_x_WDn on the Variable Dependencies panel to create Rules that fire based on specific content of a MLWTO message. The &LINE_x variable resolves for each line of the MLWTO and the WDn variable resolves to words on the first (major) line. For example, specify the following values on the Selection Criteria Panel:  
<p>| Text ID    | IEA995I    |
| MLWTO Minor | ALL        |</p>
<table>
<thead>
<tr>
<th>Selection Criteria field</th>
<th>Event type</th>
<th>Description</th>
</tr>
</thead>
</table>
| MLWTO Minor (Continued)  | MSG        | On the Variable Dependencies panel, specify: **LINE6_WORD1:6:11 EQ MYPROG**  
The Rule fires for the message IEA995I only if the 6th through 11th characters of the first word on the sixth line equals MYPROG.  
Specify SINGLE so that the Rule treats each minor line of the MLWTO as a separate entity.  
This value means that each minor line of the MLWTO can cause the Rule to fire. When a single minor line matches the Selection Criteria, the Rule fires immediately.  
BMC Software recommends that you use the Variable Dependencies panel with the &LINE2_WORDn variable or &IMFMTEXT variables to filter out unwanted minor lines.  
On the Variable Dependencies panel, when you use the variable &LINE2_WORDn, &LINE2 resolves to any line of the MLWTO and the individual &WORDn variables resolve to words on the first (major) line.  
For example, specify the following values on the Selection Criteria Panel:  

<table>
<thead>
<tr>
<th>Text ID</th>
<th>IEA995I</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLWTO Minor</td>
<td>SINGLE</td>
</tr>
</tbody>
</table>

On the Variable Dependencies panel, specify: **LINE2_WORD1:6:11 EQ MYPROG**  
The Rule fires for the message IEA995I if the 6th through 11th characters of the first word of any line equals MYPROG.
<table>
<thead>
<tr>
<th>Selection Criteria field</th>
<th>Event type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLSEG Minor</td>
<td>IMS</td>
<td>Specify how a Rule processes the minor segments of a multisegment IMS (MLSEG) message. Valid values are ALL or SINGLE. Specify ALL if you want the Rule to treat the entire MLSEG (the major plus all the minor segment lines) as a single entity. A Rule with ALL specified does not fire unless the entire MLSEG has been received. In other words, it does not fire until IMS has sent the last segment of a message (or last response for a command), or the maximum time to wait between receiving segments (IMSMSTIM) has been reached. When you specify ALL, BMC Software recommends that you use the variables &amp;LINEx_WORDn on the Variable Dependencies panel to create Rules that fire based on specific content of a MLSEG message. The &amp;LINEx variable resolves for each segment of the MLSEG and the individual &amp;WORDn variables resolve to the words on the first (major) segment. For example, specify the following values on the Selection Criteria Panel: Text ID ===&gt; DFS2503W MLSEG Minor ===&gt; ALL On the Variable Dependencies panel, specify: LINE3_WORD3:1:8 EQ BE2PCUST The Rule fires for the IMS message DFS2503W only if the 1st through 8th characters of the third word on the third segment equals BE2PCUST.</td>
</tr>
<tr>
<td>Selection Criteria field</td>
<td>Event type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MLSEG Minor</td>
<td>IMS</td>
<td>Specify SINGLE if you want the Rule to treat each minor segment of the MLSEG as a separate entity. SINGLE means that each minor segment of the MLSEG can cause the Rule to fire. When a single segment line matches the Selection Criteria, the Rule fires immediately. BMC Software recommends that you use the Variable Dependencies panel with the &amp;LINE2_WORDn variables or &amp;IMFTEXT to filter out unwanted minor segments. On the Variable Dependencies panel, when you use the variable &amp;LINE2_WORDn, &amp;LINE2 resolves to any segment of the MLSEG and the individual &amp;WORDn variables resolve to words on the first (major) segment. Example: Specify the following values on the Selection Criteria Panel: Text ID ===&gt; DFS2503W MLSEG Minor ===&gt; SINGLE On the Variable Dependencies panel, specify: LINE2_ WORD3:1:8 EQ BE2PCUST The Rule fires for message DFS2503W if the 1st through 8th characters of the third word of any segment equals BE2PCUST.</td>
</tr>
<tr>
<td>Selection Criteria field</td>
<td>Event type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>MLTDQ</td>
<td>CICS</td>
<td>Specify whether a CICS Rule should process only the first line of a multiline CICS (MLTDQ) message or the entire message. Valid values are ALL or blank. Specify ALL if you want the Rule to treat the entire MLTDQ (the major plus any minor lines) as a single entity. A Rule with MLTDQ ==&gt; ALL specified does not fire unless the entire MLTDQ is received. In other words, it does not fire until CICS has sent the last line of a message. When you specify ALL, use the variables &amp;LINEx_WORDn on the Variable Dependencies panel to base the firing of the Rule on specific content of the MLTDQ. &amp;LINE1 resolves to the line of the MLTDQ message; &amp;LINE1 contains the contents of the major line of the MLTDQ, and &amp;LINE2 through &amp;LINEx contain the subsequent minor lines. WORDn resolves to a word on &amp;LINEx. For example, assume that you specify the following values on the Selection Criteria panel: Text ID ===&gt; DFHXS1111 MLTDQ ===&gt; All On the Variable Dependencies panel, specify: LINE2_WORD7:4:8 EQ 00000008 The Rule fires for the CICS message DFHXS1111 only if the fourth through eighth characters of the seventh word on the second line equals 00000008.</td>
</tr>
<tr>
<td>Name</td>
<td>VAR</td>
<td>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.</td>
</tr>
<tr>
<td>Origin</td>
<td>JRNL</td>
<td>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.</td>
</tr>
<tr>
<td>Priority</td>
<td>ALRM</td>
<td>Enter the priority associated with the Alarm. SHARED variables, EVENT variables, and pattern matching can be used in this field.</td>
</tr>
<tr>
<td>Queue</td>
<td>ALRM ALRT</td>
<td>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.</td>
</tr>
<tr>
<td>Selection Criteria field</td>
<td>Event type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>RACF Group</td>
<td>CICS CMD EOS EXT IMS MSG TLM</td>
<td>Enter the RACF group ID (1- to 8-bytes) 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.</td>
</tr>
<tr>
<td>RACF User</td>
<td>CICS CMD EOS EXT IMS MSG TLM</td>
<td>Enter the RACF user ID (1- to 8-bytes) 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.</td>
</tr>
<tr>
<td>Route codes</td>
<td>MSG</td>
<td>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.</td>
</tr>
<tr>
<td>Route list</td>
<td>IMP</td>
<td>Specify a 1- to 30-byte Route list of the IMS Type-2 command. The ROUTE list is a list of client names separated by commas. The ROUTE list can contain a single asterisk for a client name, which represents all clients. SHARED variables, EVENT variables, and pattern matching can be used in this field.</td>
</tr>
<tr>
<td>Scope</td>
<td>ALRM</td>
<td>Enter the scope associated with the Alarm. SHARED variables, EVENT variables, and pattern matching can be used in this field.</td>
</tr>
<tr>
<td>Seconds</td>
<td>TLM</td>
<td>Specify the maximum number amount of time in seconds allowed for extensions for this address space. Values can be from 0 to 9999. SHARED variables and EVENT variables can be used in this field.</td>
</tr>
<tr>
<td>Severity</td>
<td>BRC</td>
<td>Select all events that match the severity. SHARED variables, EVENT variables, and pattern matching can be used in this field.</td>
</tr>
<tr>
<td>Selection Criteria field</td>
<td>Event type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Start Time</td>
<td>Time</td>
<td>Specify (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 A.M. You can enter a start time of ATSTART. This specifies that Rules will begin firing at BBI-SS start-up. <strong>Note:</strong> TIME-initiated Rules that use ATSTART are automatically suspended or disabled if the Rule is modified. When you modify TIME-initiated Rules that uses ATSTART, a confirmation panel is displayed that warns you about suspending or disabling the Rule, and allows you to cancel or proceed with the changes. Refer to “Example 2: creating a TIME-initiated Rule with ATSTART” on page 121 to see an example of the Confirm Timer Rule Update panel.</td>
</tr>
<tr>
<td>Step CC</td>
<td>EOS</td>
<td>Select all messages whose condition code matches. SHARED variables, EVENT variables, and pattern matching can be used in this field.</td>
</tr>
<tr>
<td>Step Name</td>
<td>TLM</td>
<td>Specify the name of the step for this Rule to fire. The following values are possible: Batch Jobs: Name of the Step on the EXEC statement. If the JCL invokes a procedure, it will still be the name on the EXEC statement. TSO: TSO user IID STC: Name of a Started Task. If started with an identifier, the identifier is used. SHARED variables, EVENT variables, and pattern matching can be used in this field.</td>
</tr>
<tr>
<td>Step Number</td>
<td>EOS</td>
<td>Select all messages whose step number matches. SHARED variables, EVENT variables, and pattern matching can be used in this field.</td>
</tr>
<tr>
<td>Step Number</td>
<td>TLM</td>
<td>Specify the number of the step where this Rule will fire. Valid values can be from 1 to 255. SHARED variables and EVENT variables can be used in this field.</td>
</tr>
<tr>
<td>Selection Criteria field</td>
<td>Event type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Stop Count</td>
<td>Time</td>
<td>Specify the number of times for the Rule to be fired from the Start Time within a 24-hour interval. 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.</td>
</tr>
<tr>
<td>Stop Time</td>
<td>Time</td>
<td>Specify (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.</td>
</tr>
<tr>
<td>Target Name</td>
<td>BRC</td>
<td>Select all events that match the target name. SHARED variables, EVENT variables, and pattern matching can be used in this field.</td>
</tr>
<tr>
<td>Termination Type</td>
<td>EOM</td>
<td>Select termination type for the End-of-Memory event. NORMAL, ABNORMAL, SHARED variables, EVENT variables can be used in this field.</td>
</tr>
</tbody>
</table>
| 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. 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. |
<table>
<thead>
<tr>
<th>Selection Criteria field</th>
<th>Event type</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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. 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 |
| Time Limit Type          | TLM        | Specify with kind of Time Limit expiration for a Rule to fire. Possible values are CPU, JCPU, SCPU or WAIT.  
  **Note:** It might be difficult to discern whether an address space's Step or Job CPU will expire. Specifying CPU will account for both the Job and the Step CPU time. |
<p>| Tool Name                | BRC        | Select all events that match the tool name. SHARED variables, EVENT variables, and pattern matching can be used in this field. |
| Tool Severity            | BRC        | Select all events that match the tool severity. SHARED variables, EVENT variables, and pattern matching can be used in this field. |
| Type                     | ALRM CMD EOM EOS EXT IMS MSG TLM | 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. |</p>
<table>
<thead>
<tr>
<th>Selection Criteria field</th>
<th>Event type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserId</td>
<td>IMP</td>
<td>Specify a 1- to 8-byte name user ID of application where the IMS Type-2 command originated. SHARED variables, EVENT variables, and pattern matching can be used in this field.</td>
</tr>
<tr>
<td>Variable is</td>
<td>VAR</td>
<td>Specify the action that will fire the Rule; valid values are as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Created</strong>: The Rule will fire when the variable is created in the SHARED pool.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Deleted</strong>: The Rule will fire when the variable is deleted from the SHARED pool.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Upd</strong>: The Rule will fire when the value of the variable is updated in the SHARED pool.</td>
</tr>
</tbody>
</table>

Variable Dependencies fields

The following figure shows an example of the Variable Dependencies panel.

**Figure 158: Variable Dependencies panel**

BMC Software -------- Variable Dependencies - MSG ------- MAINVIEW AutoOPERATOR COMMAND ===> Rule-set === AAORULXX Rule-id === ICH70001 OR/AND Variable-name Op Variable-Value QSMFID__________________________ EQ SYSB_______________________________ OR_ QSMFID__________________________ EQ SYSC_______________________________  

--> ATTENTION: Use ASV command to add ASV criteria to the rule <--  
Press ENTER to continue, END return to Detail Control, CANCEL to cancel changes

The following table describes the fields on this panel.
### Table 25: Variable Dependencies panel field descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable-Name</td>
<td>Is the name of the SHARED or LOCAL variable to be tested.</td>
</tr>
<tr>
<td>Op</td>
<td>Is the operand used in the variable evaluation; valid operands are as follows:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operand</th>
<th>Character or hexadecimal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EQ</td>
<td>character</td>
<td>Specify that the variable must equal a specified value.</td>
</tr>
<tr>
<td>NE</td>
<td>character</td>
<td>Specify that the variable must not be equal to a specified value.</td>
</tr>
<tr>
<td>GT</td>
<td>character</td>
<td>Specify that the variable must greater in value than a specified value.</td>
</tr>
<tr>
<td>LT</td>
<td>character</td>
<td>Specify that the variable must be less in value than a specified value.</td>
</tr>
<tr>
<td>GE</td>
<td>character</td>
<td>Specify that the variable must be greater than or equal in value to a specified value.</td>
</tr>
<tr>
<td>LE</td>
<td>character</td>
<td>Specify that the variable must be less than or equal in value to a specified value.</td>
</tr>
<tr>
<td>EX</td>
<td>hexadecimal</td>
<td>Specify that the queried variable excludes the value specified.</td>
</tr>
</tbody>
</table>

For example, use this operand to verify that the text MQRO_COD is contained in the value of the WebSphere MQ variable IMFQ_MD_REPORT (Report Options) which resolves to MQRO_PASS_CORREL_ID MQRO_COD.

For example, use this operand to verify that the text MQRO_COA is excluded from the value of the WebSphere MQ variable IMFQ_MD_REPORT (Report Options) which resolves to MQRO_PASS_CORREL_ID MQRO_COD.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HE</td>
<td>hexadecimal Specify that the queried variable is equal to the specified value. For example, the WebSphere MQ variable IMFQ_MD_CORREL ID (Correlation ID) contains the hexadecimal 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 contains the specified value x'80BE'.</td>
</tr>
<tr>
<td>HN</td>
<td>hexadecimal Specify that the queried variable is not equal to the specified value. For example, the WebSphere MQ 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'.</td>
</tr>
<tr>
<td>HG</td>
<td>hexadecimal Specify that the queried variable is greater than the specified value. For example, the WebSphere MQ 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'.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>Op</td>
<td>Is the operand used in the variable evaluation; valid operands are as follows:</td>
</tr>
<tr>
<td></td>
<td><strong>Operand</strong></td>
</tr>
</tbody>
</table>
| HL    | hexadecimal | Specify that the queried variable is less than the specified value.  
For example, the WebSphere MQ 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 | Specify that the queried variable is greater than or equal to the specified value.  
For example, the WebSphere MQ 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'. |
| LX    | hexadecimal | Specify that the queried variable is less than or equal to the specified value.  
For example, the WebSphere MQ variable IMFQ_MD_CORRELID (Correlation ID) contains the value x'0B21'. 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 | Specify 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. |
**Op**

Is the operand used in the variable evaluation; valid operands are as follows:

<table>
<thead>
<tr>
<th>Operand</th>
<th>Character or hexadecimal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FN</td>
<td>hexadecimal</td>
<td>Specify 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 the specified mask but not in the variable.</td>
</tr>
</tbody>
</table>

**Variable-Value**

Is the value that the variable is compared against. You can specify a numeric value up to 15 digits long or specify a variable that resolves to a number up to 15 digits long.

**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.

---

**Describing Action Specification fields**

Actions Specifications are shown on two panels for each event type: Action Specification(s) I - *event type* and Action Specification(s) II - *event type*.

Not every action specification field can be used for every event type. The second column of the following table shows which event types can use an action specification field. For any event type for which you create a Rule, an action specification panel for that event type appears and contains a subset of the actions listed in Table 26 on page 299.

The following table lists all of the available actions that you can select.
<table>
<thead>
<tr>
<th>Action Specification field</th>
<th>Event type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audit reject</td>
<td>IMP</td>
<td>Specify whether to issue Audit message to the Journal if this Rule rejects OM (IMSplex) command with <em>Reject cmd ===&gt; YES</em>. Valid values are YES and NO. NO is the default. YES causes the following Journal message to be issued if this Rule fires and has <em>Reject cmd ===&gt; YES</em>: DX0642I RULE rrrrrrr RULESET sssssss REJECTED OM cmd: cccccccc.</td>
</tr>
<tr>
<td>Aux Log</td>
<td>All event types</td>
<td>Specifies that the event message is captured and displayed in the AuxLog logspace in the MainView Logger. For more information about how to set up the AuxLog logspace and how to use MainView Logger, see the <em>MainView AutoOPERATOR Customization Guide</em>. For more information about viewing event messages written to AuxLog by using the AUXLOG windows-mode view, see the <em>MainView AutoOPERATOR Basic Automation Guide, Volume 2</em>.</td>
</tr>
<tr>
<td>Cancel Job</td>
<td>EOS</td>
<td>Specify whether to cancel the job. If you leave this field blank, the Job is not canceled.</td>
</tr>
<tr>
<td>Color</td>
<td>MSG</td>
<td>Specify a color for the WTO message; valid values are as follows: RED, GREEN, BLUE, YELLOW, TURQUOISE, PINK, WHITE. Default value is none.</td>
</tr>
<tr>
<td>Console Name</td>
<td>MSG</td>
<td>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.</td>
</tr>
<tr>
<td>Cmd (Type xxx)</td>
<td>All event types</td>
<td>Refer to “Select Command Type panel” on page 311.</td>
</tr>
<tr>
<td>Action Specification field</td>
<td>Event type</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>CTM Force Job</td>
<td>All event types</td>
<td>Specifies that the FORCEJOB request is sent to all subscribing Control-O or CMEM regions in the local LPAR. From there it can be sent to other INCONTROL products in the sysplex by using INCONTROL resources and facilities. For complete details and samples of the subfields for this action, please see the Control-O User Guide. The sub-fields are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <strong>Table</strong>: (required field) name of Control-M scheduling table</td>
</tr>
</tbody>
</table>
|                            |             | ■ **Job**: (optional field) job name  
|                            |             | If this field is blank, all jobs in the specified table are forced. |
|                            |             | ■ **Date**: scheduling date for the jobs; valid values can be:  
|                            |             | — **spec_date**: a specific date, in *mmddyy*, *ddmmyy* or *yymmdd* format, depending on the site standard  
|                            |             | — **ODAT**: (default value) resolves to the Control-M installation working date  
|                            |             | — **DATE**: resolves to the system date  
|                            |             | ■ **Library**: (required field) name of the Control-M scheduling library containing the specified table |
CTM Set Condition

All event types

Specifies to set a condition.

The request to set a condition is sent to all subscribing Control-O or CMEM regions in the local LPAR. From there it can be sent to other INCONTROL products in the sysplex by using INCONTROL resources and facilities.

For complete details and samples of the subfields for this action, please see the Control-O User Guide.

The sub-fields are:

- **Name**: descriptive name of from 1 through 20 characters. Only trailing blanks are permitted.

- **Date**: (optional field) 4-character date reference; valid values can be:
  - **Date**: a specific date in either **mmdd** or **ddmm** format, depending on the site standard
  - **ODAT**: resolves to the current installation working date (default)
  - **DATE**: resolves to the system date
  - **STAT**: indicates that the condition, such as IMS-ACTIVE, is not date dependent

- **Action**: (required field) valid values are **ADD** and **DEL**

Descriptor codes

MSG

Specify new MVS descriptor codes or replace the original MVS descriptor codes of the message (38-character maximum). 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 addition of new codes. Variables cannot be used in this field.
### Action Specification field

<table>
<thead>
<tr>
<th>Action Specification field</th>
<th>Event type</th>
<th>Description</th>
</tr>
</thead>
</table>
| Display at dest.           | ALRTCICSDB2IMSJR NLMSG | Specify 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; in addition:  
- 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). 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 Rule 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.  
**Note:** When you specify the selection criteria **MLWTO Minor = ALL** or **MLWTO Minor = SINGLE** for a MSG-initiated Rule, the Rule cannot take the action of suppressing the message. Therefore, the action field Display at dest. (which normally appears for MSG-initiated Rules on the Action Specification panel), does not appear when you specify **MLWTO Minor = ALL** or **MLWTO Minor = SINGLE**. |

<table>
<thead>
<tr>
<th>DOM ID</th>
<th>All event types</th>
<th>Specify the name of a SHARED variable that contains the delete-operator-message ID of a previously issued WTO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOM MSG</td>
<td>MSG</td>
<td>Specify that MainView AutoOPERATOR will issue message DX9505I when the action message which caused this Rule to fire is deleted.</td>
</tr>
<tr>
<td>EXEC name/Parms</td>
<td>All event types</td>
<td>Refer to “REXX Exec or CLIST panel” on page 308.</td>
</tr>
<tr>
<td>Action Specification field</td>
<td>Event type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| Extend Time               | TLM        | Specify a value in seconds to extend the CPU or WAIT time for a Job, Started Task or TSO User. If the Rule is firing for Step CPU, the Step CPU is extended. If the Rule is firing for a Job, the Job CPU is extended. If the Rule is firing for Wait time, the wait time is extended. Valid values are as follows:  
  - blank - Do not extend the job.  
  - numeric value from 1 to 255.  
Event or SHARED variables are supported. |
| Hilite                    | MSG        | Specify a highlighting value for the WTO message; valid values are as follows:  
  - NO specifies that the Rule does not change the highlight attribute.  
  - BLINK causes the message to blink when it appears.  
  - UNDER changes the message to appear with an underscore.  
  - REVERSE changes the message to appear with reverse video highlighting.  
You cannot specify more than one value. Default value is none. |
| Intensity                 | MSG        | Specify an intensity value for the WTO message; valid values are as follows:  
  - HIGH causes the message to appear with high intensity.  
  - NORMAL causes the message to appear with normal intensity.  
The message intensity setting is ignored for multicolored consoles. Default value is none. |
<p>| Issue Msg (WTO, JRN, or WJ)| All event types | Refer to “Issue Message panel” on page 318. |</p>
<table>
<thead>
<tr>
<th>Action Specification field</th>
<th>Event type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Journal</td>
<td>ALRT, CICS, CMD, DB2, EXT, IMP, IMS, MSG, TIME, VAR</td>
<td>Specify 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 <strong>Reword Msg</strong> field. For event type MSG:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- if you journal the message, the entire message (including the reply ID) is written to the journal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- if you use IMFTEXT, the reply ID is stripped from the message</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- WORD1 is the first word after the reply ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- TEXT ID is the first word after the reply ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- TEXT STRING does not include the reply ID</td>
</tr>
<tr>
<td>POST Target</td>
<td>All event types</td>
<td>Specify the SSID of the target BBI-SS PAS that will execute the POST action of an EXEC. If this field is not specified, the target is the local BBI-SS PAS that fires the Rule. The target name can be 1-32 characters in length and can be a variable name.</td>
</tr>
<tr>
<td>POST ID</td>
<td>All event types</td>
<td>Specify the matching name information that is specified in the WAIT parameter of the EXEC that causes a waiting EXEC to be posted to run The POST ID can be 1-32 characters long and may be a variable name. For example, to post an EXEC that is waiting for a name of BACKUP, enter BACKUP in this field.</td>
</tr>
<tr>
<td>POST Parms</td>
<td>All event types</td>
<td>When you are using the POST function in a Rule, you can specify optional parameter information that is passed in the IMFPOST variable to the waiting EXEC. The POST Parms value can be 1-56 characters and special characters and imbedded blanks are supported. One or more variable names are allowed in this field.</td>
</tr>
<tr>
<td>Reject Command</td>
<td>CMD</td>
<td>Specify whether command will be issued or rejected.</td>
</tr>
<tr>
<td>Action Specification field</td>
<td>Event type</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Reject Command</td>
<td>IMP</td>
<td>Specify whether the command is rejected. Valid values are YES and NO. NO is the default. YES means the command will not be executed. <strong>Note:</strong> When you specify YES, the OM command is still passed to all matching Rules in all eligible MainView AutoOPERATOR address spaces for automation. After all matching Rules fire, the OM command is rejected and is not passed to any other IMS OM input user exits.</td>
</tr>
<tr>
<td>Reword ALERT</td>
<td>ALRT</td>
<td>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.</td>
</tr>
<tr>
<td>Reword CMD</td>
<td>CMD</td>
<td>Enter a new command that replaces the original command. All attributes associated with the command are retained. The maximum length of a command replacement is 126 characters.</td>
</tr>
<tr>
<td>Reword CMD</td>
<td>IMP</td>
<td>Enter the command text to be sent to its destination. Note that the reworded command text is used for all subsequent Rule matching comparisons. The length of the reworded text cannot exceed the length of the original command plus 80 characters up to a maximum of 255 characters.</td>
</tr>
<tr>
<td>Action Specification field</td>
<td>Event type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| Reword Msg                | IMSJRNLMSGCICS | 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 event type, 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 limited to 212 characters. If variables are used, the maximum length is 512 characters (after the contents of the variables are resolved). Any resolved value greater than 512 characters will be truncated to 512 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        | Specify new MVS route codes or replace the original MVS route codes of the message (61 characters maximum). 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 addition of new codes.
Variables cannot be used in this field. |
| Send (TSO)                | All event types | 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. |
### Action Specification field

<table>
<thead>
<tr>
<th>Event type</th>
<th>Description</th>
</tr>
</thead>
</table>
| All event types | Set the value of any global IOA variable. The request to set an IOA variable is sent to all subscribing Control-O or CMEM regions in the local LPAR. From there it can be sent to other INCONTROL products in the sysplex by using INCONTROL resources and facilities. You can use this set of fields to assign a new variable value by using a MainView AutoOPERATOR variable, character string, or an unsigned constant in the second field. Valid values are as follows:  
  - blank: no variable is modified  
  - First Set IOA Variable field: can be up to 33 characters and contain any IOA variable name  
  - Second Set IOA Variable field: can be 43 characters and contain a valid value for the specified variable. |

| MSG | Specify whether the message will be recorded in the MVS SYSLOG. Valid values are YES and NO. YES is the default. Note: When you specify the selection criteria MLWTO Minor = ALL or MLWTO Minor = SINGLE for a MSG-initiated Rule, the Rule cannot take the action of preventing it from being recorded in the MVS SYSLOG. Therefore, the action field SYSLOG Display (which normally appears for MSG-initiated Rules on the Action Specification panel), does not appear when you specify MLWTO Minor = ALL or MLWTO Minor = SINGLE. |

| MSG | Add or replace the route and descriptor codes that are 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. |

### Additional panels

Some of the fields on the Action Specification(s) --event type panels are prefixed with an asterisk (*); the fields are

- * EXEC Name/Parms
In these fields, you can enter the values directly into the input fields or you can enter a question mark and a blank (?) that causes additional panels to be displayed where you can select from a list of options and specify additional keywords.

The following sections describe these panels.

**REXX Exec or CLIST panel**

Use the REXX Exec or CLIST panel to enter the name of an EXEC and any additional (optional) parameters. This panel is available to all event types. This panel is also displayed when you want to specify an EXEC in the Final Action field for a Rule-initiated ALERT.

The following figure shows an example of the REXX Exec or CLIST panel.

**Figure 159: REXX Exec or CLIST panel**

The **EXEC/Parms** field supports up to 255 characters. You can enter EXEC names and additional parameters. If you are invoking this panel from the Alert Actions I or Alert Actions II panel, you can specify multiple EXECs (and associated parameters for each). Separate each EXEC name with double colons (::) between one EXEC string and the next EXEC string. This is called "stacking."

Stacking is not supported in all fields (for example, in the **EXEC/Parms** field on the Action Specifications I panel). If you need to schedule multiple EXECs, you can enter stacked BBI commands in the **Cmd (Type BBI)** field of the panel. An example follows:
Cmd( Type BBI ) ===> %MYEXEC &WORD5::%MYEXEC2 &IMFRLID

**Note**

Depending on how you accessed this panel, the total length of the EXEC name and any additional parameters might have different length limitations.

The following table describes the fields on this panel.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC Parms</td>
<td>Specify the name of an EXEC and any additional (optional) parameters that the Rule schedules to run in the BBI-SS PAS. The first parameter specified in this field is the first parameter passed to the EXEC, which 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.</td>
</tr>
</tbody>
</table>

Appendix A  Field descriptions for the Rule Processor 309
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>Specify the scheduling priority to be assigned to the EXEC. Valid values are Norm, High, hoT, First, Single, and Xsingle. You can abbreviate the value to the single capitalized letter in its name. For example, you can specify an EXEC to run with a hot priority by entering T. The following list describes how the priority affects the way the EXECs are scheduled:</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Norm</strong></td>
</tr>
<tr>
<td></td>
<td>■ <strong>hoT</strong></td>
</tr>
<tr>
<td></td>
<td>■ <strong>First</strong></td>
</tr>
<tr>
<td></td>
<td>■ <strong>Single</strong></td>
</tr>
<tr>
<td></td>
<td>■ <strong>Xsingle</strong></td>
</tr>
</tbody>
</table>
Field name | Description
--- | ---
Delay | Specify the number of seconds to wait before the EXEC is scheduled. Valid values can be 1 to 99999 seconds. All EXECs are delayed by the specified value and the order in which the EXECs are executed is not predictable. If the BBI-SS PAS is stopped for any reason before MainView AutoOPERATOR can run the EXECs that are scheduled with a Delay, the delayed EXECs are not rescheduled when the BBI-SS PAS restarts.

*Daylight Saving Time considerations:* For more information about how changing the local MVS time affects specifying a delayed action, see Appendix, "Managing daylight saving time and MainView AutoOPERATOR time-initiated automation" in the *MainView AutoOPERATOR Basic Automation Guide*, Volume 2.

## Select Command Type panel

When you enter a question mark and a blank (?) in the **Cmd (Type)** field, an additional selection panel is displayed where you can select an EXEC or a command type.

See the following figure.

**Figure 160: Select Command Type panel**

BMC Software ------------ Select Command Type -------------- AutoOPERATOR

COMMAND ===>                                                 TGT --- AOAO

Command:

Line Command: (S)elect

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>MVS/JES command(s) issued from originating address space</td>
</tr>
<tr>
<td>SS</td>
<td>MVS/JES command(s) issued from BBI-PAS address space</td>
</tr>
<tr>
<td>_</td>
<td>BBI Control command(s)</td>
</tr>
<tr>
<td>_</td>
<td>IMS command(s)</td>
</tr>
<tr>
<td>_</td>
<td>IMS Plex command(s)</td>
</tr>
<tr>
<td>_</td>
<td>CICS command(s)</td>
</tr>
<tr>
<td>_</td>
<td>MQ command(s)</td>
</tr>
<tr>
<td>_</td>
<td>NetView command</td>
</tr>
<tr>
<td>_</td>
<td>TOM Total Object Manager command</td>
</tr>
</tbody>
</table>

Press ENTER to continue filling in command. END to return to Action Specification with selected command type. CANCEL to cancel selection.

**Figure 161 on page 312** shows an example of the MVS Command(s) panel that is displayed when you select the MVS command type.

Use the MVS Command(s) panel to specify commands that the Rule issues. This panel is available to all event types. You can enter the following types of commands on the panel:

- **MVS**: MVS or JES commands that are issued from originating address space
SS: MVS or JES commands that are issued from BBI-PAS address space

BBI: BBI Control commands

IMS: IMS commands

IMP: IMS Plex commands

CICS: CICS commands

MQ: IBM MQ commands

NV: NetView commands

TOM: Total Object Manager command

**Figure 161: MVS Command(s) panel**

```
BMC Software ----------------- MVS Command(s) ----------------    AutoOPERATOR
COMMAND ===>                                                 TGT  --- AOAO
Command ===>                                                 
Command ===>                                                 
Command ===>                                                 
Command ===>                                                 
Command ===>                                                 

Console, Command and Response Token keywords apply to all commands
Console ===>                                                 
CART ===>                                                   

Note: Total length of all input values must be less than 126 characters.
Press END to return with command(s), CANCEL to cancel selection.
```

Some versions of the Command(s) panel include fields for additional keywords. Press **PF1/Help** for more information or refer to the following table for information about the keywords.

---

**Note**

Depending on how you accessed this panel, the total length of all input values might have different length limitations.

---

**Table 28: Commands panel**

<table>
<thead>
<tr>
<th>Command type field</th>
<th>Panel displayed and fields description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cmd (Type )</td>
<td><strong>Blank</strong>: no command is issued</td>
</tr>
<tr>
<td>Command type field</td>
<td>Panel displayed and fields description</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>Cmd (Type BBI)</td>
<td><strong>BBI Control Command(s)</strong> panel</td>
</tr>
<tr>
<td></td>
<td><strong>Command:</strong> Specify BBI Control commands in the Command field.</td>
</tr>
<tr>
<td></td>
<td><strong>Target:</strong> Specify the target name of the BBI-SS PAS where the command is to run.</td>
</tr>
<tr>
<td></td>
<td>You can specify only one target name that is 1 to 8-characters long. The Rule sends the BBI Control commands to the specified target. When you specify more than one command, there is no way to ensure the order the commands execute in the target BBI-SS PAS.</td>
</tr>
<tr>
<td></td>
<td><strong>Delay:</strong> Specify a number of seconds to wait before the commands are executed. Valid values can be 1 to 99999 seconds. All commands are delayed by the specified value and the order in which the commands are executed is not predictable.</td>
</tr>
<tr>
<td></td>
<td>If the BBI-SS PAS is stopped for any reason before MainView AutoOPERATOR can run the commands that are scheduled with a Delay, the delayed commands are not re-scheduled when the BBI-SS PAS restarts.</td>
</tr>
<tr>
<td></td>
<td><strong>Daylight Saving Time considerations:</strong> For more information about how changing the local MVS time affects specifying a delayed action, see Appendix, &quot;Managing daylight saving time and MainView AutoOPERATOR time-initiated automation&quot; in the <em>MainView AutoOPERATOR Basic Automation Guide</em>, Volume 2.</td>
</tr>
</tbody>
</table>
### Command type field

<table>
<thead>
<tr>
<th>Command type field</th>
<th>Panel displayed and fields description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cmd (CIC)</strong></td>
<td><strong>CICS Command(s) panel</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Command</strong> field: Specify MainView AutoOPERATOR for CICS commands or CICS transactions (using the CICSTRAN command).**</td>
</tr>
<tr>
<td></td>
<td>If you specify a CICS command, you must include a target name. If you specify a CICSTRAN command, you can include (optional) a user ID specification.</td>
</tr>
<tr>
<td></td>
<td>Examples:</td>
</tr>
<tr>
<td></td>
<td>Cmd (type CIC) =&gt; CICSTEST:ENABLE FILE FILEK</td>
</tr>
<tr>
<td></td>
<td>Cmd (type CIC) =&gt; CICSPROD:CICSTRAN FCPY 'FILEY'</td>
</tr>
<tr>
<td></td>
<td>Cmd (type CIC) =&gt; CI(CICSPROD) U(BAOJDB1):CICSTRAN FCPY 'FILEX'</td>
</tr>
<tr>
<td></td>
<td>When using CICSTRAN, you must enclose CICS transaction parameters with single quotation marks (&quot;'&quot;) and only CICS transactions that run terminal unattached are supported. For terminal attached transactions, use the OSPI application and refer to the <em>MainView AutoOPERATOR Advanced Automation Guide</em>.</td>
</tr>
<tr>
<td></td>
<td>When you specify a user ID, it is added to the EXEC CICS START TRANSID command used to start the CICS transaction.</td>
</tr>
<tr>
<td></td>
<td>In addition, when you specify a user ID with U(<em>userid</em>) parameter, you must also specify the CICS region name with the CI(<em>cicsName</em>) parameter.</td>
</tr>
<tr>
<td></td>
<td>If you do not specify a user ID and CICS security is active, the transaction runs under the authority of the default CICS user ID specification.</td>
</tr>
<tr>
<td></td>
<td>Refer to the <em>CICS Transaction Server for z/OS RACF Security Guide</em> for information about activating transaction security and defining resources, including surrogate CICS USERIDs, to the RACF database.</td>
</tr>
<tr>
<td></td>
<td>For a list of CICS commands, refer to the section &quot;CICS Command Parameters&quot; in the <em>MainView AutoOPERATOR Advanced Automation Guide</em>.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> When you use any of the listed commands in this field, do not prefix the command name with IMFEXEC CICS.</td>
</tr>
<tr>
<td><strong>Cmd (CIC)</strong></td>
<td><strong>(continued)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Target</strong> field: (required) Specify the name of the target CICS for issuing commands.**</td>
</tr>
<tr>
<td></td>
<td><strong>Userid:</strong> <em>(optional)</em> Specify a user ID with a command that starts a transaction.**</td>
</tr>
<tr>
<td></td>
<td><strong>Delay:</strong> Specify a number of seconds to wait before the command is executed. Valid values can be 1 to 99999 seconds. All commands are delayed by the specified value and the order in which the commands are executed is not predictable.</td>
</tr>
<tr>
<td></td>
<td>If the BBI-SS PAS is stopped for any reason before MainView AutoOPERATOR can run the commands that are scheduled with a Delay, the delayed commands are not re-scheduled when the BBI-SS PAS restarts.</td>
</tr>
<tr>
<td></td>
<td><strong>Daylight Saving Time considerations:</strong></td>
</tr>
<tr>
<td></td>
<td>For more information about how changing the local MVS time affects specifying a delayed action, see Appendix, &quot;Managing daylight saving time and MainView AutoOPERATOR time-initiated automation&quot; in the <em>MainView AutoOPERATOR Basic Automation Guide</em>, Volume 2.</td>
</tr>
<tr>
<td>Command type field</td>
<td>Panel displayed and fields description</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>Cmd (IMS)</td>
<td><strong>IMS Command(s) panel</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Command</strong> field: specify IMS commands; for example</td>
</tr>
<tr>
<td></td>
<td><code>/@STA TRAN IDC%IMP</code></td>
</tr>
</tbody>
</table>

When the at sign (@) follows the forward slash sign (/), the IMS command bypasses MainView AutoOPERATOR generic parameter processing (where commands are processed by default) and instead, it is issued directly to IMS. When IMS processes the command, IMS uses its own rules for generic parameter processing.

For more information about how to use generic parameters in IMS commands and which commands and keywords are supported by IMS, refer to the section "Generic Parameters" in the IBM manual *IMS Command Reference*.

**Delay**: specify a number of seconds to wait before the command is executed.

Valid values can be 1 to 99999 seconds. All commands are delayed by the specified value and the order in which the commands are executed is not predictable.

If the BBI-SS PAS is stopped for any reason before MainView AutoOPERATOR can run the commands that are scheduled with a Delay, the delayed commands are not re-scheduled when the BBI-SS PAS restarts.

**Daylight Saving Time considerations**:

For more information about how changing the local MVS time affects specifying a delayed action, see Appendix, "Managing daylight saving time and MainView AutoOPERATOR time-initiated automation" in the *MainView AutoOPERATOR Basic Automation Guide, Volume 2*. 
<table>
<thead>
<tr>
<th>Command type field</th>
<th>Panel displayed and fields description</th>
</tr>
</thead>
</table>
| **Cmd (Type IMP)** | **IMSPlex Command(s) panel**  
Command field: specify IMSPlex commands; for example  
Cmd (Type IMP) ===> STA DB PROD1  
**Route**: specify the names of IMS OM (Operations Manager) Clients (also known as IMS control regions) in the IMSPlex to which the commands are sent; for example:  
Cmd (Type IMP) ===> ROUTE(IMS81X,IMS81Y) : INITIATE OLC PHASE(PREPARE) TYPE(MODBLKS)  
If Route is not specified, the commands are sent to all IMS regions in the IMSPLEX that have registered with the IMS OM to process this command. You can specify up to seven OM Clients.  
**Delay**: specify a number of seconds to wait before the commands are executed. Valid values can be 1 to 99999 seconds. All commands are delayed by the specified value and the order in which the commands are executed is not predictable.  
If the BBI-SS PAS is stopped for any reason before MainView AutoOPERATOR can run the commands that are scheduled with a Delay, the delayed commands are not re-scheduled when the BBI-SS PAS restarts.  
Daylight Saving Time considerations:  
For more information about how changing the local MVS time affects specifying a delayed action, see Appendix, “Managing daylight saving time and MainView AutoOPERATOR time-initiated automation” in the MainView AutoOPERATOR Basic Automation Guide, Volume 2. |
| **Cmd (Type MVS)** | **MVS Command(s) panel**  
Command field: specify MVS console commands in this field; the address space that originated the event will issue the commands  
By default, the MVS commands are issued from Console zero (0). If you use RACF to restrict the usage of MVS commands or MVS consoles, you must ensure that the originating address space has the correct security permissions to use the specified MVS command and MVS console.  
**Console**: specify the name of the console to be used for issuing the MVS commands  
If an invalid console name is specified, the DX0671E message is written to the BBI Journal and the command is not issued. The console does not have to be active for the command to be issued.  
**CART**: specify a Command And Response Token (CART) that allows you to add a token to an MVS command in a Rule and process the response from that specific command in another Rule  
If the CART is larger then 8 characters it is truncated to the first 8 characters and if the CART value is less than 8 characters, the unused characters will be filled in with blanks. |
<table>
<thead>
<tr>
<th>Command type field</th>
<th>Panel displayed and fields description</th>
</tr>
</thead>
</table>
| Cmd (Type MQ)      | **IBM MQ Command(s)** panel  
**Command** field: specify IBM MQ commands; this field is 126 characters long  
For more information about how to enter IBM MQ commands, refer to the *MainView AutoOPERATOR for MQ Installation and User Guide*. |
| Cmd (Type NV)      | **NetView Command** panel  
**Jobname**: for non-NetView event types, you must specify the JOBNAME keyword that identifies the target where the command will be sent  
The JOBNAME keyword uses the format:  
JOBNAME(jobname):NetView_command  
J(jobname):NetView_command  
For the NetView event type, the JOBNAME keyword is optional; if you do not specify it, the default jobname is the NetView system specified on the NETVIEW= parameter in BBPARM member BBISSP00.  
You cannot enter multiple NetView commands.  
**Delay**: specify a number of seconds to wait before the commands are executed  
Valid values can be 1 to 99999 seconds. All commands are delayed by the specified value and the order in which the commands are executed is not predictable.  
If the BBI-SS PAS is stopped for any reason before MainView AutoOPERATOR can run the commands that are scheduled with a Delay, the delayed commands are not re-scheduled when the BBI-SS PAS restarts.  
*Daylight Saving Time considerations*:  
For more information about how changing the local MVS time affects specifying a delayed action, see Appendix, "Managing daylight saving time and MainView AutoOPERATOR time-initiated automation" in the *MainView AutoOPERATOR Basic Automation Guide, Volume 2*. |
**Cmd (Type SS)**

**MVS Command(s) from BBI-SS panel**

**SS**: specify MVS commands that originate from the BBI-SS PAS

The command is issued from the console specified in with the CMDCON parameter in BBPARM member BBISSPxx. The default is Console zero (0). If you use RACF to restrict the usage of MVS commands or MVS consoles, you must ensure that the originating address space has the correct security permissions to use the specified MVS command and MVS console.

**Delay**: specify a number of seconds to wait before the commands are executed

Valid values can be 1 to 99999 seconds. All commands are delayed by the specified value and the order in which the commands are executed is not predictable.

If the BBI-SS PAS is stopped for any reason before MainView AutoOPERATOR can run the commands that are scheduled with a Delay, the delayed commands are not re-scheduled when the BBI-SS PAS restarts.

**Daylight Saving Time considerations**:

For more information about how changing the local MVS time affects specifying a delayed action, see Appendix, "Managing daylight saving time and MainView AutoOPERATOR time-initiated automation" in the *MainView AutoOPERATOR Basic Automation Guide, Volume 2*.

**Cmd (Type TOM)**

**TOM Commands panel**

**Command**: Specify up to 5 TOM operator commands that the Rule issues to the TOM PAS, which must be associated with this MainView AutoOPERATOR PAS. Only the TOM operator commands (such as BLOCK, LOCK, RESET, BOUNCE, and so on) are valid. The commands go directly to the TOM PAS and execute in the same order that they are entered.

**Note**: MainView AutoOPERATOR submits the commands issued in this field only to the TOM that it is connected to.

---

**Issue Message panel**

Use the Issue Message panel to enter message text that the Rule sends.

This panel is available to all event types. The following figure shows an example of the Issue Message panel.

**Figure 162: Issue Message panel**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Issue Message</th>
<th>AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND ====&gt;</td>
<td>WTO, JRN, or WJ</td>
<td>TGT ====&gt; AOAO</td>
</tr>
<tr>
<td>Message Type ====&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Message Text ====&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optional Route and Descriptor Codes (ignored for JRN messages)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Route Codes ====&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Desc Codes ====&gt;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The following table describes the fields on this panel.

### Table 29: Issue Message panel

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Type</td>
<td>Specify where Rule sends the message. Possible values are as follows:</td>
</tr>
<tr>
<td></td>
<td>- WTO: specifies that the message is a write-to-operator (WTO) message</td>
</tr>
<tr>
<td></td>
<td>- JRN: specifies that the message is written to the BBI Journal</td>
</tr>
<tr>
<td></td>
<td>- WJ or JW: specifies that the message is both a WTO and also written to</td>
</tr>
<tr>
<td></td>
<td>the BBI Journal</td>
</tr>
<tr>
<td></td>
<td>EVENT and SHARED variables are valid.</td>
</tr>
<tr>
<td></td>
<td>- if you use IMFTEXT, the reply ID is stripped from the message</td>
</tr>
<tr>
<td></td>
<td>- WORD1 is the first word after the reply ID</td>
</tr>
<tr>
<td></td>
<td>- TEXT ID is the first word after the reply ID</td>
</tr>
<tr>
<td></td>
<td>- TEXT STRING does not include the reply ID</td>
</tr>
<tr>
<td></td>
<td>- if you journal the message, the entire message (including the reply ID)</td>
</tr>
<tr>
<td></td>
<td>is written to the journal</td>
</tr>
</tbody>
</table>

| Message text   | 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 (;) or if you use the **Route Codes** or **Descriptor codes** fields. |

| Route Codes    | Specify one or more route codes to be used for a message sent as a WTO. Separate multiple route codes with a blank; for example 1 2 3 |

| Descriptor Codes | Specify one or more descriptor codes to be used for a message sent as a WTO. Separate multiple descriptor codes with a blank; for example: 5 9 |
### Set Variables action fields

Use the Set Variables - xxx panel to set, increment, or decrement the value of one or more SHARED variables. This panel is available to all event types.

The following figure shows an example of the Action: Set Variables - MSG panel.

#### Figure 163: Action: Set Variables panel

<table>
<thead>
<tr>
<th>Variable-Name</th>
<th>Variable-Value(functions/equations allowed)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Press END to return to AV with changes or CANCEL to cancel changes

The following table describes the fields on this panel.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Console Name</td>
<td>Specify the name of MVS Console to send WTO. If not specified, the WTO is sent to the console or consoles defined by Route Codes.</td>
</tr>
<tr>
<td>Delay Action</td>
<td>Specify the number of seconds to wait before the Rule issues this message. Valid values 1 through 99999. If the BBI-SS PAS is stopped for any reason before MainView AutoOPERATOR can issue the messages that are scheduled with a Delay, the delayed messages are not re-issued when the BBI-SS PAS restarts. Daylight Saving Time considerations: For more information about how changing the local MVS time affects specifying a delayed action, see Appendix, &quot;Managing daylight saving time and MainView AutoOPERATOR time-initiated automation&quot; in the MainView AutoOPERATOR Basic Automation Guide, Volume 2.</td>
</tr>
</tbody>
</table>
Table 30: Issue Message panel

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable Name</td>
<td>Specify the name of SHARED variable to be set. The maximum length for this field is 33 characters. SHARED variables can be used in the variable name. You can also use substring specifications; for more information about how to use substringing with variable names, refer to “Variable substringing” on page 166. You can specify up to 10 variables and each variable is independent of any others that might be specified on the panel. The order in which the variables will be resolved when the Rule fires is unpredictable. You can set, increase the amount, or decrease the amount for the value of a SHARED variable. A variable's numerical value may be increased or decreased by specifying +n or -n, where n is a number.</td>
</tr>
<tr>
<td>Variable Value</td>
<td>Specify the value used to set variable. The maximum length of this field is 17 characters. You can use this field to assign a new value to the variable (by using a variable name or an unsigned constant) or modify a numerical 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, a slight delay might occur before the value of the variables are set. In the following example, the first variable, SYSABENDS, is a counter and it will be increased by one. The next two variables use variable &amp;WORD3 to help form the name (assuming a value of &quot;00B123&quot;) to create variable names of ERRB123 and CNTB123.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Variable Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) SYSABENDS_______________________</td>
<td>= +1_______________</td>
</tr>
<tr>
<td>2) ERR&amp;WORD3:3:4____________________</td>
<td>= HIT______________</td>
</tr>
<tr>
<td>3) CNT&amp;WORD3:3:4____________________</td>
<td>= +1_______________</td>
</tr>
</tbody>
</table>

Alert Actions I and II fields

The following figure shows an example of the first ALERT Action(s) I panel.

BMC Software ---------- Alert Action(s) I - MSG  MAINVIEW AutoOPERATOR
COMMAND ===>                                                 TGT --- A0A0
Rule-set === RULTEST1 Rule-id === FT407IA
Function ===>                                   (ADD, DELETE, DELETEQ)
Key      ===>                                   |
Text     ===>                                   |
Queues   ===>                                   Alert Queue Name(s)
Priority ===>                                   (CRIT, MAJ, WARN, INFO, CLEAR)
Color    ===>                                   RED, PINK, YEL, DKBL, LTBL, GRE, WHI
PCMD     ===>                                   |
System   ===>                                   |
* Exec   ===>                                   Follow-up EXEC
Help     ===>                                   Associated HELP Panel
Targets  ===>                                   Target System
The following table describes the fields on this panel.

### Table 31: Alert Action(s) I panel field descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Function</strong></td>
<td>Specify the function to be performed. This field is used in conjunction with the <strong>Queue</strong> field where possible values are:</td>
</tr>
<tr>
<td>■ <strong>ADD</strong>: adds an ALERT to the specified queue</td>
<td></td>
</tr>
<tr>
<td>■ <strong>DELETE</strong>: deletes an ALERT from the specified queue</td>
<td></td>
</tr>
<tr>
<td>■ <strong>DELETEQ</strong>: deletes the ALERT and the ALERT queue</td>
<td></td>
</tr>
<tr>
<td>■ <strong>MODIFY</strong>: available with the ALRT event type only</td>
<td></td>
</tr>
<tr>
<td>For the ALRT event type, using MODIFY allows you to change any ALERT specifications except Key, Queue, and Target. For more information, refer to “(optional) Modifying an ALERT in a Rule: using the MODIFY function” on page 100.</td>
<td></td>
</tr>
<tr>
<td><strong>Key</strong></td>
<td>Specify 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 <em>cannot be modified</em> by using the MODIFY function for an ALRT-initiated Rule.</td>
</tr>
<tr>
<td><strong>Text</strong></td>
<td>Specify 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 &amp;WORDn variables, or literals.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Queues</td>
<td>Specify a 33-byte field specifying the names of the ALERT queues. This field is used in conjunction with the <strong>Function</strong> 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: &amp;IMFOJOB:::MAIN. This field cannot be modified by using the MODIFY function for an ALRT-initiated Rule.</td>
</tr>
<tr>
<td>Priority</td>
<td>Specify the priority of the ALERT. Valid values are CRITICAL, MAJOR, MINOR, WARNING, INFORMATIONAL, and CLEARING. The default is INFO.</td>
</tr>
<tr>
<td>Color</td>
<td>Specify the color of the alert text. This field overrides the color assigned by Alert Priority. Valid colors are RED, PINK, YELLOW, DKBLUE, LTBLUE, GREEN, and WHITE.</td>
</tr>
<tr>
<td>PCMD</td>
<td>Specify 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 that can expand to 252 bytes.</td>
</tr>
<tr>
<td>System</td>
<td>Specify the SYSTEM command after the PCMD is issued. This action passes control back to the target. Valid values are Yes and No; Yes is the default.</td>
</tr>
<tr>
<td>Exec</td>
<td>Refer to “REXX Exec or CLIST panel” on page 308 for more information.</td>
</tr>
<tr>
<td>Help</td>
<td>Specify the name of the help panel associated with the ALERT that can be accessed from the ALERT Management Facility.</td>
</tr>
<tr>
<td>Targets</td>
<td>Specify 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 by using the MODIFY function for an ALRT-initiated Rule.</td>
</tr>
<tr>
<td>Udata</td>
<td>Specify any desired user data string.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Origin</td>
<td>Specify a user-defined name of the origin to assign to the ALERT. When you specify an Origin and a follow-up EXEC that includes performing certain actions (such as issuing the IMFEXEC RES or IMFEXEC CICS commands), you must ensure that the Origin resolves to a user ID that has been authorized in your security definitions to issue these commands. For example, you might create a Rule that traps an event in a CICS address space and creates an ALERT with a follow-up EXEC. The follow-up EXEC has the &quot;IMFEXEC RES ST jobname&quot; command in it. When AutoOPERATOR passes control to SYSPROG, the value specified in Origin for the ALERT is passed to SYSPROG as the user attempting the action. If that Origin value is not an authorized user ID defined in your security definitions to issue SYSPROG commands, the SYSPROG command will fail. When an Origin is not specified, the Origin value defaults the jobname of the event that created the ALERT.</td>
</tr>
<tr>
<td>User</td>
<td>Specify the user assigned to the ALERT. See also the description for the Origin field</td>
</tr>
<tr>
<td>Alarm</td>
<td>Specify whether to emit an audible alarm.</td>
</tr>
<tr>
<td>Publish</td>
<td>Specify whether an ALERT is published and how it is published to connected PATROL EM workstations or to BMC Impact Manager cells that have been customized to receive ALERTs through the General Message Exchange (GME). Settings for GME are defined in BBPARM member AAOGME00. 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 AAOALxx. 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 MainView AutoOPERATOR are documented. For more information about implementing MainView AutoOPERATOR connection to BMC Impact Manager cells, refer to Chapter 17, &quot;Implementing the MainView AutoOPERATOR Interface to BMC Impact Manager&quot; in the MainView AutoOPERATOR Customization Guide.</td>
</tr>
</tbody>
</table>

### Alert Action(s) II fields

The following figure shows an example of the second ALERT Action(s) Specification panel.

```
BMC Software -------- Alert Action(s) II - MSG --------- MAINVIEW AutoOPERATOR
COMMAND ===>                                                 TGT  --- AOAO
Rule-set === RULCICS     Alert    Rule-id  === FT407IA
Auto Delete             ===>                                      Yes/No
Retain                ===>                                      Yes/No
Escalate Direction    ===>                                      Up/Down
```
This following describes the fields on this panel:

### Table 32: Alert Action(s) II panel

<table>
<thead>
<tr>
<th>ALERT Actions field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto Delete</td>
<td>Allows you to specify that for a message, when a delete operator action message is performed (DOMed), the ALERT associated with the message is also deleted automatically. In other words, when you create Rules for action messages that issue ALERTs, you can specify that the ALERT is deleted automatically 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, specify <strong>No</strong> in the <strong>Auto Delete</strong> field. <strong>Note:</strong> You should use this field only for messages that will be DOMed Usually only WTORs and action messages are DOMed. When Auto Delete is used, MainView AutoOPERATOR uses a control block in CSA to store the message's DOM ID. When the message is DOMed, MainView AutoOPERATOR frees this CSA storage. If Auto Delete is used and the WTO is never DOMed, MainView AutoOPERATOR continues to maintain this storage in CSA until MainView AutoOPERATOR is restarted with START=COLD.</td>
</tr>
</tbody>
</table>
### Alert Actions Field

<table>
<thead>
<tr>
<th>Alert Actions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retain</td>
<td>Allows you to specify an ALERT will be retained across BBI-SS PAS restarts (both cold and warm restarts) and MVS IPLs. Specifies whether the ALERT should be created as a nonvolatile ALERT. Possible values are YES</td>
</tr>
<tr>
<td>Escalate Direction</td>
<td>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; <em>they are mutually exclusive.</em> Possible values for this parameter are:</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Up</strong>: specifies that the ALERT will be upgraded in severity when the time interval elapses</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Down</strong>: specifies that the ALERT will be downgraded in severity when the time interval elapses</td>
</tr>
<tr>
<td></td>
<td>The default value is Up. When you use the Escalate Direction field, you must specify <em>at least one time interval</em> in the <strong>Interval</strong> field.</td>
</tr>
</tbody>
</table>
## ALERT Actions field

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Interval</strong></td>
</tr>
</tbody>
</table>
| **Disposition** | Used with the **Escalate Direction** field. Allows you to specify what will happen to an 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. |
| **Final Act ()** | Allows you to specify a final action to be taken when the ALERT reaches its final priority level. Action types that you can specify are as follows:  
  - **EXEC**: Schedule an EXEC to run.  
  - **MVS, BBI, CICS, IMS, IMP, MQ, NV, and TOM**: Execute a command of this type. Enter a question mark to display the Alert Escalation Final Action Type panel. From this panel, you can select the action and display an additional panel where you can enter the action and associated keywords. Refer to “ALERT Final Actions panels” on page 327 for more information. |

### ALERT Final Actions panels

The following figure shows an example of the Alert Escalation Final Action Type panel that is displayed when you enter a question mark in the **Final Act ()** field.
Use this panel to select the final action that an ALERT should take when it reaches its final priority level and display additional panels where you specify the action and any additional associated keywords.

**Figure 164: Alert Escalation Final Action Type panel**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>REXX Exec or CLIST</td>
</tr>
<tr>
<td>MVS</td>
<td>MVS/JES command</td>
</tr>
<tr>
<td>BBI</td>
<td>BBI Control command</td>
</tr>
<tr>
<td>IMS</td>
<td>IMS command</td>
</tr>
<tr>
<td>IMP</td>
<td>IMS Plex command</td>
</tr>
<tr>
<td>CICS</td>
<td>CICS command</td>
</tr>
<tr>
<td>MQ</td>
<td>WebSphere MQ command</td>
</tr>
<tr>
<td>NV</td>
<td>NetView command</td>
</tr>
<tr>
<td>TOM</td>
<td>Total Object Manager</td>
</tr>
</tbody>
</table>

Command:  
Press ENTER to continue filling in command. END to return to Alert Actions with selected command type, CANCEL to cancel selection.

When you select EXEC, the REXX Exec or CLIST panel is displayed (Figure 165 on page 328). Refer to “REXX Exec or CLIST panel” on page 308 for information about the fields on this panel.

**Figure 165: REXX Exec or CLIST panel**

Priority and Delay keywords apply to all execs

<table>
<thead>
<tr>
<th>Priority</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Norm, High, Hot, First, Single, Xsingle; default is Norm)</td>
<td></td>
</tr>
<tr>
<td>(Seconds to wait before scheduling the exec)</td>
<td></td>
</tr>
</tbody>
</table>

Note: Total length of all input values must be less than 56 characters.
Press END to return with exec(s), CANCEL to cancel selection.

You can select one of the command types on Figure 164 on page 328; Figure 166 on page 328 shows an example of the MVS Command(s) from BBI-SS panel that is displayed when you select MVS. Refer to “Select Command Type panel” on page 311 for information about the fields on this panel.

**Figure 166: MVS Command(s) from BBI-SS panel**
The following table describes the additional keywords that are available from the panels shown in preceding panels.
### Table 33: Additional fields for different command types

<table>
<thead>
<tr>
<th>Command Type</th>
<th>Field names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>EXEC/Parms</td>
<td>Specify that an EXEC is scheduled as the final action. The EXEC is scheduled to execute in the BBI-SS PAS address space. The number of EXECs and additional parameters is limited to 56 characters. If you specify multiple EXEC names, the EXEC can execute in any order, and does not necessarily execute in the order that you specified on the panel. If you specify values in the <strong>Priority</strong> and <strong>Delay</strong> fields, these values apply to all of the specified EXECs.</td>
</tr>
<tr>
<td>Priority</td>
<td></td>
<td>Specify the priority assigned to the EXEC. Valid values are normal (N), high (H), hot (T), first (F), Single (S), and Xsingle (X).</td>
</tr>
<tr>
<td>Delay</td>
<td></td>
<td><strong>Note:</strong> The Delay keyword can be specified when you enter any valid action. For the BBI and IMS actions, the Delay keyword is the only additional keyword available. Specify 1 to 99999 seconds to wait before the final action is taken (either schedule an EXECs or issue a commands). When you use this keyword, all of the listed EXECs or commands are delayed, and the order in which they are executed is unpredictable. If the BBI-SS PAS is stopped for any reason before MainView AutoOPERATOR can run the commands that are scheduled with a Delay, the delayed commands are not re-scheduled when the BBI-SS PAS restarts. <strong>Daylight Saving Time considerations:</strong> For more information about how changing the local MVS time affects specifying a delayed action, see Appendix, &quot;Managing daylight saving time and MainView AutoOPERATOR time-initiated automation&quot; in the <em>MainView AutoOPERATOR Basic Automation Guide, Volume 2.</em></td>
</tr>
<tr>
<td>Command Type</td>
<td>Field names</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>MVS</td>
<td>Console</td>
<td>Specify the name of a console that will issue the MVS commands; the console does not have to be active for the command to be issued.</td>
</tr>
<tr>
<td>CART</td>
<td></td>
<td>Command and Response Token (CART) field specifies a token for an MVS command issued by a Rule.</td>
</tr>
<tr>
<td>IMP</td>
<td>Route</td>
<td>Specify the Operations Manager (OM) Clients (also referred to as IMS control regions) in the IMSplex where the commands are issued. If the Route is not specified, the commands are sent to all IMS regions in the IMSPLEX that have registered with the IMS OM to process commands. You can specify up to seven OM client names.</td>
</tr>
<tr>
<td>CICS</td>
<td>Target</td>
<td>Specify the name of the target CICS where the commands are issued.</td>
</tr>
<tr>
<td>MQ</td>
<td></td>
<td>Refer to <em>MainView AutoOPERATOR for WebSphere MQ Installation and User Guide</em> for information about using additional keywords when issuing WebSphere MQ commands from a Rule.</td>
</tr>
<tr>
<td>NV</td>
<td>Jobname</td>
<td><em>(optional)</em> Specify the jobname of the NetView address space where the commands are issued. The default address space is the NetView address space that generated the event. For non-NetView events, the default is the jobname that is specified in the NETVIEW parameter of BBPARM member BBISSIPx.</td>
</tr>
<tr>
<td>TOM</td>
<td>(none)</td>
<td>There are no extra fields are available for this command type.</td>
</tr>
</tbody>
</table>

---

**Event Action — BEIM fields**

The following table lists the options that are available when you write a Rule to send MainView AutoOPERATOR events to a BMC Impact Manager cell.
<table>
<thead>
<tr>
<th>Event Action BEIM field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send Event</td>
<td>Specify whether or not Rule will send the slot and class data to the BMC Impact Manager cell when the Rule fires. Use this option when you want to create the Rule action but do not yet want to send the BMC Impact Integration for z/OS data to the cell. Specify one of the following values (abbreviations are permitted):</td>
</tr>
<tr>
<td></td>
<td>■ Yes</td>
</tr>
<tr>
<td></td>
<td>■ No</td>
</tr>
<tr>
<td>Journal Details</td>
<td>Specify that the event details are written to the BBI Journal including</td>
</tr>
<tr>
<td></td>
<td>■ the Rule Set and Rule ID that created the event</td>
</tr>
<tr>
<td></td>
<td>■ the message text of the event</td>
</tr>
<tr>
<td></td>
<td>■ the severity and Component Alias of the event</td>
</tr>
<tr>
<td></td>
<td>■ the targets</td>
</tr>
<tr>
<td></td>
<td>■ the Component Alias name</td>
</tr>
</tbody>
</table>

The default setting for this field is NO (do not send messages to the BBI Journal). If you specify BIIZJRNL=Y in BBPARM member AAOPRMxx, messages for this event are written to the BBI Journal.

If you specify BIIZJRNL=N in BBPARM member AAOPRMxx, but you specify Y in the **Journal Details** field, messages for this event only will be written to the BBI Journal.
<table>
<thead>
<tr>
<th>Event Action BEIM field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target(s)</td>
<td>Specify that the Rule will send events to BMC Impact Manager cells. You can specify up to six target cell names that can be up to 16-characters long each and can contain variables. If no targets are specified, the event is sent to all cells. Use the TGTNAME parameter in BBPARM member AAOGME.xx to assign cells names. More than one cell can be assigned to the same target name. To use wildcards in the target name: ▪ A plus sign matches a single character. ▪ An asterisk (<em>) indicates the target name should match an unlimited number of characters. For example, the target name IMPACT</em> matches all cell names that begin with the name IMPACT.</td>
</tr>
<tr>
<td>Event Action BEIM field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Event class</td>
<td>Specify a class to which all of the slots specified on the Event Action BEIM panel belong. The class must be defined as a subclass of the default class which inherits all the slots from the default class (MV_EVENT_xxxx, where xxxx is the Rule or event type for this event). If this value is not specified, the slots will be part of the default event class for the event (which is based on the type of Rule). Refer to the Help panels for examples of how to use this field. When you specify an event class on this panel, you must also define the same class to the BMC Impact Manager cell as a subclass of the default class for this event. The event class name should meet the following guidelines:</td>
</tr>
<tr>
<td></td>
<td>■ 1 to 32 character class name (2 to 33 characters if the name begins with an ampersand, denoting a variable)</td>
</tr>
<tr>
<td></td>
<td>■ valid characters include all upper and lower case alphabetic characters as well as numerics, national characters and most special characters</td>
</tr>
<tr>
<td></td>
<td>■ lower case characters are not converted to uppercase (all characters are accepted exactly as received)</td>
</tr>
<tr>
<td></td>
<td>You can use variables but they must meet the following requirements:</td>
</tr>
<tr>
<td></td>
<td>■ begin with an ampersand</td>
</tr>
<tr>
<td></td>
<td>■ be 1 - 32 characters in length</td>
</tr>
<tr>
<td></td>
<td>■ be either a LOCAL or SHARED variable</td>
</tr>
<tr>
<td>Event Classes:</td>
<td>The following lists the default BMC Impact Manager event class and the corresponding Rule event type for which slots are defined for an event on the cell.</td>
</tr>
<tr>
<td>MV_EVENT_ALRM (ALRM)</td>
<td>MV_EVENT_JRNL (JRNL) MV_EVENT_VAR (VAR)</td>
</tr>
<tr>
<td>MV_EVENT_ALERT (ALERT)</td>
<td>MV_EVENT_EXT (EXT) MV_EVENT_TIME (TIME)</td>
</tr>
<tr>
<td>MV_EVENT_DB2 (DB2)</td>
<td>MV_EVENT_MSG (MSG) MV_EVENT_CMD (CMD)</td>
</tr>
<tr>
<td>MV_EVENT_HWTO (HWTO)</td>
<td>MV_EVENT_IMP (IMP) MV_EVENT_IMS (IMS)</td>
</tr>
<tr>
<td>MV_EVENT_CICS (CICS)</td>
<td>MV_EVENT_MQS_EVENT (MQS) MV_EVENT_NVC (NVC)</td>
</tr>
<tr>
<td>MV_EVENT_NVM (NVM)</td>
<td>This field is optional and has no default.</td>
</tr>
<tr>
<td>Event Action BEIM field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Severity</strong></td>
<td>Specify the severity of the event. This Rule sends this value to the BMC Impact Manager cell along with any other specified fields. This field is required; the default is WARNING and the following values are valid:</td>
</tr>
<tr>
<td></td>
<td>■ UNKNOWN</td>
</tr>
<tr>
<td></td>
<td>■ INFO</td>
</tr>
<tr>
<td></td>
<td>■ MINOR</td>
</tr>
<tr>
<td></td>
<td>■ CRITICAL</td>
</tr>
<tr>
<td></td>
<td>■ OK</td>
</tr>
<tr>
<td></td>
<td>■ WARNING</td>
</tr>
<tr>
<td></td>
<td>■ MAJOR</td>
</tr>
<tr>
<td></td>
<td>■ DOWN</td>
</tr>
<tr>
<td></td>
<td>You can also specify a LOCAL or SHARED variable. In addition, the variable must:</td>
</tr>
<tr>
<td></td>
<td>■ begin with an ampersand</td>
</tr>
<tr>
<td></td>
<td>■ be 1 - 32 characters in length</td>
</tr>
<tr>
<td></td>
<td>■ be defined when the Rule fires</td>
</tr>
<tr>
<td>Event Action BEIM field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Component Alias</td>
<td>Specify a value in mc_smc_alias slot. The event class name should meet the following guidelines:</td>
</tr>
<tr>
<td></td>
<td>■ Names contains 1 to 55 characters (2 to 55 characters if the name begins with an ampersand, denoting a variable).</td>
</tr>
<tr>
<td></td>
<td>■ Valid characters include all upper and lowercase alphabetic characters as well as numerics, national characters and most special characters.</td>
</tr>
<tr>
<td></td>
<td>■ Lowercase characters are not converted to uppercase (all characters are accepted exactly as received).</td>
</tr>
<tr>
<td></td>
<td>You can use variables but they must meet the following requirements:</td>
</tr>
<tr>
<td></td>
<td>■ begin with an ampersand</td>
</tr>
<tr>
<td></td>
<td>■ be 1 - 32 characters in length</td>
</tr>
<tr>
<td></td>
<td>■ be either a LOCAL or a SHARED variable</td>
</tr>
<tr>
<td></td>
<td>■ must be defined when the Rule fires</td>
</tr>
<tr>
<td></td>
<td>If you own the BMC product Discovery \textit{for} z/OS, and you specified a VSAM data set name in BBPARM member AAOPRMxx with the DZDSN parameter, you can specify the variable <code>&amp;!QCIA.QSERVER.IMFCIJOB</code> in this field. Refer to the \textit{MainView AutoOPERATOR Customization Guide} for more information about the DZDSN parameter.</td>
</tr>
</tbody>
</table>
Event Action BEIM field | Description
--- | ---
Event Slots and Slot Value | Specify additional data that you consider relevant to the particular event being sent to a BMC Impact Manager cell. These fields are optional and have no default. Both Event Slot name and a corresponding Slot Value must be specified together on the panel; you can specify
- a 1- to 32 character slot name
- a 1-38 character slot value

The msg slot is used by the BIM cell to populate the event text. The msg must be specified in lowercase.

Event Slot and Event Value fields are case sensitive. If a variable name is specified it must be in uppercase. For example: &WORD3.

Refer to the MainView AutoOPERATOR BMC Impact Integration for z/OS User Guide for information about slot and class names.

You can also specify a LOCAL or SHARED variable. In addition, the variable must:
- begin with an ampersand
- be 1-32 characters in length
- be defined when the Rule fires

---

Alert Action III BEIM - panel fields

The following table lists the fields that are available when you write a Rule to do the following actions:

- **Add a new ALERT:** When you specify that a Rule sends an ALERT to a BMC Impact Manager cell, the Alert Action III BEIM panel provides input fields where you can enrich the ALERT information.

- **Modify an ALERT:** When an ALERT-initiated Rule modifies an existing ALERT, the ALERT being modified can also contain enriched information from the Alert Action III BEIM panel.
### Table 35: Alert Action III BEIM panel fields

<table>
<thead>
<tr>
<th>Alert Action III BEIM field</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Journal Details**        | Specify that the ALERT details are written to the BBI Journal including  
                                  ■ the Rule Set and Rule ID that created the event  
                                  ■ the message text of the event  
                                  ■ the severity and Component Alias of the event  
                                  ■ the targets  
                                  ■ the Component Alias name  
                                  The default setting for this field is NO (do not send messages to the BBI Journal). If you specify BIIZJRNL=Y in BBPARM member AAOPRMxx, messages for this event are written to the BBI Journal. If you specify BIIZJRNL=N in BBPARM member AAOPRMxx, but you specify Y in the **Journal Details** field, messages for this event only will be written to the BBI Journal. |
| **Target(s)**              | Specify that the Rule will send events to BMC Impact Manager cells.  
                                  You can specify up to six target cell names that can be up to 16-characters long each and can contain variables. If no targets are specified, the ALERT is sent to all cells. Use the TGTNAME parameter in BBPARM member AAOGMExx to assign cell names. More than one cell can be assigned to the same target name.  
                                  To use wildcards in the target name:  
                                  ■ A plus sign matches a single character.  
                                  ■ An asterisk (*) indicates a the target name should match an unlimited number of characters.  
                                  For example, the target name IMPACT* matches all cell names that begin with the name IMPACT.  
                                  Additional considerations for sending ALERTs to cells follow:  
                                  ■ If a target is not specified, the ALERT is published by default to all connected cells  
                                  ■ An ALERT should always be published to the same cells  
                                  ■ An ALERT should always use the same PUBLISH parameter; if you choose to use ADD, use it consistently for all the ALERTS or use REPLACE consistently. Do not use ADD for some ALERTS and REPLACE for others |
**Event class**

Specify a class to which all of the slots specified on the Event Action BEIM panel belong.

The class must be defined as a subclass of the default class which inherits all the slots from the default class (MV_EVENT_xxxx, where xxxx is the Rule or event type for this event).

If this value is not specified, the slots will be part of the default event class for the event (which is based on the type of Rule). Refer to the Help panels for examples of how to use this field.

When you specify an event class on this panel, you must also define the same class to the BMC Impact Manager cell as a subclass of the default class for this event.

The event class name should meet the following guidelines:

- 1 to 32 character class name (2 to 33 characters if the name begins with an ampersand, denoting a variable)

- valid characters include all upper and lower case alphabetic characters as well as numerics, national characters and most special characters

- lower case characters are not converted to uppercase (all characters are accepted exactly as received)

You can use variables but they must meet the following requirements:

- begin with an ampersand

- be 1 - 32 characters in length

- be either a LOCAL or SHARED variable

**Event Classes**: The following lists the default BMC Impact Manager event class and the corresponding Rule event type for which slots are defined for an event on the cell.

- MV_EVENT_ALRM (ALRM)
- MV_EVENT_JRNL (JRNL)
- MV_EVENT_VAR (VAR)
- MV_EVENT_ALERT (ALERT)
- MV_EVENT_EXT (EXT)
- MV_EVENT_TIME (TIME)
- MV_EVENT_DB2 (DB2)
- MV_EVENT_MSG (MSG)
- MV_EVENT_CMD (CMD)
- MV_EVENT_IMS (IMS)
- MV_EVENT_CICS (CICS)
- MV_EVENT_MQS_EVENT (MQS)
- MV_EVENT_NVC (NVC)
- MV_EVENT_NVM (NVM)

This field is optional and has no default.
### Alert Action III BEIM field

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Severity</td>
<td>Specify the severity of the event. This Rule sends this value to the BMC Impact Manager cell along with any other specified fields.</td>
</tr>
<tr>
<td></td>
<td>This field is required; the default is WARNING and the following values are valid:</td>
</tr>
<tr>
<td></td>
<td>- UNKNOWN</td>
</tr>
<tr>
<td></td>
<td>- INFO</td>
</tr>
<tr>
<td></td>
<td>- MINOR</td>
</tr>
<tr>
<td></td>
<td>- CRITICAL</td>
</tr>
<tr>
<td></td>
<td>- OK</td>
</tr>
<tr>
<td></td>
<td>- WARNING</td>
</tr>
<tr>
<td></td>
<td>- MAJOR</td>
</tr>
<tr>
<td></td>
<td>- DOWN</td>
</tr>
<tr>
<td></td>
<td>You can also specify a LOCAL or SHARED variable. In addition, the variable must</td>
</tr>
<tr>
<td></td>
<td>- begin with an ampersand</td>
</tr>
<tr>
<td></td>
<td>- be 1 - 32 characters in length</td>
</tr>
<tr>
<td></td>
<td>- be defined when the Rule fires</td>
</tr>
<tr>
<td>Alert Action III BEIM field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Component Alias</td>
<td>Specify a value in the mc_smc_alias slot. The event class name should meet the following guidelines:</td>
</tr>
<tr>
<td></td>
<td>■ 1 to 55 character class name (2 to 55 characters if the name begins with an ampersand, denoting a variable)</td>
</tr>
<tr>
<td></td>
<td>■ valid characters include all upper and lower case alphabetic characters as well as numerics, national characters and most special characters</td>
</tr>
<tr>
<td></td>
<td>■ lower case characters are not converted to uppercase (all characters are accepted exactly as received)</td>
</tr>
<tr>
<td></td>
<td>You can use variables but they must meet the following requirements:</td>
</tr>
<tr>
<td></td>
<td>■ begin with an ampersand</td>
</tr>
<tr>
<td></td>
<td>■ be 1 - 32 characters in length</td>
</tr>
<tr>
<td></td>
<td>■ be either a LOCAL or SHARED variable</td>
</tr>
<tr>
<td></td>
<td>■ must be defined when the Rule fires</td>
</tr>
</tbody>
</table>

If you own the BMC product Discovery for z/OS, and you specified a VSAM data set name in BBPARM member AAOPRMxx with the DZDSN parameter, you can specify the variable &!QCIA.QSERVER.IMFCIJOB in this field. Refer to the MainView AutoOPERATOR BMC Impact Integration for z/OS User Guide for more information about the DZDSN parameter.
<table>
<thead>
<tr>
<th>Alert Action III BEIM field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Event Slots and Slot Value | Specify additional data that you consider relevant to the particular event being sent to a BMC Impact Manager cell. These fields are optional and have no default. Both Event Slot name and a corresponding Slot Value must be specified together on the panel; you can specify  
- a 1- to 32 character slot name  
- a 1-38 character slot value  
Refer to the *BMC Impact Integration for z/OS User Guide* for information about slot and class names. You can also specify a LOCAL or SHARED variable. In addition, the variable must  
- begin with an ampersand  
- be 1-32 characters in length  
- be defined when the Rule fires |
Printing Rules and Rule Sets

You can print hardcopy versions of MainView AutoOPERATOR Rules and Rule Sets by using the BBSAMP member RULPRINT JCL in a batch job.

This job prints selected Rules or Rule Sets that are defined in the BBIPARM DD data set concatenation.

You can specify that the output is written to SYSOUT or to a data set. When writing to a data set, the SYSPRINT DD requires the following DCB specification:

```
DCB=(RECFM=FB,LRECL=133,BLKSIZE=13300)
```

To use the RULPRNT JCL

1. Make a copy of BBSAMP(RULEPRNT).

2. Modify the JCL and parameters as directed within the comments of BBSAMP(RULEPRNT).

3. Submit the job.

The following table describes the available RULPRNT input parameters that you can use to specify which Rules or Rule Sets should be printed.
## Table 36: Reporting options

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS</td>
<td>RULESET=</td>
</tr>
<tr>
<td>RULESET=AAORULAA</td>
<td>prints a single Rule Set named AAORULAA.</td>
</tr>
<tr>
<td>RULESET=(AAORULAA,AAORULAB,AAORULAC)</td>
<td>prints the three Rule Sets named AAORULAA, AARULAB, and AARULAC.</td>
</tr>
<tr>
<td>RULESET=AAORULA*</td>
<td>prints all Rule Sets whose names start with AAORULA.</td>
</tr>
<tr>
<td>RULESET=RUL+++A*</td>
<td>prints all Rule Sets whose names start with RUL, followed by any three characters, followed by A. (The name can either end with the A or include additional characters.)</td>
</tr>
<tr>
<td>RULESET=*</td>
<td>prints all of the Rule Sets.</td>
</tr>
<tr>
<td>RU</td>
<td>RULE=</td>
</tr>
<tr>
<td>RULE=MYRULE1</td>
<td>prints a single Rule named MYRULE1.</td>
</tr>
<tr>
<td>RULE=(MYRULE1,MYRULE2,MYRULE3)</td>
<td>prints the three Rules named MYRULE1, MYRULE2, and MYRULE3.</td>
</tr>
<tr>
<td>RULE=MYRULE*</td>
<td>prints all of the Rules that start with MYRULE.</td>
</tr>
<tr>
<td>RULE=MY+++E*</td>
<td>prints all of the Rules that start with the characters MY, have any 3 characters followed by E, and after which can have additional characters in the name.</td>
</tr>
<tr>
<td>RULE=(MY*,YOUR*,ANY*)</td>
<td>prints all Rules that start with the characters MY, YOUR, or ANY.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| **TY | TYPE=** | optional parameter that specifies printing only the Rules for indicated event type (or types)  
If you do not specify a value, the event type attribute is not considered.  
Valid values are any valid event types (for example, MSG, ALRT, JRNL, CICS, and so on).  
If specifying multiple event types, separate them with commas, and enclose them within parentheses. Examples follow:  
- TYPE=MSG prints Rules for the MSG event type.  
- TYPE=(MSG, JRNL, ALRT) prints Rules for event types MSG, JRNL, and ALRT. |
| **ST | STAT=** | optional parameter that specifies printing only the Rules that match the specified state  
If you do not specify a value, the Rule's state is not considered.  
Examples follow:  
- STAT=ENA prints Rules that are ENABLED.  
- STAT=DIS prints Rules that are DISABLED. |
| **H | HAS=** | optional parameter that specifies printing only Rules that match the criteria specified  
If you do not specify a value, the Rule's contents are not considered.  
Valid values are  
- SV (for Set Variables specifications, standard or advanced)  
- AV (for Variable Dependencies specifications, standard or advanced)  
- AA (ALERT Actions specifications). Examples follow:  
Examples follow:  
- HAS=SV means the Rule must have Set Variables specifications.  
- HAS=(SV, AA) means the Rule must have both Set Variables and ALERT Actions specifications. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>UDATE=</td>
</tr>
</tbody>
</table>
Index

@ sign, IMS generic parameter processing 311

A
A1 primary command, Rule Processor Detail Control panel 73, 84
AA primary command, Rule Processor Detail Control panel 73, 84
Acct Info field, Rule Processor Selection Criteria panel 279
Action
  Advanced Set Variables (AAV) panel 92
Action field, Rule Processor Detail Control panel
  Criteria match rate
    Rule setting
      then status field 76
action specification, Rule Processor application 25, 90
Activated at field, Event Activity Statistics application 47
ADD primary command, Rule Processor application
  Automation Control Panel 210
  naming conventions 210
  Rule Set Overview panel 235
Advanced Variable Dependencies (ASV) panel 88
After command, Rule Processor application
  Automation Control panel 217
  Rule Set Overview panel 235
Alarm field
  Rule Processor application, ALERT Action(s) Specification panel 321
Alarm ID field, Rule Processor Selection Criteria panel 279
ALERT Action(s) Specification, Rules fields 321
  sample panel 95
ALL Automation Strategy, Rule Processor application 218

ALRT
  event type 35
  event type, variables 183
AMEORGN 183
AMFALARM 98, 183
AMFCOLOR 98, 183
AMFCURIN 98
AMFEDIR 98, 183
AMFEDISP 98, 183
AMFEXEC 98, 183
AMFEINT 98, 183
AMFEVENT 98, 183
AMFEXEC 98, 183
AMFHELP 98, 183
AMFIDATE 98, 183
AMFITIME 98, 183
AMFKEY 183
AMFORGN 98, 183
AMFPUB 98, 183
AMFPUBS 98
AMFQUEUE 98, 183
AMFTPUB 98, 183
AMFTGT 98, 183
AMFPUBS 98
AMFUSER 98, 183
AND/OR logic, Variable Dependencies panel 90
Author field
  Rule Processor Detail Control panel 78
  Rule Set Overview panel 229
Auto Delete, ALERT Action(s) panel field 324
Automation Control application, Rule Processor application 31, 32
Automation Control panel
  Rule Processor application
    grouping Rules 214
Automation Control panel, Rule Processor application
  Automation Status 209
  fields 213
  line commands 217
  primary commands 210
  sample panel 207
Automation Statistics
  Rules 213
Automation Status field, Automation Control panel 213
automation strategy
  Rule Processor application 31, 32
  Rules 218
Automation Strategy field, Automation Control panel 213
AutoOPERATOR
  features 13
  introduction 13, 19
  overview figure 13
  Primary Option Menu 17
  Rule Processor application 21
  Rules 238

B
Before command, Rule Processor application
  Automation Control panel 217
  Rule Set Overview panel 235
BMC EM/SIM events
  Rule action specification fields
    Send Event 331
    Severity 331, 337
BRC
  event type 35
BRC event variables 181
BRC, event type 49
built-in functions 185
  Rules functions 185

C
Changed field
  Rule Set Overview panel 229
CICa
  event type 35
CICa, event type 50
CICS
  event type 35
CICS TDQ field, Rule Processor Selection Criteria panel 279
Cmd field, Rule Processor Action Specification panel 311
CMD, event type 35
CMDSHOW primary command, Rule Processor application
  Automation Control Panel 210
  naming conventions 210
Code field
  Rule Processor Detail Control panel 78
  Rule Set Overview panel 229
Color field
  ALERT Action(s) Specification panel 321
  comparison operators 187
  Rules functions 187
  connector operators 188
  Rules functions 188
Console Name field
  Rule Processor Action Specification panel 298
  Rule Processor Selection Criteria panel
    CMD-Initiated Rule 279
    JES3-Initiated Rule 279
    MSG-Initiated Rule 279
Context field, Rule Processor Selection Criteria panel 279
  conventions, documentation 12
Copy command, Rule Set Overview panel 235
Count field, Event Activity Statistics application 47
  creating Rules
    See Rule Processor application 21

D
Date field
  Rule Processor application
    Automation Control panel 214
DB2
  event type 35, 51
Delete command
  Rule Set Overview panel 235
  delete operator message 324
  deleting
    Rule Set 238
Desc codes field, Rule Processor Selection Criteria panel
  JES3-Initiated Rule 279
  MSG-Initiated Rule 279
Description field
Rule Processor Detail Control panel 78
Rule Set Overview panel 229
Descriptor codes field
Rule Processor Action Specification panel 298
Disable command
Rule Processor application
Automation Control panel 217
Rule Set Overview panel 235
DISABLED
Status field 214
Display Suppressed field
Automation Control panel 213
documentation information 11
DOM 324

E

Enable command
Rule Processor application
Automation Control panel 217
Rule Set Overview panel 235
EOM
event type 35
EOM, event type 51
EOS
event type 35
EOS, event type 52
event
defining events 35, 241
Rule Processor application 22
special events
BRC 49
CICa 50
DB2 51
EOM 51
EOS 52
IMP 55
IMSa 56
IMSs 56, 57
MSGa 58
MSGs 59
TIME 60
TLM 52, 61
VAR 62
types
ALRM 35
ALRT 35
BRC 35
CICa 35
CICS 35
CMD 35
DB2 35
EOM 35
EOS 35
EXT 35
IMP 35
IMSa 35
IMSs 35
JRNL 35
MQS 35
MSG 35
MSGa 35
MSGs 35
NVC 35
NVM 35
TIME 35
TLM 35
VAR 35
Event Activity Statistics application accessing 42
MCTSIZE 47
Rule Processor application
fields 47
line commands 44
primary commands 44
sample panel 42
Event field, Rule Processor Selection Criteria panel
ALRT-Initiated Rule 279

event type
See event 35, 241

Event Type field
Rule Processor Detail Control panel 75

event variables 160, 182

&IMFMSTYP 169
BRC 181
IMFEVFRD 169
IMFGROUP 169

Rule Processor application
## Index

### F

**features**

- AutoOPERATOR 13

**fields**

- Rule Processor application
  - ALERT Action(s) Specification panel 321
  - Automation Control panel 213
  - Event Activity Statistics application 47
  - Rule Processor Detail Control panel 73
  - Variable Dependencies 294

- Filter command, Rule Set Overview panel 235
- Filtered field
  - Automation Control panel 214
  - values 214

- filtering
  - Rule Sets 214
- FIND command
Rule Processor application, Event Activity Statistics application 44
Fired field
Rule Processor application
Automation Control panel 214
Fired field, Rule Processor application
Rule Set Overview panel 229
FIRST, Rule Processor application, Automation strategy 218
Function field
Rule Processor ALERT Action(s) panel, ALERT Action(s) Specification panel 321
Rule Processor application, Rule Set Overview panel 229

G
Group field
Rule Set Overview panel 229

H
Handled field, Event Activity Statistics application 47
Hardcopy Suppressed field
Automation Control panel 213
Help field
Rule Processor ALERT Action(s) panel, ALERT Action(s) Specification panel 321
Honor MPF Suppression field, Automation Control panel 213

I
ID field, Rule Set Overview panel 229
If matched field
Rule Processor Detail Control panel
Criteria match rate
Rule setting
If matched field

Ignore leading plus field, Rule Processor Selection Criteria panel
MSG-Initiated Rule 279
IMFACCTG
event variables 169
IMFALGID
event variables 169

IMFALID
event variables 169
IMFALRM
event variables 169
IMFAPRI
event variables 169
IMFAQID
event variables 169
IMFAUTH
event variables 169
IMFCLOCK
event variables 169
IMFCNVTXT
event variables 169
IMFCOLOR
event variables 169
IMFCONNM
event variables 169
IMFDAY
event variables 169
IMFDADM
event variables 169
IMFDOMID
event variables 169
IMFETYPE
event variables 169
IMFEVFRD
event variables 169
IMFGROUP
event variables 169
IMFHILIT
event variables 169
IMFINTEN
event variables 169
IMFJCAN
event variables 169
IMFJCLAS
event variables 169
IMFJNUM
event variables 169
IMFJTYPE
event variables 169
IMFLPROD
event variables 169
IMFLTYPE
event variables 169
IMFLUSER
event variables 169

MainView AutoOPERATOR Basic Automation Guide Volume 1: Using Rules
IMFMPFAU  
event variables 169
IMFMPFSF  
event variables 169
IMFMPSTYP  
event variables 169
IMFOASID  
event variables 169
IMFODATE  
event variables 169
IMFOJOB  
event variables 169
IMFOQID  
event variables 169
IMFORGN  
event variables 169
IMFORGSS  
event variables 169
IMFOTIME  
event variables 169
IMFPCMD  
event variables 169
IMFPLUS  
event variables 169
IMFREPLY  
event variables 169
IMFRFRD  
event variables 169
IMFRLID  
event variables 169
IMFRMLMAT  
event variables 169
IMFRUSER  
event variables 169
IMFSCOPE  
event variables 169
IMFSTEP  
event variables 169
IMFSUPR  
event variables 169
IMFTCPCT  
event variables 169
IMFTCPSC  
event variables 169
IMFTERMT  
event variables 169
IMFTEXT  
event variables 169
IMFTHTGT  
event variables 169
IMFTIME  
event variables 169
IMFTOKEN  
event variables 169
IMFTSNUM  
event variables 169
IMFTSSID  
event variables 169
IMFTTLM  
event variables 169
IMFTWTCT  
event variables 169
IMFTWTSC  
event variables 169
IMFVALUE  
event variables 169
IMFVIEW  
event variables 169
IMFXDESC  
event variables 169
IMFXOJOB  
event variables 169
IMFXRROUT  
event variables 169
IMP  
event type 35
IMP, event type 55
IMS generic parameter process 311
IMS, event type 35
IMSa  
event type 35
IMSa, event type 56
IMSPlex commands issuing from a Rule 311
IMSs  
event type 35
IMSs, event type 56, 57
in seconds field  
Rule Processor Detail Control panel  
Criteria match rate  
Rule setting  
in seconds field 76
Index Term 221, 243, 245, 246, 248, 249, 251, 252, 257, 259, 260, 263, 268
INDIVIDUAL, Rule Processor application,
Automation strategy 219
Insert command, Rule Set Overview panel 235
Interval field, Rule Processor Selection Criteria panel
Time-Initiated Rule 279
introduction, AutoOPERATOR 13

J
Job Class field, Rule Set Overview panel 229
Job name field
Rule Processor Selection Criteria panel 279
Rule Set Overview panel 229
Job Type field, Rule Set Overview panel 229
Jobclass field, Rule Processor Selection Criteria panel 279
Journal field
Rule Processor Action Specification panel 298
JRNL, event type 35

K
Key field
Rule Processor ALERT Action(s) panel, ALERT Action(s) Specification panel 321
Rule Processor Selection Criteria panel, ALRT-Initiated Rule 279

L
Last Modified field
Rule Processor Detail Control panel 78
LC field
Rule Processor application, Event Activity Statistics application 47
line commands
Rule Processor application
Automation Control panel 207
Event Activity Statistics application 44
Rule Set Overview panel 229
LOADED
Status field 214
LOCATE command
Rule Processor application, Event Activity Statistics application 44
LOCATE primary command, Rule Processor application
Automation Control Panel 210
naming conventions 210
long variable, restrictions 161

M
MainView Alarm Manager
view ALERTs 102
managing Rules 30
See also Rule Processor application 21
MLSEG Minor field
Rule Processor Selection Criteria panel, IMS-Initiated Rule 279
MLTDQ Minor field
Rule Processor Selection Criteria panel, CICS-Initiated Rule 279
MLWTO Minor field
Rule Processor Selection Criteria panel, MSG-Initiated Rule 279
MODIFY, Alert Action(s) I panel 100
modifying Alerts 100
Move command, Rule Processor application
Automation Control panel 217
Rule Set Overview panel 235
MQS, event type 35
MSG-Initiated Rule 279
MSG, event type 35
MSGa, event type 35, 58
MSGs, event type 35, 59

N
Name field
Rule Processor Selection Criteria panel, VAR-Initiated Rule 279
NVC, event type 35
NVM, event type 35

O
Op field, Variable Dependencies panel 294
operators 187
Rules functions 187
organizing Rules 30
See also Rule Processor application 21
Origin field
Rule Processor ALERT Action(s) panel, ALERT Action(s) Specification panel 321
Rule Processor Selection Criteria panel, JRNL-Initiated Rule 279

P

PARMSHOW primary command, Rule Processor application 210
naming conventions 210
pattern matching, Rule Processor application debugging 160
description 155, 160
equation 158
wildcard characters 155
PCMD field
Rule Processor ALERT Action(s) panel 321
Peak Arrival Rate field
Automation Control panel 213
POST ID field
Rule Processor Action Specification panel 298
POST Parms field
Rule Processor Action Specification panel 298
POST Target field
Rule Processor Action Specification panel 298
primary commands
Rule Processor application Automation Control 210
Event Activity Statistics application 44
Rule Processor Detail Control panel 73, 84
Rule Set Overview panel 235
Primary Option Menu accessing AutoOPERATOR 17
Priority field
Rule Processor ALERT Action(s) panel, ALERT Action(s) Specification panel 321
Rule Processor Selection Criteria panel, ALRM-Initiated Rule 279
publications, related 11
Publish field
ALERT Action(s) Specification panel 321

Q

Queue(s) field

R

RACF Group field
Rule Processor Selection Criteria panel 279
Rule Set Overview panel 229
RACF User field
Rule Processor Selection Criteria panel 279
Rule Set Overview panel 229
Reject Command field
Rule Processor Action Specification panel 298
related publications 11
Repeat command, Rule Set Overview panel 235
reserved words 188
Rules functions 188
RESET command, Rule Set Overview panel 235
restriction, long variable 161
Retain field
Rule Processor application, ALERT Action(s) Specification panel 324
Reword ALERT field
Rule Processor Action Specification panel 298
Reword CMD field
Rule Processor Action Specification panel 298
Reword Msg field
Rule Processor Action Specification panel 298
RFIND command, Event Activity Statistics application 44
Route codes field
Rule Processor Action Specification panel 298
Rule Processor Selection Criteria panel JES3-Initiated Rule 279
MSG-Initiated Rule 279
Rule generated ALERTs field
Automation Control panel 213
Rule ID field
naming conventions valid characters 75
Rule Processor Detail Control panel 75
Rule invoked EXECs field
Automation Control panel 213
Rule Processor application
action specification fields 90
ALERT Action(s) Specification 94, 102
AND/OR logic 88
Automation Control 207
Automation Control panel 213
Automation Status 209
Automation Strategy 209, 218
compound variables 162
Event Activity Statistics application 41, 47
event types 35, 62, 241
FIRST Automation strategy 218
grouping into Rule Sets 214
INDIVIDUAL Automation strategy 219
introduction
  action specification 25
  Automation Control application 31
  automation strategy 31
  defining Rule Sets 30
  defining Rules 22
  events 22
  Rule Set Overview application 32
  selection criteria 25
managing 30
organizing 30
pattern matching 155, 160
Rule Processor Detail Control panel 84
Rule Set Overview 229, 238
sample panels
  action specification 90
  ALERT Action(s) I specification 95
  Automation Control 66
  Automation Menu 42
  Event Activity Statistics 42
  Rule Processor Detail Control 66
  Rule Processor Event Types 66
  Rule Set Overview-View 1 229
  Rule Set Overview-View 2 229
  Rule Set Overview-View 3 229
  Rule Set Overview-View 4 229
  Rule Set Overview-View 5 229
  selection criteria 85
selection criteria fields 85, 279
setting occurrences 78
variable 160
variable dependencies 86, 90
wildcard characters 155
Rule Processor Detail Control
  field summary 80
  fields 75
  primary commands 73, 84
  sample panel 66
  setting occurrences 78
Rule Set
  adding 210
  deleting 238
  filtering 219
  grouping 214
  naming conventions 210
Rule Set Overview application
  introduction 32
  line commands 235
  primary commands 235
  sample panels 229
Rule Set to be Updated field, Event Activity
  Statistics application 47
Rule-id field, Rule Set Overview panel 229
Rule-Set field
  Automation Control panel 214
Rules
  Rule Processor Detail Control panel 76
  Rules field
    Automation Control panel 214
    Rules functions 185, 187, 188

S
S command, Event Activity Statistics application 44
S1 primary command, Rule Processor Detail
  Control panel 73, 84
SAVE command
  Rule Processor application, Automation
    Control panel 217
Select command
  Rule Processor application
    Automation Control panel 217
    Rule Set Overview panel 235
Selection Criteria fields, Rule Processor application
  85, 279
selection criteria, Rule Processor application 25
Send (TSO) field
  Rule Processor Action Specification panel 298
Set IOA Variable field
  Rule Processor Action Specification panel 298
setting occurrences
  Rule Processor Detail Control panel 78
SHOWALL command, Event Activity Statistics application 44
SHOWMESSAGE command, Event Activity Statistics application 44
SHOWPER command, Event Activity Statistics application 44
SHOWTOP command, Event Activity Statistics application 44
single quotation marks 185
slots 182
SORT command
Rule Processor application
Event Activity Statistics application 44
Rule Set Overview panel 235
Sort Criterion field, Event Activity Statistics application 47
special events
   CICa automation strategy 50
   IMP automation strategy 55
   IMSa automation strategy 56
   IMSs automation strategy 56, 57
   MSGa automation strategy 58
   MSGs automation strategy 59
Stat field, Rule Set Overview panel 229
STATSHOW primary command, Rule Processor application
   Automation Control Panel 210
   naming conventions 210
Status field
   Automation Control panel 214
   Rule Processor Action Specification panel 298
Strategy field
   Automation Control panel 214
   substringing, Rule Processor application, variables 166
SV primary command, Rule Processor Detail Control panel 73, 84
SYSDATE event variables 169
SYSLOG Display field
   Rule Processor Action Specification panel 298
SYSTEM field
   ALERT Action(s) Specification panel 321
SYSTIME event variables 169

T
TapeSHARE parameters implementing 241, 242, 262, 266, 267, 270–274
Target field
   Rule Processor ALERT Action(s) panel
   ALERT Action(s) Specification panel 321
TEST Status field 214
Test command
   Rule Processor application
   Automation Control panel 217
   Rule Set Overview panel 235
Text field
   Rule Processor ALERT Action(s) panel, ALERT Action(s) Specification panel 321
   Text ID field, Rule Processor Selection Criteria panel 279
Text string field
   Rule Processor Selection Criteria Panel 279
   Rule Set Overview panel 229
Text-id field, Rule Set Overview panel 229
Text-ID/Description field, Event Activity Statistics application 47
Time field
   Automation Control Panel 214
   Rule Processor Action Specification panel 298
TIME, event type 35, 60
TLM event type 35
TLM, event type 52, 61
Total Events field
   Automation Control panel 213
   Rule Processor application
   Event Activity Statistics application 47
trimming
   Rule Processor application
     list of 169
   Rule Processor Action
     list of 169
   variables 166, 167
Type field
   Rule Processor application
     Event Activity Statistics application 47
     Rule Set Overview panel 229
   Rule Processor Selection Criteria panel 279
Udata field
  ALERT Action(s) Specification panel 321
UNIQUE field, Event Activity Statistics application 47
UNSORT command, Rule Set Overview panel 235
Update Rout/Desc Codes field
  MSG Rule Action Specification panel 298
User field
  ALERT Action(s) Specification panel 321

V
VAR, event type 35, 62
Variable dependencies
  AND/OR logic 88
  primary commands 88
  Rules Processor application 88
variable dependencies with AND/OR logic
  Rules Processor application 88
variable dependencies, Rule Processor application fields 294
  sample panel 86, 88, 92
variable usage
  Rule Processor application 155
  variables 181
  Variable-Name field, Variable Dependencies panel 294
  Variable-Value field, Variable Dependencies panel 294
variables 181
  Rule Processor application defining 160, 161
  event variables 161
  resolving 161
  search order 161
  simple 162
  substringing 166
  trimming 166, 167
See also event variables
  variables for ALRT, AMFKEY 98, 183

W
wildcard characters, Rule Processor application 155
WORDxxx
  event variables 169