MainView AutoOPERATOR
Customization Guide

Supporting products

Version 8.1 of MainView AutoOPERATOR

December 2016
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 or 1 800 841 2031</td>
<td>1 713 918 8000</td>
</tr>
<tr>
<td>2103 CITYWEST BLVD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HOUSTON TX 77042-2827 USA</td>
<td></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, MQSeries, MVS, NetView, OS/390, RACF, RETAIN, System z, Tivoli, VTAM, WebSphere, z/OS, z/VM, and z9 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.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

UNIX is the registered trademark of The Open Group in the US and other countries.

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 ................................................................. 15
Related publications .................................................................................................................................................................................. 15
Conventions .................................................................................................................................................................................. 16

Chapter 1       Migrating from MainView AutoOPERATOR version 7.4.xx to version 8.1 ................................................................. 17
What you need to know ........................................................................................................................................................................... 17
    Migrating Rules, EXECs, solutions, and parameters to the new release .................................................. 18
General information about the new release ........................................................................................................................................... 18
    What has changed about Rules ........................................................................................................................................... 18
    What has changed about EXECs .................................................................................................................................. 20
    What are the changes made to variables ......................................................................................................................... 21
    What DDNAMEs have changed ................................................................................................................................... 21
    What parameter members have changed ......................................................................................................................... 21
    What commands have changed .................................................................................................................................. 25
    What views or panels have changed ................................................................................................................................. 26
    What messages have changed ........................................................................................................................................... 27
    Migrating MainView Total Object Manager (MVTOM) ........................................................................................................... 27
    What else has changed ......................................................................................................................................................... 27
Other changes .................................................................................................................................................................................. 27
    Documentation name changes ........................................................................................................................................... 29
    What PTFs should I apply ................................................................................................................................................... 30
How to secure MainView AutoOPERATOR ........................................................................................................................................... 30
How much CPU does this release use ............................................................................................................................................... 30
Does this version of MainView AutoOPERATOR use common or private storage differently .................................................................................................................................................. 30
Does this version communicate with earlier versions ........................................................................................................................................... 30
Does version 7.4.xx communicate with version 8.1 ........................................................................................................................................... 31
What other BMC products run with this version of MainView AutoOPERATOR ........................................................................................................................................... 31
Which IBM software does this version of MainView AutoOPERATOR require ........................................................................................................................................... 31
What functions will not be supported in future versions ........................................................................................................................................... 32
Installation considerations for MainView AutoOPERATOR 8.1 ........................................................................................................................................... 32

Chapter 2       Migrating from MainView AutoOPERATOR version 7.3.xx to version 7.4 ........................................................................................................................................... 35
What you need to know .................................................................35
  Migrating Rules, EXECs, solutions, and parameters to the new release ..............36
General information about the new release .........................................................36
  What has changed about Rules .................................................................37
  What has changed about EXECs .................................................................38
  What are the changes made to variables .......................................................40
  What DDNAMEs have changed ...................................................................41
  What parameter members have changed .......................................................41
  What commands have changed .................................................................43
  What views or panels have changed ............................................................44
  What messages have changed ...................................................................44
  Migrating MainView Total Object Manager (MVTOM) .....................................44
  What else has changed ................................................................................45
Other changes ...............................................................................................46
  Documentation name changes .....................................................................47
  What PTFs should I apply ............................................................................47
How to secure MainView AutoOPERATOR ......................................................48
How much CPU does this release use .............................................................48
Does this version of MainView AutoOPERATOR use common or private storage differently .................................................................48
Does this version communicate with earlier versions ........................................48
Does version 7.3.xx communicate with version 7.4 ............................................49
What other BMC Software products run with this version of MainView AutoOPERATOR .............................................................49
Which IBM software does this version of MainView AutoOPERATOR require .............................................................49
What functions will not be supported in future versions ..................................49
Installation considerations for MainView AutoOPERATOR 7.4 ............................50

Chapter 3  Implementing Rules Management ..................................................53
Overview ........................................................................................................53
  Rules Registry ............................................................................................53
  AOPLex .......................................................................................................53
  Deployment ................................................................................................54
Implementation considerations ........................................................................54
  Additional Considerations ........................................................................55
  Migrating Rules created earlier to Rules Management registry-based Rules .....................................................................................56
Registry considerations ..................................................................................56
  Registry allocation ....................................................................................56
  Registry organization ................................................................................56
| Parameters required for Rules Management | 57 |
| Connecting a registry to a MainView AutoOPERATOR PAS | 61 |
| ADMIN mode | 61 |
| FULL mode | 61 |

| Chapter 4 | Generating early or interim WTOs | 63 |
| Early WTOs | 63 |
| Interim WTOs | 65 |

| Chapter 5 | Implementing support for EOS and TLM events | 67 |

| Chapter 6 | Continuous operation with the Dynamic Parameter Manager | 69 |
| Using the Dynamic Parameter Manager: overview | 69 |
| Accessing the Dynamic Parameter Manager | 70 |
| Describing the line commands | 72 |
| Describing the fields | 73 |
| Filtering the display of data | 74 |
| Modifying ALERT thresholds in AAOALSxx | 74 |
| Setting ALERTs storage thresholds | 75 |
| Modifying Automation Reporter application parameters in AAOARPxx | 79 |
| Setting Automation Reporter parameters | 79 |
| Modifying EXEC parameters in AAOEXPxx | 82 |
| Setting EXEC Thresholds | 83 |
| Setting EXEC Parameters | 88 |
| Setting high-priority EXECs | 91 |
| Additional EXEC parameters | 92 |
| Modifying MainView AutoOPERATOR parameters in AAOPRMxx | 94 |
| Setting MainView AutoOPERATOR parameters | 95 |
| Modifying TapeSHARE parameters in AAOTSPxx | 109 |
| Customizing TapeSHARE | 110 |
| Dynamically implementing new settings with the Activate command | 115 |
| Describing the fields | 115 |

| Chapter 7 | Sending Aux Log actions to MainView Logger | 117 |
| Enabling logging to MainView Logger | 117 |
| Accessing the AUXLOG view | 118 |

| Chapter 8 | MVS console considerations | 121 |
| Overview | 121 |
| What extended MCS consoles (EMCS) are | 121 |
| Determining how many consoles to allocate | 123 |
Chapter 9  Propagating enqueues to other MVS systems  125

Chapter 10  Implementing TapeSHARE  127
Overview  ............................................................................................................................127
   Prerequisites  ........................................................................................................... 127
   Step 1: Entering the password key for TapeSHARE ..............................................127
   Step 2: Activating TapeSHARE  ............................................................................128
   Step 3: Securing TapeSHARE (optional) ............................................................. 128
   After TapeSHARE is installed .............................................................................. 129

Chapter 11  Implementing Automatic Restart Manager (ARM)  131
Overview  ............................................................................................................................131
Implementation considerations  ...................................................................................... 131
   Implementing ARM support: default ARM policy ...........................................132
   Implementing ARM support: creating a user-defined policy ............................132
   Enabling ARM support ...................................................................................... 133
   Stopping MainView AutoOPERATOR when ARM is enabled ......................... 133

Chapter 12  Implementing support for MPF  135
Overview  ............................................................................................................................135
Implementation considerations  ...................................................................................... 136
   Setting the MPFFILTR parameter  ...................................................................... 136

Chapter 13  Supporting IMFEXEC CMD TYPE(ZLINUX) and TYPE(VM)  139
Prerequisites  ...................................................................................................................... 139
Install MainView for Linux — Servers ........................................................................ 139
Configure BBPARM member BBISSP00 ...................................................................... 139
   Security considerations  .................................................................................... 140
   Identifying zLinux systems .............................................................................. 140

Chapter 14  Implementing support for sysplex variables  143
What sysplex variables are .............................................................................................. 143
Enabling support for sysplex variables ......................................................................... 143
   Activating support for sysplex variables ........................................................... 144
   Specifying the default sysplex variable prefix (optional) ................................ 144
   Specifying the name of the coupling facility ...................................................... 145
   Additional considerations  .................................................................................... 146
Using the coupling facility ......................................................................................... 146
   Coupling facility hardware requirements .......................................................... 147
Rebuilding the coupling facility ................................................................. 147
Coupling facility structure sizing ............................................................ 148
Coupling facility structure definition ....................................................... 149
Modifying the coupling facility structure ................................................ 149
Coupling facility structure connections ................................................... 150
Coupling facility structure persistence .................................................... 151

Chapter 15   Defining OSPI virtual terminals 153
Overview ............................................................................................................................ 153
Using the OSPI ACB pool ............................................................................................... 153
   Naming conventions ..................................................................................................... 154
Defining the ACB pool ..................................................................................................... 154
Defining the ACB pool to MainView AutoOPERATOR ................................................. 155
OSPI application definitions ........................................................................................... 155

Chapter 16   Considerations for OMVS and TCP/IP 157

Chapter 17   Implementing BBI-SS PAS to CICS communication 159
Overview ............................................................................................................................ 159
Implementation considerations ...................................................................................... 160
Site-specific security ......................................................................................................... 160
   Temporary Storage Table (TST) compatibility ............................................................ 160
Specifying resources to CICS .......................................................................................... 161
   Making MainView AutoOPERATOR transactions and programs available
to CICS and CTS ............................................................................................................. 161
   Modifying the Program List Table (PLT) .................................................................... 161
Using auto maintenance .................................................................................................. 163
   TERM requests .......................................................................................................... 163
   INIT requests ............................................................................................................. 164
   ON requests .............................................................................................................. 164
   OFF requests ............................................................................................................. 165
Before you begin: auto maintenance checklist ......................................................... 165
   Initiating auto maintenance ....................................................................................... 165
Modifying CICS startup JCL ........................................................................................ 170
Restarting CICS target regions .................................................................................... 170
MainView AutoOPERATOR for CICS functions (with BBI-SS PAS-to-CICS communication) ................................................................................................................... 171
MainView AutoOPERATOR for CICS functions (without BBI-SS PAS-to-CICS communication) ................................................................................................................... 172

Chapter 18   MainView AutoOPERATOR for CICS optional tailoring 175
Overview ............................................................................................................................175
Customizing terminal types eligible for broadcast messages (CAOTTAB) .............175
Limiting CICS transient data queues eligible for Rule processing (CAODTAB) ....177
Using the CAODTAB macro ...........................................................................................177
Describing CAODTAB macro parameters .................................................................178

Chapter 19         Implementing MainView products in IMS 181
Deleting modules from a prior release .......................................................................... 181
Setting up BBPARM data sets .........................................................................................182
Customizing BBPARM members for an IMS system .................................................. 184
Activating MainView AutoOPERATOR for IMS ........................................................185
  Enabling product initialization .................................................................................... 185
  Enabling product authorization .................................................................................. 185
Modifying the IMS control region JCL ......................................................................... 186
  Allocating a BBPARM data set ................................................................................ 186
  Giving IMS access to MainView AutoOPERATOR for IMS..................................... 186
Enabling AO exit routines ............................................................................................... 187
  Flow of control ............................................................................................................ 189
  Enhancement to the IMS DFSAOE00 interface ......................................................... 189
MainView AutoOPERATOR for IMS considerations ............................................... 190
  Debug EXEC IAODEBUG ...................................................................................... 191
Stopping message queuing to the IMS master terminal ............................................. 191
Dynamic installation and uninstallation ....................................................................... 192
  Dynamic install JCL ................................................................................................... 193
  Dynamic uninstall JCL ............................................................................................. 195

Chapter 20         Implementing MainView AutoOPERATOR Access NV 197
Overview ............................................................................................................................197
Allocating SYSUDUMP data sets ................................................................................... 198
Modifying NetView JCL .................................................................................................199
Creating DSIPARM member NAIOPT00 ..................................................................... 199
Modifying DSIPARM member DSICMDU ................................................................ 200
Modifying DSIPARM member DSIDMNU .................................................................. 201
Modifying DSIPARM member CNMSTYLE ................................................................. 201
Copying BBSAMP members to the NetView DSICLD data set ....................................202
Modifying DSIPARM member DSIOPFU .....................................................................202
Verifying operator profiles ............................................................................................203
Reviewing NetView security for Access NV operator IDs ............................................203
Reviewing the use of NetView exit DSIEX02A ............................................................ 204
Reviewing the use of NetView exit DSIEX03 ................................................................205
Reviewing the use of NetView exit DSIEX14 ............................................................... 205
Reviewing the use of NetView exit DSIEX16 ............................................................... 206
Modifying BBPARM member BBISSP00 ............................................................... 207

Chapter 21    Implementing BMC Event Manager to work with BMC products 209
Overview of bidirectional communications .............................................................. 209
Establishing a connection to a BMC Event Manager .................................................. 210
  Example of AAOGMEExx settings ........................................................................216
Configuring BBPARM member BBISSP00 to allow rules to send events to BMC
  Event Manager ........................................................................................................216
Establishing a connection to other BMC products ....................................................217

Chapter 22    Using the MainView AutoOPERATOR interface to Control-O and
  CMEM ......................................................................................................................... 219
Implementing the MainView AutoOPERATOR interface to Control-O and
  CMEM ......................................................................................................................... 220

Chapter 23    Implementing advanced JES support ...................................................... 223
Specifying JESFLTR parameter values ........................................................................ 223
  Implementing the JESFLTR parameter ...................................................................224
  Removing the JESFLTR setting ..............................................................................224
  Determining the status of the JESFLTR setting .....................................................225
Additional implementation considerations ...............................................................225

Chapter 24    Implementing Continuous State Manager ............................................227
Performance considerations ...................................................................................... 227
Initiating CSM ............................................................................................................ 227
  Overriding the default group when CSM is initiated .............................................227
  End-of-Memory initiated CSM EXEC (CSMEOM) .................................................228
  Required security resources ..................................................................................229
Accessing CSM for the first time ................................................................................ 229
  Migrating from a previous version of MainView AutoOPERATOR .......................232
  Creating a shared repository data set for CSM .....................................................233
  Migrating a CSM database from a previous release .............................................235
  Resolving object-definition conflicts after converting .......................................238
Using the AATWAIT program for testing Continuous State Manager or the
  MainView Total Object Manager ...........................................................................240

Chapter 25    Specific MainView AutoOPERATOR parameters and JCL ......................245
Enabling REXX support for EXECs ...........................................................................245
Detecting EXEC loops ...............................................................................................245
Prioritizing and threading of EXECs .........................................................................246
Manual activation from a z/OS console that is defined to CICS ........................................280

Appendix E  BBPARM data set members for MainView AutoOPERATOR 281
System variables substitution in parameter library members ...........................................284
BBPARM member AAOALS00 parameters ........................................................................285
BBPARM member AAOARP00 parameters ........................................................................289
BBPARM member AAOEXP00 parameters ........................................................................291
BBPARM member AAOGME00 parameters .......................................................................296
BBPARM member AAOMQL00 parameters .......................................................................301
BBPARM member AAONCD00 parameters ........................................................................303
BBPARM member AAOPRX00 parameters ........................................................................304
BBPARM member AAOPRM00 parameters .......................................................................307
BBPARM members AAORUL00 parameters .......................................................................325
BBPARM member AAOEPR00 parameters ..........................................................................326
BBPARM member BBIVAR00 parameters .........................................................................331

Appendix F  Sample data set members 335
Sample members for MainView AutoOPERATOR only .........................................................335
Sample members for MainView AutoOPERATOR for CICS and MainView for CICS ........337
Sample members for MainView AutoOPERATOR for IMS, MainView for IMS, and MainView for DBCTL ................................................................................................................................40

Appendix G  Expanding the PROFILE pool data set 341
Using the XTLOAD utility .......................................................................................................341

Appendix H  OSPI terminal definitions for CICS and IMS 343
IMS terminal definitions ........................................................................................................343
CICS terminal definitions ....................................................................................................343
  Autostall terminals ...........................................................................................................344
  Explicit terminal definitions .........................................................................................344
  RDO macro .....................................................................................................................344

Appendix I  Implementing Print LOGJRNL records 347
Printing LOGJRNL records .................................................................................................347

Appendix J  BBPARM member BBISSPx parameters 351
About this book

This book contains detailed product information and is intended for system administrators and database administrators (DBAs).

Like most BMC documentation, this book is available in printed and online formats. To request printed books or to view online books and notices (such as release notes and technical bulletins), see the support website at http://www.bmc.com/support.

Note

Online books are formatted as PDF or HTML files. To view, print, or copy PDF books, use the free Adobe Reader from Adobe Systems. If your product installation does not install the reader, you can obtain the reader at http://www.adobe.com.

The software also offers online Help. To access Help, press F1 within any product or click the Help button in graphical user interfaces (GUIs).

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.

**Tip**

You can access the BMC Support Central site at [http://www.bmc.com/support](http://www.bmc.com/support).

---

**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: 
  `testsys/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.
Migrating from MainView AutoOPERATOR version 7.4.xx to version 8.1

You should read this chapter in conjunction with the MainView AutoOPERATOR version 8.1 Release Notes.

**Tip**
In addition to reviewing the MainView AutoOPERATOR version 8.1 Release Notes, you should also review the following publications:

- the latest MainView AutoOPERATOR BMC Impact Integration for z/OS Release Notes

For information about migrating MVTOM, see “Migrating MainView Total Object Manager (MVTOM)” on page 27 and Chapter 1 of the MainView Total Object Manager User Guide.

What you need to know

The following sections include important discussions such as:

- what you must do to migrate from MainView AutoOPERATOR 7.4.xx to MainView AutoOPERATOR 8.1

- what changes have been made to existing parameters, Rules, and EXECs

- what else has changed with the MainView AutoOPERATOR 8.1 release
Migrating Rules, EXECs, solutions, and parameters to the new release

You can run your current Rules, EXECs and solution automation without any manual modifications by copying the existing MainView AutoOPERATOR 7.4.xx Rules, EXECs, and solutions to the new MainView AutoOPERATOR 8.1 PAS. You can also copy your parameter members from the MainView AutoOPERATOR 7.4.xx BBI-SS PAS and use them for the MainView AutoOPERATOR 8.1 PAS.

However, MainView AutoOPERATOR Rules, EXECs, solutions, and BBPARM members might have been updated in the new version of MainView AutoOPERATOR. If you customized the MainView AutoOPERATOR Rules, EXECs, solutions, or BBPARM members in an earlier release, you should be prepared to migrate those changes to the new release.

You can do this process either by recustomizing the Rules, EXECs, solutions, and BBPARM members in the new release, or by reviewing the new members shipped by MainView AutoOPERATOR to see what changes are made in the new release that you can use in your user-modified version.

You can activate any new parameters or parameter values either by editing the BBPARM member, changing the value, and restarting the BBI-SS PAS, or by using the Dynamic Parameter Manager application, which does not require a BBI-SS PAS restart.

Some compatibility limitations exist between MainView AutoOPERATOR 7.4 Rules and Rules from earlier releases. See “What has changed about Rules” on page 18 for information about enhancements that were made to Rules and the compatibility limitations and see “What has changed about EXECs” on page 20 for information about changes that were made to EXECs.

General information about the new release

The following sections contain general information about MainView AutoOPERATOR 8.1.

What has changed about Rules

This version introduces the new Rules Management application, which allows you to create and manage Rules in windows-mode. For more information about using this new optional application, refer to the MainView AutoOPERATOR Customization Guide and the MainView AutoOPERATOR Basic Automation Guide Volume 1: Using Rules.
The following table describes changes made to Rules for this release:

<table>
<thead>
<tr>
<th>Changes made to Rules and the Rule Processor</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Rule Processor supports the following new event types if you have the BMC MainView Automation solution (product code MVN) active in the BBI-SS PAS:</td>
<td></td>
</tr>
<tr>
<td><strong>HWTO</strong>: specifies either early or interim WTOs that are issued while MainView AutoOPERATOR is not up.</td>
<td>the following chapters in the <em>MainView AutoOPERATOR Basic Automation Guide, Volume 1: Using Rules</em>:</td>
</tr>
<tr>
<td>The Rule Processor processes these WTOs as HWTO events.</td>
<td>■  Chapter 3, &quot;Describing events&quot;</td>
</tr>
<tr>
<td>The following new built-in functions are supported in this version:</td>
<td>the following chapters in the <em>MainView AutoOPERATOR Basic Automation Guide, Volume 1: Using Rules</em>:</td>
</tr>
<tr>
<td>■  EXtract( ): extracts a variable-length delimited string from another string</td>
<td>■  Chapter 7, &quot;Creating more powerful Rules&quot;</td>
</tr>
<tr>
<td>■  PArs( ): returns the nth word of a string</td>
<td></td>
</tr>
<tr>
<td>■  CNVTUOM( ): returns a normalized value from a value with a scaling suffix</td>
<td></td>
</tr>
<tr>
<td>■  LIKE( ): returns a TRUE or FALSE after matching a string to a pattern</td>
<td></td>
</tr>
</tbody>
</table>

Additional compatibility issues for Rules from earlier releases

If you are using a terminal session (TS) from MainView AutoOPERATOR version 8.1 and you change the target in the **TGT** field to connect to a MainView AutoOPERATOR version 7.4.xx (or earlier) BBI-SS PAS, you can view and edit the MainView AutoOPERATOR 7.4.xx Rules and Rule Sets.

If you are using a TS from MainView AutoOPERATOR version 7.4.xx (or earlier) and you change the target in the **TGT** field to connect to a MainView AutoOPERATOR version 8.1 (or later) BBI-SS PAS, on the Automation Control panel you can view and edit the MainView AutoOPERATOR version 8.1 (or later) Rules and Rule Sets but you cannot see or use any of the newly added options.

BMC recommends that you always use the same level of TS to update Rules and Rule Sets. Features added to newer releases are not supported by the earlier releases.
What has changed about EXECs

MainView AutoOPERATOR 8.1 has added or modified the following EXEC commands; for more information, see the MainView AutoOPERATOR Advanced Automation Guide.

- The IMFEXEC CICS ALTER command supports the following new parameters:
  - APPLICATION
  - BUNDLE
  - EVENTBINDING
  - FILE

- The IMFEXEC CICS CONN supports the following new parameters:
  - ENDAFFINITY
  - EXITTRACE
  - NOEXITTRACE
  - CANCEL
  - FORCENOTALERT
  - FORCEPURGE
  - KILL
  - NORECOV
  - BACKOUT
  - COMMIT
  - FORCENOTALERT
  - RESYNC
  - ZCPTRACE
  - NOZCPTRACE
What are the changes made to variables

The following table lists new variable added for this release. This new SHARED variable is documented in Chapter 4 of the *MainView AutoOPERATOR Advanced Automation Guide.*

**Table 1: Variable added for the MainView AutoOPERATOR 8.1 release**

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QCASJOBN</td>
<td>contains the jobname of the connected CAS</td>
</tr>
</tbody>
</table>

What DDNAMEs have changed

No changes were made to existing DDNAMEs in this release.

What parameter members have changed

The following table lists the new or changed parameters, (including the default setting, and a brief description) that are included in MainView AutoOPERATOR version 8.1. You can locate the complete definition and additional usage documentation in this manual for each of the parameters in either of the following chapters:

- “Continuous operation with the Dynamic Parameter Manager” on page 69
  
  You can modify the parameter settings from the Dynamic Parameter Manager application and for most of the parameter changes, you can dynamically implement the changes.

- “BBPARM data set members for MainView AutoOPERATOR” on page 281
  
  You can manually edit the parameter settings in the BBPARM member.

**Note**

The parameters for BBPARM member AAOPLX00 are not available for editing from the Dynamic Parameter Manager. The parameters in this member support the Rules Management application, which requires a COLD start to take effect.
Table 2: Added or changed parameters for MainView AutoOPERATOR version 8.1

<table>
<thead>
<tr>
<th>BBPARM member name</th>
<th>Parameter</th>
<th>Brief description</th>
</tr>
</thead>
</table>
| AAOARP00           | AROPTS=[TOM | ELU | RLU] | specifies that windows mode Automation Reporter will collect data about TOMEXECs (TOM), EXECs (ELU) or Rules (RLU) When you specify one of the following options:  
  ■ TOM: collects data about TOMEXEC EXECs  
  ■ ELU: collects data about EXECs  
  ■ RLU: collects data about Rules  
  For the ELU and RLU options, you must define a CASPERM registry. For more information, see the MainView Administration Guide. If you do not have a CASPERM registry defined, you will get the AAOR0720I or the AAOR0820I messages  
  ■ ALL: all collectors will collect data (but does not include the ELU or RLU options) |
| AAOEXP00           | MSGID     | updates the TSO user profile MainView AutoOPERATOR uses to turn on (and off) the TSO MSGID option  
  The MSGID parameter defaults to MSGID=Y, which means that if you used a TSO Segment in an external security manager (ESM) with PROFILE NOMSGID, you will find TSO messages now use the MSGID as the prefix. This might impact Rules automation if JRNL event Rules were written for messages that do not use MSGID as the prefix. |
<table>
<thead>
<tr>
<th>BBPARM member name</th>
<th>Parameter</th>
<th>Brief description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAOPLX00</td>
<td>AO_NAME</td>
<td>specifies the name to identify this PAS within the AOPlex</td>
</tr>
<tr>
<td></td>
<td>AOPLEX_NAME</td>
<td>specifies the name of the AOPlex to which this PAS belongs</td>
</tr>
<tr>
<td></td>
<td>AOPLEX_XCF_GROUP</td>
<td>specifies the name of the XCF group that the AOPlex will use to communicate</td>
</tr>
<tr>
<td></td>
<td>REGISTRY_DSN</td>
<td>specifies the data set name of the linear VSAM data set allocated for use by this AOPlex</td>
</tr>
<tr>
<td></td>
<td>REGISTRY_WARNHWM</td>
<td>specifies the Registry high-water-mark percentage warning message threshold</td>
</tr>
<tr>
<td></td>
<td>USAGE_MODE_RULES</td>
<td>specifies to the PAS how to use the registry</td>
</tr>
<tr>
<td></td>
<td>AGSLOG</td>
<td>specifies whether the MainView AutoOPERATOR Generalized Services (AGS) messages are sent to a logger</td>
</tr>
<tr>
<td></td>
<td>AGSLOG_FAILOVER</td>
<td>specifies what happens to AGS messages when AGSLOG=YES, FORCEWTO=NO, and AGSJRN=NO are specified and the logger is not available</td>
</tr>
<tr>
<td></td>
<td>FORCEWTO</td>
<td>specifies whether AGS product messages are sent to the PAS JES job log</td>
</tr>
<tr>
<td></td>
<td>AGSJRNl</td>
<td>specifies whether AGS product messages are sent to the BBI-SS journal</td>
</tr>
<tr>
<td></td>
<td>FORCE_CLASSIC_RSNAMES</td>
<td>specifies whether Rule Sets created in the registry by the user address space (UAS) follow the classic Rule Set naming that requires the name to start with RUL or AAORUL</td>
</tr>
<tr>
<td></td>
<td>AUDIT_STARTUP_RULESETS</td>
<td>causes message AGSIA4050I to be generated to the PAS syslog during a PAS COLD start to list the rulesets that the PAS will use and is similar to the PROCESS LIST= parameter in BBPARM member AOPRMxx</td>
</tr>
<tr>
<td>BBPARM member name</td>
<td>Parameter</td>
<td>Brief description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AAOPRM00</td>
<td>MAXDLYQ</td>
<td>The default setting for this parameter has been changed from 100 to 999.</td>
</tr>
<tr>
<td></td>
<td>MSGJOBNM</td>
<td>specifies how MainView AutoOPERATOR should derive batch, Started Task, and TSO job names for MSG events. The default for this option is changed in this version from ACTUAL to ORIGIN. You should consider the impact of this change. Rules might be affected. MSG event Rules that report the origin of a message as CONSOLE might suddenly find that value is changed and your automation might not perform as expected. If you are uncertain about the impact of this change, you should ensure that this parameter is specified as MSGJOBNM=ACTUAL.</td>
</tr>
<tr>
<td></td>
<td>RTCSCMD= MVS_startCommand_f or _RTCS</td>
<td>specifies the MVS command start command for BMC Runtime Component System (RTCS)</td>
</tr>
<tr>
<td></td>
<td>UCCICTRN</td>
<td>specifies whether CICSTRAN transaction parameters will be translated to uppercase</td>
</tr>
<tr>
<td></td>
<td>WTOEARLY</td>
<td>specifies that early WTOs, captured by the QAOCNZ15 utility, are available as HWTO events to the Rule Processor</td>
</tr>
<tr>
<td></td>
<td>WTOINTRM</td>
<td>specifies that WTOs, captured during brief MainView AutoOPERATOR outages, are available as HWTO events to the Rule Processor</td>
</tr>
</tbody>
</table>
### What commands have changed

The following sections describes new or changed BBI control commands.

#### .DISPLAY ACTIVE

The BBI control command .DISPLAY ACTIVE command response includes a new message when there are one or more MainView product packages installed in the BBI-SS PAS. The following shows an example of the message:

```
CF0201I  MAINVIEW PRODUCT PACKAGES ACTIVATED BY PRODUCT PASSWORD CODE:
CF0210I      BMC MAINVIEW FOR CICS MANAGEMENT   (MVK)
CF0210I      BMC MAINVIEW FOR DB2 MANAGEMENT    (MV2)
CF0210I      BMC MAINVIEW FOR IMS MANAGEMENT    (MVF)
CF0210I      BMC MAINVIEW AUTOMATION            (MVN)
CF0210I      BMC MAINVIEW MESSAGE MANAGEMENT    (MVG)
CF0210I      BMC MAINVIEW FOR NETWORKS          (MVR)
```
**.SET**

The BBI control command .SET supports a new parameter, EXIT, that allows you to set the exit as ACTIVE or INACTIVE by either passing the exit name or passing the event type. When EOS or TLM events are passed, all exits required for that event type will be activated. The syntax is as follows:

- `.T EXIT exitname ACT` sets the exit as active
- `.T EXIT exitname INACT` sets the exit as inactive
- `.T EXIT EOS` or `.T EXIT TLM` specifies that when you enter either EOS or TLM, all of the exits for that event type are activated

**What views or panels have changed**

The following table lists all of the views and panels that were changed in MainView AutoOPERATOR in this release.

<table>
<thead>
<tr>
<th>Modified or new views and panels in version 8.1 of MainView AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTOEVNT</td>
</tr>
<tr>
<td>EZAOTAS</td>
</tr>
<tr>
<td>HEALTH</td>
</tr>
<tr>
<td>ARXCMDQD</td>
</tr>
<tr>
<td>ARXCMDS</td>
</tr>
<tr>
<td>ARXCMDR</td>
</tr>
<tr>
<td>ARHEALTD</td>
</tr>
<tr>
<td>RULEDTL</td>
</tr>
<tr>
<td>ASYSCHK</td>
</tr>
<tr>
<td>RULESETS</td>
</tr>
<tr>
<td>VPL</td>
</tr>
<tr>
<td>DEPCHSB</td>
</tr>
<tr>
<td>DPLYITMS</td>
</tr>
<tr>
<td>RSET</td>
</tr>
<tr>
<td>ARLUBNAM</td>
</tr>
<tr>
<td>AOEVENTS</td>
</tr>
</tbody>
</table>
What messages have changed

No messages from were changed in this release.

Migrating MainView Total Object Manager (MVTOM)

Review the section, "Migrating from an earlier version of MainView TOM" in the MainView Total Object Manager User Guide for information about migrating a definition base from an earlier version to version 8.1.

What else has changed

The following sections describe other changes that are part of this version.

Other changes

The following table describes some other changes that are part of MainView AutoOPERATOR version 8.1.

Table 3: Additional changes made in MainView AutoOPERATOR version 8.1

<table>
<thead>
<tr>
<th>Change</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>The product named MainView AutoOPERATOR for WebSphere MQ is now named MainView AutoOPERATOR for MQ.</td>
<td>All of the documentation for MainView AutoOPERATOR version 8.1 has been updated to reflect this name change.</td>
</tr>
<tr>
<td>Change</td>
<td>Documentation</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>The ARXCMDS view now supports showing what type of IMFEXEC and AOEXEC commands are issued from each EXEC thread during a time period. The data is reported under the name of the first EXEC scheduled in an EXEC thread. In earlier MainView AutoOPERATOR versions, the Automation Reporter application collected and displayed EXEC-related data for each EXEC, regardless of how the EXEC was invoked. With this version, the EXEC-related data is collected per EXEC thread. Therefore, if an EXEC is invoked as a result of another EXEC issuing IMFEXEC SELECT EXEC() WAIT(YES), the statistics for the called EXEC (and any other EXECs within the same thread) are also collected and displayed under the main EXEC. The other issue addressed in this version is related to disabled EXECs. In earlier versions of MainView AutoOPERATOR, the Automation Reporter statistics were not displayed for disabled EXECs. With this version, statistics for currently disabled EXECs are included in Automation Reporter views.</td>
<td>Chapter 4 of the MainView AutoOPERATOR Basic Automation Guide, Volume 2.</td>
</tr>
<tr>
<td>The following views now include a hyperlink to show a new TOMEXEC Functions view that shows a summary of the TOMEXEC functions used by EXECs during the chosen time period:  ■ EZAOAR – Automation Reporter menu  ■ ARXCMDS - EXEC Commands/Overview  ■ ARXCMDG - IMFEXEC General  ■ ARXCMDC - IMFEXEC CICS  ■ ARXCMDQ - IMFEXEC MQ  ■ ARXCMDR - IMFEXEC ARRAY  ■ ARXCMDM - IMFEXEC MV/MVX  ■ ARXCMDI - IMFEXEC OSPI  ■ ARXCMDA - AOEXEC Commands</td>
<td>Chapter 4 of the MainView AutoOPERATOR Basic Automation Guide, Volume 2.</td>
</tr>
</tbody>
</table>
The documentation has been removed from the following documents:
- MainView AutoOPERATOR Options User Guide
- MainView AutoOPERATOR Customization Guide

The name of the MainView AutoOPERATOR for WebSphere MQ Installation and User Guide has been changed in this version to MainView AutoOPERATOR for MQ Installation and User Guide.
What PTFs should I apply

If you install MainView AutoOPERATOR 8.1 in the same environment where earlier versions of MainView AutoOPERATOR are running, you might need to apply additional PTFs to the earlier MainView AutoOPERATOR BBI-SS PASs.

Refer to the MainView AutoOPERATOR version 8.1 Release Notes for a complete list of any PTFs that you need to apply.

How to secure MainView AutoOPERATOR

Just as with previous versions of MainView AutoOPERATOR, you can implement security for the 8.1 version through an External Security Manager (ESM). While you can continue to use BMC Software proprietary security, you might want to control security of MainView AutoOPERATOR through your ESM.


Review the MainView AutoOPERATOR version 8.1 Release Notes for the list of security resource definitions that are added to this version for MainView AutoOPERATOR and MainView TOM.

How much CPU does this release use

This release does not use a significantly higher amount of CPU than the previous release.

Does this version of MainView AutoOPERATOR use common or private storage differently

This version of MainView AutoOPERATOR does not use common or private storage differently than earlier versions.

Does this version communicate with earlier versions

Any BBI-SS PAS with an activated MainView AutoOPERATOR 7.4.xx product can schedule an ALERT or EXEC to any MainView AutoOPERATOR 8.1 BBI-SS PAS. In
addition, a MainView AutoOPERATOR 7.4.xx terminal session can access MainView AutoOPERATOR 8.1 panels.

However, MainView AutoOPERATOR commands (or Rules) that schedule EXECs or create ALERTs that target a version 7.4.xx BBI-SS PAS from an version 8.1 BBI-SS PAS cannot contain any of the new keywords or features that are introduced in MainView AutoOPERATOR 8.1. MainView AutoOPERATOR rejects those commands before the commands are sent. When this situation occurs, depending on how the command is initiated, MainView AutoOPERATOR might issue a message.

A good approach is to install the new version of MainView AutoOPERATOR onto a test system and use it to review the new features and functions. Then you can migrate the new version of MainView AutoOPERATOR to your other systems as your time and schedules permit.

**Does version 7.4.xx communicate with version 8.1**

ALERTs, EXECs, and Rules can be sent from a MainView AutoOPERATOR 7.4.xx BBI-SS PAS to a MainView AutoOPERATOR 8.1 BBI-SS PAS.

**What other BMC products run with this version of MainView AutoOPERATOR**

MainView AutoOPERATOR version 8.1 works with other currently supported versions of MainView products.

**Which IBM software does this version of MainView AutoOPERATOR require**

This version of MainView AutoOPERATOR works with supported IBM software versions; the minimum levels are

- z/OS version 2.1 or later
- CICS Transaction Server version 4.1
- IMS version 13
- Tivoli NetView for z/OS version 5.4
- IBM MQ for z/OS version 7.1.0
What functions will not be supported in future versions

BMC has identified the following MainView AutoOPERATOR solutions from the MainView AutoOPERATOR for MQ option as items for which support will be discontinued in a future version:

- The Automation Power Line solution (QMQPOWER)
- The Basic Intercommunication Solution

Installation considerations for MainView AutoOPERATOR 8.1

This section contains technique-specific installation considerations for migrating from previous versions of MainView AutoOPERATOR to MainView AutoOPERATOR version 8.1.

Getting new passwords

MainView AutoOPERATOR version 8.1 uses BMC Software Product Authorization passwords. For a description of the Product Authorization utility and information about obtaining and applying passwords, see the Installation System User Guide.

The following table shows how the product component name (AOOPTION) for earlier MainView AutoOPERATOR versions correlates to the new password codes.

Table 4: List of MainView AutoOPERATOR password product codes

<table>
<thead>
<tr>
<th>MainView AutoOPERATOR product component name</th>
<th>MainView AutoOPERATOR AOOPTION</th>
<th>MainView AutoOPERATOR password product code</th>
</tr>
</thead>
<tbody>
<tr>
<td>MainView AutoOPERATOR Access NV</td>
<td>ANV</td>
<td>BKG</td>
</tr>
<tr>
<td>MainView AutoOPERATOR TapeShare</td>
<td>TSH</td>
<td>BCG</td>
</tr>
<tr>
<td>MainView AutoOPERATOR for CICS</td>
<td>CAO</td>
<td>BCC</td>
</tr>
<tr>
<td>MainView AutoOPERATOR for IMS</td>
<td>IAO</td>
<td>BCD</td>
</tr>
<tr>
<td>MainView AutoOPERATOR for MQ</td>
<td>QAO</td>
<td>BCI</td>
</tr>
<tr>
<td>MainView AutoOPERATOR for z/OS</td>
<td>MAO</td>
<td>BCE</td>
</tr>
<tr>
<td>BMC MainView for CICS Management</td>
<td>CAO</td>
<td>MVK</td>
</tr>
<tr>
<td>BMC MainView for IMS Management</td>
<td>IAO</td>
<td>MVF</td>
</tr>
</tbody>
</table>
Even if you are migrating from an earlier version of MainView AutoOPERATOR or changing from one CPU to another CPU, you must acquire new BMC Software Product Authorization passwords to use any version MainView AutoOPERATOR version 6.4 or later.

If you want to use these passwords in multiple PASs, but you want only certain product options to be active within a specific PAS, you can install the individual MainView AutoOPERATOR component passwords into separate password library data sets or use the AOPTION parameter in BBPARM member AAOPRM.xx to limit the components that are activated in the PAS.

After this process is complete, concatenate together the password library data sets for the specific product options that you want active for the PAS. For example, your concatenation might look like the following example where the two product components that are activated for this PAS are:

- MainView AutoOPERATOR for CICS (CAO)
- MainView AutoOPERATOR for z/OS (MAO)

```bash
//BMCPSWD DD DISP=SHR,DSN=AAO.BMCPSWD.CAO   (CAO ONLY)
//         DD DISP=SHR,DSN=AAO.BMCPSWD.MAO   (MAO ONLY)
```

In the following example, the first line of the concatenation shows that all of the MainView AutoOPERATOR product components, with the exception of TapeSHARE, are included. This exception is because you can run TapeSHARE in only one PAS per LPAR (the first one that is started with that password). Therefore, you should include the TapeSHARE password in the JCL of only one BBI-SS PAS.

**Figure 1: Example of specifying passwords for all MainView AutoOPERATOR product components**

```bash
//BMCPSWD DD DISP=SHR,DSN=AAO.BMCPSWD       (ALL AAO No/TapeSHARE)
//** DD DISP=SHR,DSN=AAO.BMCPSWD.ALL   (ALL AAO)
//** DD DISP=SHR,DSN=AAO.BMCPSWD.ANV   (ANV ONLY)
//** DD DISP=SHR,DSN=AAO.BMCPSWD.CAO   (CAO ONLY)
//** DD DISP=SHR,DSN=AAO.BMCPSWD.IAO   (IAO ONLY)
//** DD DISP=SHR,DSN=AAO.BMCPSWD.MAO   (MAO ONLY)
//** DD DISP=SHR,DSN=AAO.BMCPSWD.MQS   (MQS ONLY)
//** DD DISP=SHR,DSN=AAO.BMCPSWD.SHA   (SHA ONLY)
//** DD DISP=SHR,DSN=AAO.BMCPSWD.TSH   (TSH ONLY)
```
To create a universal JCL, you can use symbolics with the &SSID within the data set name. Parameters that are passed to the PROC statement allow you to select one set of passwords to use; see the following example:

Figure 2: Example of using symbolics for passwords for MainView AutoOPERATOR product components

//BMCPSWD DD DISP=SHR,DSN=AAO.&SS..BMCPSWD (Set for this PAS)

When the SS PAS JCL passes the parameter SS=SYSA to the PROC, this example shows that AAO.&SS..BMCPSWD resolves to AAO.SYSA.BMCPSWD, which is the data set that contains the passwords to use for PAS.

Re-implement JES2MAO exit

If you are migrating by a using Standard installation with the BMC Installation System and you are replacing any MainView AutoOPERATOR version, and all of the following statements are true, you must implement the JES2MAO exit again:

- Your site is licensed for the MainView AutoOPERATOR for z/OS option.
- You elected to implement the JES2MAO exit.
- The JES2MAO exit is linked to the existing system’s BBLINK library.

Installing MainView AutoOPERATOR into an existing zone that includes other BMC products

Ensure all of the other BMC products in the target zone are currently supported versions before installing MainView AutoOPERATOR into the same zone. MainView AutoOPERATOR does not support coexistence with earlier, unsupported versions of BMC products.

MainView AutoOPERATOR also requires that all products running within the same BBI-SS PAS are installed into the same zone and executed out of the same libraries. MainView AutoOPERATOR does not support concatenation of product libraries in BBILOAD or STEPLIB.

When MainView AutoOPERATOR is installed into a zone with other products, it is important to note that MainView AutoOPERATOR shares the use of some FMIDs with other products. Therefore, installing a new version of one product can impact other products. For example, when installing a new version of MainView AutoOPERATOR into a zone where both MainView AutoOPERATOR for CICS and MainView for CICS are running, the new version MainView AutoOPERATOR version might impact MainView for CICS.
Migrating from MainView AutoOPERATOR version 7.3.xx to version 7.4

You should read this chapter in conjunction with the MainView AutoOPERATOR version 7.4 Release Notes.

Tip
In addition to reviewing the MainView AutoOPERATOR version 7.4 Release Notes, you should also review the following publications:

- the latest MainView AutoOPERATOR BMC Impact Integration for z/OS Release Notes
- that latest MainView SYSPROG Release Notes
  Refer also to Compatibility with MainView SYSPROG Services version 3.8 on page 45 for migration and compatibility information.

For information about migrating MainView Total Object manager, see. “Migrating MainView Total Object Manager (MVTOM)” on page 44

What you need to know

The following sections include important discussions such as

- what you must do to migrate from MainView AutoOPERATOR 7.3.xx to MainView AutoOPERATOR 7.4
- what changes have been made to existing parameters, Rules, and EXECs
- what else has changed with the MainView AutoOPERATOR 7.4 release
Migrating Rules, EXECs, solutions, and parameters to the new release

You can run your current Rules, EXECs and solution automation without any manual modifications by copying the existing MainView AutoOPERATOR 7.3.xx Rules, EXECs, and solutions to the new MainView AutoOPERATOR 7.4 PAS. You can also copy your parameter members from the MainView AutoOPERATOR 7.3.xx BBI-SS PAS and use them for the MainView AutoOPERATOR 7.4 PAS.

However, MainView AutoOPERATOR Rules, EXECs, solutions, and BBPARM members might have been updated in the new version of MainView AutoOPERATOR. If you customized the MainView AutoOPERATOR Rules, EXECs, solutions, or BBPARM members in an earlier release, you should be prepared to migrate those changes to the new release.

You can do this process either by recustomizing the Rules, EXECs, solutions, and BBPARM members in the new release, or by reviewing the new members shipped by MainView AutoOPERATOR to see what changes are made in the new release that you can use in your user-modified version.

You can activate any new parameters or parameter values either by editing the BBPARM member, changing the value, and restarting the BBI-SS PAS, or by using the Dynamic Parameter Manager application, which does not require a BBI-SS PAS restart.

See “What parameter members have changed” on page 41 for information about new parameters that were added or changed for this release and see “Continuous operation with the Dynamic Parameter Manager” on page 69 for information about using the Dynamic Parameter Manager application.

Some compatibility limitations exist between MainView AutoOPERATOR 7.4 Rules and Rules from earlier releases. See “What has changed about Rules” on page 37 for information about enhancements that were made to Rules and the compatibility limitations and see “What has changed about EXECs” on page 38 for information about changes that were made to EXECs.

General information about the new release

The following sections contain general information about MainView AutoOPERATOR 7.4.
What has changed about Rules

The following table describes changes made to Rules for this release.

<table>
<thead>
<tr>
<th>Changes made to Rules and the Rule Processor</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Rule Processor supports the following new event types: IMP: specifies the IMS type-2 (IMSplex) commands event type The Rule Processor supports the new Selection Criteria fields, Client, Route list, and UserID for the IMP event type. The Rule Processor supports the new Action Specification fields, Audit reject, Reject Command, and Reword CMD for the IMP event type.</td>
<td>the following chapters in the MainView AutoOPERATOR Basic Automation Guide, Volume 1: Using Rules: ■ Chapter 3, &quot;Describing events&quot; ■ Appendix A, &quot;Field descriptions for the Rule Processor&quot;</td>
</tr>
<tr>
<td>The Select Command Type panel in the Rule Processor is updated in the following ways: ■ The Cmd (Type BBI) supports specifying a 1 to 8-character target name where the BBI command will execute. ■ A new command type field, Cmd (Type TOM), where you can specify up to 5 TOM operator commands that the Rule sends to the TOM PAS.</td>
<td>the following chapters in the MainView AutoOPERATOR Basic Automation Guide, Volume 1: Using Rules: ■ Chapter 3, &quot;Describing events&quot; ■ Appendix A, &quot;Field descriptions for the Rule Processor&quot;</td>
</tr>
</tbody>
</table>

Additional compatibility issues for Rules from earlier releases

If you are using a terminal session (TS) from MainView AutoOPERATOR version 7.4 and you change the target in the TGT field to connect to a MainView AutoOPERATOR version 7.3.xx (or earlier) BBI-SS PAS, you can view and edit the MainView AutoOPERATOR 7.3.xx Rules and Rule Sets.

If you are using a TS from MainView AutoOPERATOR version 7.3.xx (or earlier) and you change the target in the TGT field to connect to a MainView AutoOPERATOR version 7.4 (or later) BBI-SS PAS, on the Automation Control panel you can view and edit the MainView AutoOPERATOR version 7.4 (or later) Rules and Rule Sets but you cannot see or use any of the newly added options.

BMC recommends that you always use the same level of TS to update Rules and Rule Sets. Features added to newer releases are not supported by the earlier releases.
What has changed about EXECS

MainView AutoOPERATOR 7.4 has added or modified the following EXEC commands; for information, see the referenced chapters of the *MainView AutoOPERATOR Advanced Automation Guide*.

- The IMFEXEC CICS ALTER command includes support for the following new parameters:
  - **DOCTEMPLATE**: alters the Document Template attributes the EXEC CICS SET DOCTEMPLATE command
  - **IPCONN**: alters the attributes of a Document Template using the EXEC CICS SET IPCONN command
  - **WEBSERVICE**: alters the status of an installed WEBSERVICE
  - **LIBRARY**: alters the attributes of a library resource
  - **TEMPSTORAGE**: alters the amount of storage that is available for temporary storage queues
  - **MQCONN**: alters the attributes of the connection between CICS and MQ
  - **AUTOINSTALL**: alters AutoInstall values
  - **DELETESHIPPED**: Alters the values that control the deletion of shipped terminals.
  - **DISPATCHER**: alters the dispatcher system information
  - **ENQMODEL**: alters the status of an ENQMODEL definition
  - **EVENTPROCESS**: alters the status of event processing
  - **IRC**: alters the status of interrogation communications
  - **monitor**: alters the CICS monitoring options
  - **STATISTICS**: alters the statistics recording settings of CICS
  - **DB2CONN**: alters the attributes of the CICS DB2 connection
  - **DB2ENTRY**: alters the attributes of a specific DB2ENTRY
  - **DB2TRAN**: alters the attributes of a specific DB2TRAN associated with a DB2ENTRY
  - **UOW**: commits, backs out, or forces a shunted unit of work
— UOWLINK: deletes the link to a unit of work that was created by a connection that has since been discarded

- The IMFEXEC CICS PERFORM command includes support for a new parameter, STATISTICS

- The IMFEXEC VGET and VGETL commands support using wildcard characters with the VAR parameter.

- This release includes support for a new AOEXEC command, AOEXEC CMD Type(MVS) with response, which enables AOAnywhere EXECs to issue MVS commands without requiring that the EXEC executes in a MainView AutoOPERATOR PAS.

- The AOEXEC VGET and VGETL commands support using wildcard characters with the VAR parameter.

- The AOEXEC WTO command supports a new parameter, AREAILD. This parameter does not function when the AOEXEC WTO is started in a version 7.4 MainView AutoOPERATOR PAS that targets an earlier version MainView AutoOPERATOR PAS. In this case, the EXEC writes out the WTO but the earlier version MainView AutoOPERATOR PAS ignores the AREAILD parameter.

- The IDSTAMP parameter of the AOEXEC WTO command is enhanced to return in the resolved SSID (without including wildcard characters). The AOEXEC WTO IDSTAMP parameter returns different values depending on what value you specify with the IDSTAMP parameter and what version MainView AutoOPERATOR PAS process the EXEC.

  If you specify IDSTAMP(YES) with the AOEXEC WTO SS( ) | SSID( ) parameter, the ID stamp value in the WTO message is processed the same, regardless of whether the EXEC executes in a version 7.4 MainView AutoOPERATOR PAS or an earlier version. The ID stamp value in the WTO message is the literal value that is specified in the AOEXEC WTO SS( ) | SSID( ) parameter.

  If you specify IDSTAMP(SS) with AOEXEC WTO SS( ) | SSID( ) and the (ssid) keyword contains wildcard characters, a MainView AutoOPERATOR PAS 7.3.xx (and earlier) process the IDSTAMP value differently. The IDSTAMP value from MainView AutoOPERATOR PAS version 7.4 resolves to the SSID of the PAS instead of the literal value that is specified by the SS() keyword.

  You should evaluate the EXECs that use the AOEXEC WTO SS( ) | SSID( ) parameter and modify the EXECs with the correct IDSTAMP value based on your automation needs.

Examples:

```
"AOEXEC WTO TEXT('DABERG2 THIS IS A TEST') SS(GB3*) IDSTAMP(YES)"
DABERG2 THIS IS A TEST GB3* <- All releases

"AOEXEC WTO TEXT('DABERG3 THIS IS A TEST') SS(GB3*) IDSTAMP(SS)"
DABERG3 THIS IS A TEST GB3* <- A0 V7.3 and earlier
DABERG3 THIS IS A TEST GB3A <- A0 V7.4 and later
```
The IMFEXEC SUBMIT command is no longer documented in the MainView AutoOPERATOR documentation and MainView AutoOPERATOR version 7.4 has dropped support of this command.

You must replace occurrences of SUBMIT with the IMFEXEC JESSUBM command. You should review all data sets members that are allocated to the SYSPROC ddname of the MainView AutoOPERATOR PAS JCL, and locate any EXEC that includes the SUBMIT command. This includes using the IMFEXEC SUBMIT or SUBMIT commands.

You must also rewrite these EXECs to use the IMFEXEC JESSUBM command. For more information about the IMFEXEC JESSUBM command, see the MainView AutoOPERATOR Advanced Automation Guide.

What are the changes made to variables

The following table lists new variables added or updated for this release. These variables are documented in Chapter 4 of the MainView AutoOPERATOR Advanced Automation Guide and also in Chapter 7 of the MainView AutoOPERATOR Basic Automation Guide, Volume 1: Using Rules:

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMFJBCOR</td>
<td>contains the JES Job Correlation value</td>
</tr>
<tr>
<td>IMFMSGID</td>
<td>contains the first word of the message (up to 16 characters) with the leading plus sign (+) stripped if present</td>
</tr>
<tr>
<td>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>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>IMFOMUSR</td>
<td>contains user ID of application where the IMS Type-2 command originated. Valid only for IMP events.</td>
</tr>
</tbody>
</table>

The documentation in MainView AutoOPERATOR for MQ product is updated with new variables that support IBM MQ Version 8.
What DDNAMEs have changed

No changes were made to existing DDNAMEs in this release.

What parameter members have changed

The following table lists the new parameters, (including the default setting, and a brief description) that are included in MainView AutoOPERATOR version 7.4. You can locate the complete definition and additional usage documentation in this manual for each of the parameters in either of the following chapters:

- “Continuous operation with the Dynamic Parameter Manager” on page 69
  You can modify the parameter settings from the Dynamic Parameter Manager application and for most of the parameter changes, you can dynamically implement the changes.

- “BBPARM data set members for MainView AutoOPERATOR” on page 281
  You can manually edit the parameter settings in the BBPARM member.
Table 6: Added parameters for MainView AutoOPERATOR version 7.4

<table>
<thead>
<tr>
<th>BBPARM member name</th>
<th>Parameter</th>
<th>Brief description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAOPRM00</td>
<td>DBCTLORG=[NO</td>
<td>YES]</td>
</tr>
<tr>
<td></td>
<td>IMSAUTH=[NONE</td>
<td>USER]</td>
</tr>
<tr>
<td></td>
<td>IMSPLAUD=[NO</td>
<td>YES]</td>
</tr>
<tr>
<td></td>
<td>IMSOMEX=[NO</td>
<td>YES]</td>
</tr>
<tr>
<td></td>
<td>MVIMSDB=[MSG</td>
<td>CMD]</td>
</tr>
<tr>
<td></td>
<td>OR=[Y</td>
<td>N]</td>
</tr>
<tr>
<td>BBISSP00</td>
<td>IMAGJOB=[jobName</td>
<td>SSID=ssid</td>
</tr>
<tr>
<td></td>
<td>JRNLJOB=[jobName</td>
<td>SSID=ssid</td>
</tr>
<tr>
<td></td>
<td>QSMFIDV=[NO</td>
<td>YES]</td>
</tr>
</tbody>
</table>
What commands have changed

The following sections describes new or changed BBI control commands. See the documentation for BBI control commands in the *MainView Customization Reference* for more information.

**.CANCEL**

The BBI control command .CANCEL supports a new parameter, Q, that allows you to cancel queued EXECs. The syntax is as follows:

- `.C E eid` where `eid` is the EXEC identifier and cancels either a running or queued EXEC
- `.C E ename` where `ename` is the EXEC name and cancels either a running or queued EXEC
- `.C E ename Q` where `ename` is the EXEC name and cancels only a queued EXEC
- `.C E ename ALL` where `ename` is the EXEC name and cancels all EXECs identified by `ename`, both running and queued
- `.C E all Q` where `ename` is the EXEC name and cancels only queued EXECs identified by `ename`
- `.C E ALL` where `ename` is the EXEC name and cancels all queued and running EXECs identified by `ename`
- `.C E ALL Q` where `ename` is the EXEC name and cancels only queued EXECs identified by `ename`

**Note**

Use caution when specifying the ALL keyword because this command affects all of the EXECs that are running or queued at that point in time and they will all be canceled.

**.DISPLAY AR**

The BBI control command .DISPLAY supports a new keyword: AR and allows you to display in the BBI-SS Journal all of the Automation Reporter options that are currently active in the BBI-SS PAS.
What views or panels have changed

The following sections describe changes to views or panels made in this version.

- EXEC Manager panel: this view includes a new column **TTLQ**. This column shows the running total number of times the EXEC had to wait to be executed due to a lack of threads.

- AREXECs view: the **Queued Count** and **Max Queued** fields are re-located to the far right of the view. In their place, the view includes a new column, **TTLQ**. This field shows the number of times the during the interval that an EXEC waited on a queue for available execution threads.

Review any automation you might have from an earlier version that uses the IMFEXEC MV or IMFEXEC MVX commands for these views and review reports that might use these fields.

- All of the MainView AutoOPERATOR views that display log messages are updated to hybrid views.

  This change makes it easier for you to search and find log messages.

- New view, ARHEALTH: shows information and metrics using historical data about the PAS.

  This view allows you to track the trends of important PAS metrics, which can help you avoid outages.

These views and panels will be included in a later release of the documentation.

What messages have changed

No messages from were changed in this release.

Migrating MainView Total Object Manager (MVTOM)

Review the section, "Migrating from an earlier version of MainView TOM to version 2.3" in the *MainView Total Object Manager User Guide* for information about migrating a definition base from an earlier version to version 2.3.

Earlier versions of the TOMEXEC START command started objects without verifying the object's schedules or dependency relationships. In this version, the TOMEXEC START command checks the object's schedule and dependencies when attempting to start the object. The command also does not change an object's control mode (for example, SUSPEND). The TOMEXEC START can affect the objects whose status is UNLOCK, UNBLOCK, or RESET.
To retain the behavior of the TOMEXEC START command from earlier versions of MVTOM, you can specify START_FORCE=YES in BBPARM member MAMINIxx.

Earlier versions of the TOMEXEC STOP command stopped objects in top-to-bottom order (also called dependency order). In this version, the TOMEXEC STOP command stops objects in a bottom-to-top order. The default setting of the STOP_ORDER parameter in BBPARM member MAMINIxx is also changed from TOPTOBOTTOM to BOTTOMTOTOP or DEPENDENCY.

To retain the behavior of the TOMEXEC STOP command from earlier versions of MVTOM, you can specify STOP_ORDER=TOPTOBOTTOM in BBPARM member MAMINIxx.

What else has changed

The following sections describe other changes that are part of the MainView AutoOPERATOR 7.4 release.

Compatibility with MainView SYSPROG Services version 3.8

MainView AutoOPERATOR ships a set of utility EXECs that are tested and certified to work with the new release of MainView SYSPROG Services.

BMC recommends that you review these EXECs and migrate your automation to use these EXECs because they will protect your automation from changes made in the MainView SYSPROG Services component.

For more information about the utility EXECs, see the MainView AutoOPERATOR Advanced Automation Guide.

Run MainView AutoOPERATOR BBI-SS PAS in KEY4

BMC recommends specifying the IBM default ALLOWUSERKEYCSA(NO) in the SYS1.PARMLIB(DIAGxx) member for the LPAR where MainView AutoOPERATOR runs. This setting enables the BBI-SS PAS to use KEY4 as the default TCB key and storage key for increased protection from unauthorized programs.

To start the BBI-SS PAS in KEY4

1 Stop the BBI-SS PAS.

2 Start the BBI-SS PAS using SSLOAD as the EXECuted program with the START=FREE option specified.
The BBI-SS PAS will start, free any used storage, and stop again.

3 Change the BBI-SS PAS start JCL to execute SSLOAD4 (instead of SSLOAD):
   a Locate the EXEC statement in the JCL where PGM=SSLOAD is specified.
   b Modify the PGM name from SSLOAD to SSLOAD4.

4 Re-start the BBI-SS PAS.

The following figure shows an example of the edited JCL.

**Figure 3: Example of BBI-SS PAS start JCL with SSLOAD4**

```
//SSJCL    JOB                     // CHANGE - JOBCARD
//SSJCL    PROC PREFIX='XXXXX',    // CHANGE - PREFIX
 brightness 'WARM',               // REVIEW - WARM/COLD START OPTION
 //                             SS='SSAI',        // REVIEW - B&B SUBSYSTEM ID
 //                             VPOOL='NORESET',   // REVIEW - SHARED POOL OPTION
 //                             CSMALTDB='NO'     // REVIEW - CSM ALTERNATE DB

//**                                      USE OF PARAMETERS IN THIS JCL *
//** EXEC PGM=SSLOAD4,REGION=512M,DPRTY=(15,0),TIME=1440,
//**   PARM='&SS,&START,&VPOOL,&CSMALTDB'
//**
```

This change also can be made at an IPL by changing the program executed in the BBI-SS PAS JCL from SSLOAD to SSLOAD4 before you re-IPL.

**Other changes**

The following table describes some other changes that are part of MainView AutoOPERATOR version 7.4.

**Table 7: Additional changes made in MainView AutoOPERATOR version 7.4**

<table>
<thead>
<tr>
<th>Additional changes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBPARM member AAOARP00 includes a new parameter, AROPTS</td>
<td>Allows you to specify one or more Automation Reporter collectors to be activated. For more information about how to use the parameter, refer to the comments in BBPARM member AAOARP00. A future release of the MainView AutoOPERATOR documentation will include information about this parameter.</td>
</tr>
</tbody>
</table>
### Additional changes

<table>
<thead>
<tr>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>BBPARM member AAOTRN00 includes a new parameter, USERID.</td>
<td>Allows you to submit IMS transactions with the SAF user ID associated with the MainView AutoOPERATOR PAS. For more information about how to use the parameter, refer to the comments in BBPARM member AAOTRN00. A future release of the MainView AutoOPERATOR documentation will include information about this parameter.</td>
</tr>
<tr>
<td>The EXEC Management panel supports the RESET primary command that builds a new list of EXECs.</td>
<td>For more information about how to use the command, refer to the online help for the EXEC Management panel. A future release of the MainView AutoOPERATOR documentation will include information about this parameter.</td>
</tr>
<tr>
<td>Security update for invoking IMS and IMSplex commands</td>
<td>The <em>MainView Security Reference Guide</em> contains an update for preventing IMS generic commands from failing when the default RACF class is defined with UACC(NONE).</td>
</tr>
<tr>
<td>Two new enhanced security features: FEATURE=CONSCMD and FEATURE=BBIDISP.</td>
<td>FEATURE=CONSCMD allows you to secure user access to issuing BBI commands at the console. FEATURE=BBIDISP allows you to secure access to BBI action commands, but allow users to issue BBI display commands. See the <em>MainView Security Reference Guide</em> for information about these security features.</td>
</tr>
</tbody>
</table>

### Documentation name changes

There are no documentation name changes in this release.

### What PTFs should I apply

If you install MainView AutoOPERATOR 7.4 in the same environment where earlier versions of MainView AutoOPERATOR are running, you might need to apply additional PTFs to the earlier MainView AutoOPERATOR BBI-SS PASs.

Refer to the MainView AutoOPERATOR version 7.4 Release Notes for a complete list of any PTFs that you need to apply.
How to secure MainView AutoOPERATOR

Just as with previous versions of MainView AutoOPERATOR, you can implement security for the 7.4 version through an External Security Manager (ESM). While you can continue to use BMC Software proprietary security, you might want to control security of MainView AutoOPERATOR through your ESM.


The MainView Security Guide and the MainView Security Reference manuals contain documentation for new security items and resource names for the enhancements and revisions made in MainView AutoOPERATOR 7.4.

How much CPU does this release use

This release does not use a significantly higher amount of CPU than the previous release.

Does this version of MainView AutoOPERATOR use common or private storage differently

This version of MainView AutoOPERATOR does not use common or private storage differently than earlier versions.

Does this version communicate with earlier versions

Any BBI-SS PAS with an activated MainView AutoOPERATOR 7.3.xx product can schedule an ALERT or EXEC to any MainView AutoOPERATOR 7.4 BBI-SS PAS. In addition, a MainView AutoOPERATOR 7.3.xx terminal session can access MainView AutoOPERATOR 7.4 panels.

However, MainView AutoOPERATOR commands (or Rules) that schedule EXECs or create ALERTs that target a version 7.3.xx BBI-SS PAS from an version 7.4 BBI-SS PAS cannot contain any of the new keywords or features that are introduced in MainView AutoOPERATOR 7.4. MainView AutoOPERATOR rejects those commands before the commands are sent. When this situation occurs, depending on how the command is initiated, MainView AutoOPERATOR might issue a message.

A good approach is to install the new version of MainView AutoOPERATOR onto a test system and use it to review the new features and functions. Then you can
migrate the new version of MainView AutoOPERATOR to your other systems as your time and schedules permit.

**Does version 7.3.xx communicate with version 7.4**

ALERTs, EXECs, and Rules can be sent from a MainView AutoOPERATOR 7.3.xx BBI-SS PAS to a MainView AutoOPERATOR 7.4 BBI-SS PAS.

**What other BMC Software products run with this version of MainView AutoOPERATOR**

MainView AutoOPERATOR version 7.4 works with other currently supported versions of MainView products.

**Which IBM software does this version of MainView AutoOPERATOR require**

This version of MainView AutoOPERATOR works with supported IBM software versions; the minimum levels are

- z/OS version 1.13
- CICS Transaction Server version 3.2
- IMS version 12
- Tivoli NetView for z/OS version 5.3
- IBM MQ for z/OS version 7.0.1

**What functions will not be supported in future versions**

BMC has identified the following MainView AutoOPERATOR solutions from the MainView AutoOPERATOR for MQ option as items for which support will be discontinued in a future version:

- The Automation Power Line solution (QMQPOWER)
- The Basic Intercommunication Solution
Installation considerations for MainView AutoOPERATOR 7.4

This section contains technique-specific installation considerations for migrating from previous versions of MainView AutoOPERATOR to MainView AutoOPERATOR version 7.4.

Getting new passwords

MainView AutoOPERATOR version 7.4 uses BMC Software Product Authorization passwords. For a description of the Product Authorization utility and information about obtaining and applying passwords, see the Installation System User Guide.

WARNING
BBKEYS are no longer supported for MainView AutoOPERATOR version 6.4 and later.

If you do not receive a letter that contains the BMC V3 Passwords for MainView AutoOPERATOR 7.4 with your product package, see the Licensing and Passwords information on the BMC Customer Support web site.

After the new passwords are available, refer to the Installation System User Guide for information about how to install and activate the passwords.

The following table shows how the product component name (AOOPTION) for earlier MainView AutoOPERATOR versions correlates to the new password codes.

<table>
<thead>
<tr>
<th>MainView AutoOPERATOR product component name</th>
<th>MainView AutoOPERATOR AOOPTION</th>
<th>MainView AutoOPERATOR password product code</th>
</tr>
</thead>
<tbody>
<tr>
<td>MainView AutoOPERATOR Access NV</td>
<td>ANV</td>
<td>BKG</td>
</tr>
<tr>
<td>MainView AutoOPERATOR TapeShare</td>
<td>TSH</td>
<td>BCG</td>
</tr>
<tr>
<td>MainView AutoOPERATOR for CICS</td>
<td>CAO</td>
<td>BCC</td>
</tr>
<tr>
<td>MainView AutoOPERATOR for IMS</td>
<td>IAO</td>
<td>BCD</td>
</tr>
<tr>
<td>MainView AutoOPERATOR for MQ</td>
<td>QAO</td>
<td>BCI</td>
</tr>
<tr>
<td>MainView AutoOPERATOR for z/OS</td>
<td>MAO</td>
<td>BCE</td>
</tr>
</tbody>
</table>

Even if you are migrating from an earlier version of MainView AutoOPERATOR or changing from one CPU to another CPU, you must acquire new BMC Software
Product Authorization passwords to use any version MainView AutoOPERATOR version 6.4 or later.

If you want to use these passwords in multiple PASs, but you want only certain product options to be active within a specific PAS, you can install the individual MainView AutoOPERATOR component passwords into separate password library data sets or use the AOOPTION parameter in BBPARM member AAOPRMxx to limit the components that are activated in the PAS.

After this process is complete, concatenate together the password library data sets for the specific product options that you want active for the PAS. For example, your concatenation might look like the following example where the two product components that are activated for this PAS are:

- MainView AutoOPERATOR for CICS (CAO)
- MainView AutoOPERATOR for z/OS (MAO)

```
//BMCPSWD DD DISP=SHR,DSN=AAO.BMCPSWD.CAO (CAO ONLY)
//         DD DISP=SHR,DSN=AAO.BMCPSWD.MAO (MAO ONLY)
```

In the following example, the first line of the concatenation shows that all of the MainView AutoOPERATOR product components, with the exception of TapeSHARE, are included. This exception is because you can run TapeSHARE in only one PAS per LPAR (the first one that is started with that password). Therefore, you should include the TapeSHARE password in the JCL of only one BBI-SS PAS.

**Figure 4: Example of specifying passwords for all MainView AutoOPERATOR product components**

```
//BMCPSWD DD DISP=SHR,DSN=AAO.BMCPSWD (ALL AAO No/TapeSHARE)
// ** DD DISP=SHR,DSN=AAO.BMCPSWD.ALL (ALL AAO)
// ** DD DISP=SHR,DSN=AAO.BMCPSWD.ANY (ANY AAO)
// ** DD DISP=SHR,DSN=AAO.BMCPSWD.CAO (CAO ONLY)
// ** DD DISP=SHR,DSN=AAO.BMCPSWD.IAO (IAO ONLY)
// ** DD DISP=SHR,DSN=AAO.BMCPSWD.MAO (MAO ONLY)
// ** DD DISP=SHR,DSN=AAO.BMCPSWD.MQS (MQS ONLY)
// ** DD DISP=SHR,DSN=AAO.BMCPSWD.SHA (SHA ONLY)
// ** DD DISP=SHR,DSN=AAO.BMCPSWD.TSH (TSH ONLY)
```

To create a universal JCL, you can use symbolics with the &SSID within the data set name. Parameters that are passed to the PROC statement allow you to select one set of passwords to use; see the following example:

**Figure 5: Example of using symbolics for passwords for MainView AutoOPERATOR product components**

```
//BMCPSWD DD DISP=SHR,DSN=AAO.&SS..BMCPSWD (Set for this PAS)
```

When the SS PAS JCL passes the parameter SS=SYSA to the PROC, the example shows that AAO.&SS..BMCPSWD resolves to AAO.SYSA.BMCPSWD, which is the data set that contains the passwords to use for PAS.
Re-implement JES2MAO exit

If you are migrating by a using Standard installation with the BMC Installation System and you are replacing any MainView AutoOPERATOR version, and all of the following statements are true, you must implement the JES2MAO exit again:

- Your site is licensed for the MainView AutoOPERATOR for z/OS option.
- You elected to implement the JES2MAO exit.
- The JES2MAO exit is linked to the existing system’s BBLINK library.

Refer to “Displaying MVS JES2 job numbers in address space application” on page 248, for additional information about the JES2MAO exit.

Installing MainView AutoOPERATOR into an existing zone that includes other BMC products

Ensure all of the other BMC products in the target zone are currently supported versions before installing MainView AutoOPERATOR into the same zone. MainView AutoOPERATOR does not support coexistence with earlier, unsupported versions of BMC products.

MainView AutoOPERATOR also requires that all products running within the same BBI-SS PAS are installed into the same zone and executed out of the same libraries. MainView AutoOPERATOR does not support concatenation of product libraries in BBILOAD or STEPLIB.

When MainView AutoOPERATOR is installed into a zone with other products, it is important to note that MainView AutoOPERATOR shares the use of some FMIDs with other products. Therefore, installing a new version of one product can impact other products. For example, when installing a new version of MainView AutoOPERATOR into a zone where both MainView AutoOPERATOR for CICS and MainView for CICS are running, the new version MainView AutoOPERATOR version might impact MainView for CICS.
Implementing Rules Management

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.

Overview

The Rules Management application is a windows-mode editor that supports creating and managing Rules that are stored in a registry.

You can access the Rules Management editor from the MainView Explorer interface from a web browser or from the Automation Menu. The Rules Management application allows you to create Rules that you can also easily control throughout your system. The following sections define terms and concepts that you should understand before implementing Rules via the Rules Management application.

Rules Registry

The Rules registry is a Runtime Component System (RTCS) registry that is used as a repository for Rules-specific information.

For more information about RTCS, see the *BMC Runtime Component System Configuration and Administration Guide*. 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.

AOPLex

An AOPLex is a collection of all of the MainView AutoOPERATOR PASes that share a single Rules registry.
You can have a single MainView AutoOPERATOR PAS or hundreds of PASes sharing a registry. Therefore, if you have Rules automation where multiple PASes would use the same Rules and Rule Sets, you should consider creating (or converting) and managing your Rules with the Rules Management application.

**Deployment**

You can deploy (or send) Rules and Rule Sets to the MainView AutoOPERATOR PASes that need access to them from a single registry.

This means that you can deploy Rules and Rule Sets from a single registry to one or many PASes within an AOPLex as well as deploy Rules and Rule Sets from one AOPLex to another AOPLex that is connected to the same CASPLex. You can control and manage the level of registry sharing within and between AOPLexes in as centralized or de-centralized way as your system requires.

**Implementation considerations**

This section describes important concepts that will help you determine how to implement Rules with the Rules Management application.

**Prerequisites**

The following list describes prerequisites for using the Rules Management application.

- **CAS**
  You must have a MainView Infrastructure Coordinating Address Space (CAS) installed that is started and connected to the MainView AutoOPERATOR PAS before using the user address space (UAS) to access windows-mode views to manage the contents of the registry. For information about how to install and customize a CAS, see the *MainView Customization Reference*.

- **HFS path**
  The MainView AutoOPERATOR PASes and CASes require update access to the MainView common registry when using the UAS to manage the registry contents. The MainView common registry is an HFS or zFS directory structure whose top level is defined to the CAS through the HFSPATH view. For more information about using the HFSPATH view, see the *MainView Administration Guide*. 
RTCS
You must have an RTCS address space installed, started, and active before starting a MainView AutoOPERATOR PAS. To install RTCS, see the BMC Runtime Component System Configuration and Administration Guide. For additional information about how to perform additional RTCS customization tasks, see the MainView Customization Reference.

Run MainView AutoOPERATOR in Key 4
You must start the MainView AutoOPERATOR PAS in KEY4. If the PAS detects the specification of ALLOWUSERKEYCSA(NO) in the LPAR, then MainView AutoOPERATOR will start in KEY4.
Otherwise, to do this, you must update the BBI-SS PAS JCL to execute program SSLOAD4 instead of SSLOAD. For more information about starting the MainView AutoOPERATOR PAS in key4 see the chapter "Run MainView AutoOPERATOR BBI-SS PAS in KEY4" in the MainView AutoOPERATOR Customization Guide.

Additional Considerations

In addition, you should be aware of the following considerations when deciding whether to implement Rules Management.

Security considerations
See the MainView Security Reference Manual for information about implementing security by using the latest version of MainView Infrastructure resource classes for accessing the Rules editor views and dialogs and taking actions from the views.

Continuous State Manager application is disabled
You cannot use the Continuous State Manager application in MainView AutoOPERATOR PASes that use a registry for Rules Management.

Full-screen Rules and Rule Sets are disabled
When using the Rules Management application in FULL registry mode, all Rules and Rule Sets are located in the registry during MainView AutoOPERATOR PAS initialization. The BBIPARM DDNAME concatenation data sets are not used as the source for Rules and Rule Sets and the full-screen Rules editor is disabled.
Migrating Rules created earlier to Rules Management registry-based Rules

The Rules Management application provides a utility that imports Rules and Rule Sets created in earlier versions of MainView AutoOPERATOR in full-screen mode from your BBIPARM DDNAME concatenation data sets into the registry.

You can choose from a number of implementation options for importing the BBPARM Rule Sets based on your needs and system configuration. For more information on the import utility, see Chapter x - Migrating from BBPARM Rules to Rules registry in the MainView AutoOPERATOR Basic Automation Guide Volume I: Using Rules.

Registry considerations

This section describes how to define a registry and how it is organized.

Registry allocation

Use the sample JCL found in BBSAMP member AAOAGSAL to allocate a registry.

The JCL includes comments at the top that you should review to determine the size of the registry and describes the other JCL modifications you will need to make. Submit the JCL to create the VLDS VSAM data set and specify the data set name of the registry in BBPARM member AAOPLXxx with the REGISTRY_DSN parameter.

Registry organization

The registry is organized into three areas:

- PAS definition (PASDef)
  The PASDef is the part of the registry that controls aspects of Rules and Rule Sets the way some of the definitions in BBPARM member AAOPRMxx controls full-screen mode Rules. For example, the PASDef defines which Rule Sets are available to a MainView AutoOPERATOR PAS, the automation strategy that a Rule Set has when it is started (FIRST, ALL, INDIVIDUAL, and so on), and the automation status.
  You can define a single PASDef to be shared by as many PASes as needed. So, for example if your system has one PAS on each of five different LPARs and you...
want all the PASes to have the same Rules and Rule Sets, you create one PASDef and define the names of the five PASes that will use it.

To define a new Rule Set for each of the five PASes, you can add the Rule Set to the PASDef and deploy the PASDef (with the new Rule Set) to all five LPARs with a single command.

- **Setbase**
  A setbase is a collection of Rule Sets. A setbase might include Rule Sets for a single application, a single geographical location, or a business unit of your company. The setbase can include Rules Sets for anything that makes sense in your environment. If you do not need to organize Rule Sets this way, you can put all your Rule Sets into the supplied default setbase. You can implement multiple setbases at anytime.

- **Rule Pools**
  When Rules were created in full-screen mode, a Rule is a part of a Rule Set, which is a member of a partitioned dataset (PDS). With this structure, you can not share a Rule between multiple Rule Sets to meet different requirements. You can only copy a Rule into multiple Rule Sets and then maintain them individually. With the registry, Rules are now stored in groups that are called Rule Pools. With these registry-based Rules, you create a single Rule and assign it to multiple Rule Sets but the Rule itself is not duplicated. Assigning a single Rule to multiple Rule Sets eliminates the need to individually maintain a copied Rule.

### Parameters required for Rules Management

BBPARM member AAOLXLxx contains parameters, some of which you must modify, when you use the Rules Management application.

To use Rules Management, you must specify values for all of the required parameters to enable the registry and the AOPLex.

---

**Note**

If you are not going to use the Rules Management application, ensure that the `USAGE_MODE_RULES=` parameter is set to `OFF`. When `OFF` is specified, you do not need to modify any of the parameters in BBPARM member AAOLXLxx.

You cannot use the Dynamic Parameter Manager (DPM) application to edit the AAOLXLxx member. You can change some settings without restarting the PAS from the AOPLEXP view. Otherwise, you must make changes to the AAOLXLxx parameters using ISPF EDIT (or an equivalent editor) and restart the MainView AutoOPERATOR PAS to activate your changes.
The AOPLEXP view also includes a hyperlink on the Parameter Name column to jump to the PASPARM view where you can edit the member using ISPF (or an equivalent editor) and save the changes for the next PAS restart.

The AOPLEXP view allows you to change the following debugging options dynamically while the PAS is running and therefore, do not require restarting the MaiNView AutoOPERATOR PAS:

- AGSLOG
- AGSLOG_FAILOVER
- FORCEWTO
- AGSJRNL
- FORCE_CLASSIC_RSNAMES

Note that any changes made using this view are temporary and are reset to their default values when the PAS is restarted (WARM or COLD).

**Parameters description**

The following table lists and describes BBPARM member AAOPLXxx parameters.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AO_NAME</td>
<td><em>(required)</em> provides the name to identify this PAS within the AOPlex. This name is used when identifying the PAS in the Valid PAS List and in AGS messages and reports. This value must be 8 characters long. The name can contain only alphanumeric characters. No special characters are allowed. Example: AO_NAME=AGSPROD1</td>
</tr>
<tr>
<td>AOPLEX_NAME</td>
<td><em>(required)</em> provides the name of the AOPlex to which this PAS belongs. This name must be the same for all PASes that share the same registry. This value must be 8 characters long. This name can contain only alphanumeric characters. No special characters are allowed. Example: AOPLEX_NAME=AOPLEX01</td>
</tr>
<tr>
<td>AOPLEX_XCF_GROUP</td>
<td><em>(required)</em> provides the name of the XCF group that the AOPlex will use to communicate. This name must be the same for all PASes that share the same registry dataset. This value must be 8 characters long. This name can contain only alphanumeric characters. No special characters are allowed. This value cannot be the same as AOPLEX_NAME and cannot be the same XCF group name specified in the XCFGROUP= parameter in BBPARM member BBISSPxx. Example: AOPLEX_XCF_GROUP=AOPLXCF1</td>
</tr>
<tr>
<td>Parameter name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| REGISTRY_DSN           | *(required)* is the data set name of the linear VSAM data set allocated for use by this AOPlex. This value can be 1 to 44 characters long.  
                                                                 |                                                                                                                                                                                                                                                                                                                                          |
| REGISTRY_WARNHWM       | *(optional)* is the Registry high-water-mark percentage warning message threshold  
                                                                 | If specified with a non-zero value, the AGSRY0014W warning message is issued when the percentage of the Registry space that is formatted for use exceeds the value and either of the following situations is true:  
                                                                 |  
                                                                 | - The PAS is being initialized  
                                                                 |  
                                                                 | - The registry 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 AGSRY0014W was issued.  
                                                                 | Specifying a value of 0 means that no monitoring will be done and no messages are issued. This value can be 0 to 99. If you do not specify a value, the default value is 0.                                                                 |
| USAGE_MODE_RULES       | *(required)* informs the PAS how to use the registry  
                                                                 | Valid values are:  
                                                                 | OFF: the MainView AutoOPERATOR PAS will not use a registry for Rules in any way. Rules Management is not functioning for the PAS. Rules are read from the BBIPARM DD concatenation.  
                                                                 | ADMIN: the MainView AutoOPERATOR PAS opens the registry and allows you to perform administrative tasks, which the registry records. However, MainView AutoOPERATOR is not using the Rules from the registry for any automation in the PAS. Rules are read from the BBIPARM DD concatenation.  
                                                                 | FULL: the MainView AutoOPERATOR PAS uses the Rules from the registry, allows you to perform administrative tasks on the registry, and also reads from the registry to supply all the Rules and Rule Sets for automation in the PAS. No Rules are used from the BBIPARM DD concatenation and the full-screen Rules editor is disabled.  
                                                                 | Note: When switching to FULL mode, you must perform a COLD start of the PAS to access all of the Rules from the registry.  
<pre><code>                                                             | The default is OFF.                                                                                                                                                                                                                                                                                                                        |
</code></pre>
<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Description</th>
</tr>
</thead>
</table>
| AGSLOG              | *(optional)* specifies whether the MainView AutoOPERATOR Generalized Services (AGS) messages are sent to a logger.  
Valid values are:  
**YES**: (recommended) messages are sent to a logger. If the logger becomes unavailable for any reason, and AGSJRN=NO and FORCEWTO=NO are also specified, the value on the AGSLOG_FAILOVER parameter determines what happens to the message.  
**NO**: messages are not be sent to a logger.  

| AGSLOG_FAILOVER     | *(optional)* specifies what happens to AGS messages when AGSLOG=YES, FORCEWTO=NO, and AGSJRN=NO are specified and the logger is not available.  
Valid values are:  
**JRNL**: send the message to the BBI-SS journal. JRNL is the default.  
**WTO**: send the message as a WTO to the PAS JES joblog.  
**NONE**: do not write any messages to any destination.  

| FORCEWTO            | *(optional)* specifies whether AGS product messages are sent to the PAS JES job log.  
Valid values are:  
**YES**: messages are sent to the PAS JES job log.  
**NO**: messages are not sent to the PAS JES job log. NO is the default.  

| AGSJRN              | *(optional)* specifies whether AGS product messages are sent to the BBI-SS journal.  
Valid values are:  
**YES**: messages are sent to the BBI-SS journal.  
**NO**: messages are not sent to the BBI-SS journal. NO is the default.  

| FORCE_CLASSIC_RSNAMES | *(optional)* specifies whether Rule Sets created in the registry by the user address space (UAS) follow the classic Rule Set naming that requires the name to start with RUL or AAORUL.  
Valid values are:  
**YES**: the names of Rule Sets created in the registry by the UAS are restricted to a Rule Set naming standard and the Rule Set name must begin with RUL or AAORUL. This ensures these Rule Sets can be exported and used in a BBPARM data set without a registry. YES is the default.  
**NO**: the names of Rule Sets created in the registry by the UAS are not restricted to start to begin with RUL or AAORUL. Therefore, these Rule Sets cannot be used as BBPARM Rule Sets even if you export them and place them in a valid BBPARM data set unless you manually rename them to use RUL and AAORUL.  

---

Parameters required for Rules Management
### Connecting a registry to a MainView AutoOPERATOR PAS

The section describes what you must know about connecting the registry to a MainView AutoOPERATOR PAS.

Once a registry data set is created, you must make it available to one or more MainView AutoOPERATOR PASes by customizing the required parameters in BBPARM member AAOPLX00. The default suffix for the member is 00. If you want to use an alternate suffix, copy the member and supply a 2-character suffix. Remember to add the suffix to BBPARM member BBICFG for the PAS.

#### ADMIN mode

In addition to customizing the required keywords, you must specify a value for the USAGE_MODE_RULES= parameter.

Initially, you should specify a value of ADMIN to allow the PAS to access the registry but not actually use the registry for its automation Rules. In ADMIN mode, the windows-mode views will be active to allow Rules Management actions such as importing existing Rule Sets from BBPARM data sets and creating the PAS Definition (PASDef) for the PAS.

#### FULL mode

Once everything is imported and configured, you need to change the USAGE_MODE_RULES= keyword to FULL in BBPARM member AAOPLXxx and COLD start the MainView AutoOPERATOR PAS.

---

### Parameter name | Description
---
| AUDIT_STARTUP_RULESETS | *(optional)* causes message AGSIA4050I to be generated to the PAS syslog during a PAS COLD start to list the rulesets that the PAS will use and is similar to the PROCESS LIST= parameter in BBPARM member AOPRMxx

Valid values are:
- **YES**: produce the AGSIA4050I message(s) during PAS COLD start.
- **NO**: do not produce the AGSIA4050I message(s) during PAS COLD start.
At that point, the PAS would be totally relying on the Rules Management registry as the repository for automation Rule Sets and Rules. BBPARM data sets are no longer be used as the source for Rule Sets and Rules when USAGE_MODE_RULES=FULL is specified in BBPARM member AAOPLEXxx.

Running in FULL mode also deactivates the full-screen Rules editor.
Generating early or interim WTOs

This section describes how to automate write-to-operator messages (WTOs) that are issued:

- After IPL but before MainView AutoOPERATOR is started. These are called early WTOs.
- While MainView AutoOPERATOR is not active (such as when you restart MainView AutoOPERATOR with a WARM start). These are called interim WTOs.

Note
This requires that you have BMC MainView Automation (product code MVN) active in the MainView AutoOPERATOR BBI-SS PAS.

MainView AutoOPERATOR sees the captured WTOs as HWTO (Historical WTO) events. You can use Rules to perform delayed automation with these events. HWTO event Rules are similar to MSG events. MSG events are WTOs that occur while MainView AutoOPERATOR is active. The Rule Processor sees WTOs that are collected when MainView AutoOPERATOR is briefly inactive as history (or HWTO) events.

For information about creating Rules for HWTO events, see the MainView AutoOPERATOR Basic Automation Guide Volume 1: Using Rules.

Early WTOs

You can capture WTOs that are issued during IPL and before MainView AutoOPERATOR starts by:

- Implementing QAOCNZ15 as the MVS CNZ_MsgToSyslog exit.
  For more information about implementing this exit, see the IBM manual MVS Installation Exits.

- You must also specify WTOEARLY=Y in BBPARM member AAOPRM.xx.
Automating early WTOs can help detect situations such as product password expiration warnings or errors for other software products, JCL errors in STCs that are started before MainView AutoOPERATOR starts, or significant messages that occur early in the MVS initialization.

**Note**
The timing of HWTO events will be different from the original WTOs (which the Rule Processor sees as MSG events). All the HWTO events are generated at once at BBI-SS PAS startup. For example, IEF403I and IEF404I messages for a job occur instantly without any intervening delay as though the job had taken no time to run. You must consider this when planning to automate HWTOs.

The QAOCNZ15 exit stores the early WTOs in a data space. The data space can store about 100,000 WQEs. Note several WQEs can be associated with a single multi-line WTO. When this data space is full, it issues the following informational message and QAOCNZ15 is deleted from the MVS CNZ_MsgToSyslog exit:

```
AA0501I QAOCNZD DATA SPACE IS FULL ptf_level
```

When implementing the exit, the QAOCNZ15 load module can reside in any of the following places:

- In `highLevelQualifier.BBLINK`
- In a special user library that contains only QAOCNZ15
- In SYS1.PALIB
- From a special user library that is concatenated with SYS1.PALIB

One example of specifying the EXIT statement in SYS1.PALIB PROGxx member and using QAOCNZ15 directly from the BBLINK data set.

```clike
EXIT ADD EXITNAME(CNZ_MSGTOSYSLOG) MODNAME(QAOCNZ15)
DSNAME(highLevelQualifier.bblink)
```

If you are using the DSNAME(`highLevelQualifier.bblink`) option, verify that the PTF for IBM APAR OA49360 is applied. This fixes a bug where exits loaded from DSNAME() are not effective.

**Note**
You must review the CNZ_MsgToSyslog exit after any PTF maintenance, or installing or upgrading any MainView products.

Because there is only one copy of QAOCNZ15 in MVS, when you install new MainView libraries or apply PTFs, remember to ensure you always run the newest version and PTF level of QAOCNZ15. This ensures that the QAOCNZ15 exit is compatible with all levels of MainView AutoOPERATOR.
The following WTO AU0500I is issued when QAOCNZ15 starts and it indicates the PTF level of the code:

AA0500I CREATING QAOCNZD DATA SPACE ptf_level

To see if QAOCNZ15 is active, issue the following command:

D PROG,EXIT,EN=CNZ_MsgToSyslog,DIAG

You can also use the following MVS SETPROG EXIT command to dynamically ADD this exit as ACTIVE or INACTIVE:

SETPROG EXIT,ADD,EXITNAME=CNZ_MsgToSyslog,STATE=ACTIVE, MODNAME=QAOCNZ15, DSN=data_set_which_contains_this_member

If necessary, you can then use the following MVS SETPROG EXIT command to DELETE the exit:

SETPROG EXIT,DELETE,EXITNAME=CNZ_MsgToSyslog,MODNAME=QAOCNZ15

Note

BMC recommends against concatenating any BMC-supplied load library with SYS1.PALIB, or inserting a BMC-supplied load library into the MVS Linklist. If you wish to load the QAOCNZ15 program from the SYS1.PALIB concatenation, copy only the QAOCNZ15 load module to a new small user library and concatenate this special library with SYS1.PALIB.

Interim WTOs

You can also capture WTOs that are issued during planned outages of MainView AutoOPERATOR by specifying WTOINTRM=Y in BBPARM member AAOPRMxx.

Interim WTOs are collected from the time when the MainView AutoOPERATOR PAS ASID terminates through shortly after the MainView AutoOPERATOR BBI-SS PAS re-starts. This includes about 5 WTOs at the very start of the MainView AutoOPERATOR BBI-SS PAS initialization. You cannot trap 100% of the messages during termination through initialization of the BBI-SS PAS that is collecting interim WTOs. The Rule Processor sees these interim WTOs as HWTO events.

WTOINTRM=Y must be specified before the BBI-SS PAS shuts down and it must also be specified before the BBI-SS PAS starts up.

Automating interim HWTO events can help manage situations which you know might occur during planned MainView AutoOPERATOR outages. For example, suppose you plan a brief planned outage where MainView AutoOPERATOR is
recycled to pick up new maintenance. When WTOINTRM=Y is specified, you can perform delayed automation on WTOs that are issued while MainView AutoOPERATOR was not active.

The BBI-SS PAS issues the following message indicating that it will collect interim WTOs:

AA3002I WILL CAPTURE INTERIM WTOS FOR LATER HISTORICAL PROCESSING

The CSA table that collects interim WTOs can collect about 100,000 WTOs. When the table is full, no more WTOs are collected and the following message is issued:

DX0100I MainView AutoOPERATOR INTERIM WTO TABLE IS FULL xxx

The CSA table is deleted after the BBI-SS PAS is restarted and the HWTOs are passed to the Rule Processor. The following message is issued:

AU0320I REPLAYED nnnn INTERIM WTOS

**Note**

If you specify a COLD start, the captured interim WTOs are lost. The COLD start frees CSA (including CSA used to store interim WTOs). A COLD start may also be forced by applying certain BMC PTFs.

In addition, if the MainView AutoOPERATOR BBI-SS PAS terminates abnormally, the interim WTO feature is not available. The ability to save interim WTO runs as part of a normal BBI-SS PAS termination.
Implementing support for EOS and TLM events

MainView AutoOPERATOR Rules Processing provides support for end-of-step (EOS) and Time Limit (TLM) events.

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.

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.

You must implement support for the EOS and TLM event types by specifying system exits in the SMFPRMxx member of SYS1.PARMLIB.

Specifying the IEFACTRT exit provides support for the EOS event type and the IEFUTL exit provides support for the TLM event type. If you do not specify both exits, you might receive unexpected messages during BBI-SS PAS startup.

See Figure 6 on page 67 for an example of the SMFPRMxx parameter member with the required exits shown in bold. You should also ensure that the required exits are listed in the EXITS() parameter and verify that the SMF type 30 records are being generated. Without the type 30 records, the EOS event will not function correctly.

Figure 6: Example of the SYS1.PARMLIB member SMFPRMxx

```
SYS(NOTYPE(43,45,47:49,52:58,92(10:11),99),
  EXITS(IEFU83,IEFU84,IEFU85,IEFACTRT,IEFUJV,
       IEFUSI,IEFUJ1,IEFUTL,IEFU29),
  INTERVAL(SMF,SYNC),NODetail)
```
SUBSYS(STC, NOTYPE(43,45,47:49,52:58,92(10:11),99), EXITS(IEFU29, IEFU83, IEFU84, IEFU85, IEFACRT, IEFUJ, IEFUSI, IEFUJI, IEFUTL), INTERVAL(SMF, SYNC), DETAIL)

**Note**

To deactivate the IEFACRT exit for any reason, specify DYNEXITS=N in BBPARM member BBIISPxx. When you specify N, MainView AutoOPERATOR bypasses activating the exit and it will also disable the EOS and TLM Rule event types from occurring.
Continuous operation with the Dynamic Parameter Manager

The Dynamic Parameter Manager (DPM) is a panel-driven application where you can view or modify parameters in BBPARM members from a TS session. The BBPARM members currently supported by DPM are as follows:

<table>
<thead>
<tr>
<th>BBPARM member name</th>
<th>Contains parameters to set</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAOALSxx</td>
<td>MainView AutoOPERATOR ALERT thresholds</td>
</tr>
<tr>
<td>AAOARPxx</td>
<td>MainView AutoOPERATOR Automation Reporter parameters</td>
</tr>
<tr>
<td>AAOEXPxx</td>
<td>MainView AutoOPERATOR EXEC parameters</td>
</tr>
<tr>
<td>AAOPRMxx</td>
<td>MainView AutoOPERATOR general function parameters</td>
</tr>
<tr>
<td>AAOTSPxx</td>
<td>TapeSHARE for MainView AutoOPERATOR parameters</td>
</tr>
</tbody>
</table>

Additionally, the DPM application has an Activate command that can be used to dynamically change the BBPARM members that a BBI-SS PAS currently uses without restarting the BBI-SS PAS.

Using the Dynamic Parameter Manager: overview

With the Dynamic Parameter Manager, you can

- control virtual storage consumption
  Use DPM to dynamically adjust the virtual storage throttles in AAOEXPxx and AAOALSxx (where xx is a user-defined suffix).

- control CPU & I/O usage
  Use DPM to adjust the CPU & I/O throttles dynamically in AAOEXPxx.
implement previously defined AAOALSxx, AAOARPxx, AAOEXPxx, AAOPRMxx, and AAOTSPxx members dynamically

By implementing previously defined members dynamically, different resource thresholds can be defined in response to anticipated environmental requirements without restarting the BBI-SS PAS.

activate and control the data collection time intervals for the Automation Reporter application

The Automation Reporter collects data for specific automation items and the AAOARPxx member controls how often data is collected and offloaded into the database.

The Automation Reporter application (and how to activate it with the Dynamic Parameter Manager) is described in the MainView AutoOPERATOR Basic Automation Guide, Volume 1: Using Rules.

customize the way the TapeSHARE for MainView AutoOPERATOR component performs tape device sharing automation among your z/OS images

With MainView AutoOPERATOR TapeSHARE, you can automate the sharing of tape devices between z/OS images. For more information about installing TapeSHARE, refer to “Implementing TapeSHARE” on page 127; for more information about using TapeSHARE, refer to the MainView AutoOPERATOR Options User Guide.

Accessing the Dynamic Parameter Manager

Choose Option 9, Dynamic Parameter Manager, from the PRIMARY OPTION MENU (shown in the following figure) to see a list of the AAOALSxx, AAOARPxx, AAOEXPxx, AAOPRMxx, and AAOTSPxx members in BBPARM.

Figure 7: MainView AutoOPERATOR PRIMARY OPTION MENU

<table>
<thead>
<tr>
<th>OPTION</th>
<th>MainView AutoOPERATOR</th>
<th>DATE</th>
<th>-- YYYY/MM/DD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>----------------------------</td>
<td></td>
<td>---------------</td>
</tr>
<tr>
<td></td>
<td>Operator Workstations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 ALERTS</td>
<td>ALERT Management</td>
<td>DATE</td>
<td>-- YYYY/MM/DD</td>
</tr>
<tr>
<td>2 MVS</td>
<td>OS/390 or z/OS Resources</td>
<td>TIME</td>
<td>-- 15:28:58</td>
</tr>
<tr>
<td>3 CICS</td>
<td>CICS Resources</td>
<td>USERID</td>
<td>-- BAOMXY1</td>
</tr>
<tr>
<td>4 IMS</td>
<td>IMS Resources</td>
<td>MODE</td>
<td>-- ISPF 6.0</td>
</tr>
<tr>
<td>5 NETVIEW</td>
<td>NetView Resources</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 TAPESHARE</td>
<td>Tape Drive Management</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 QO</td>
<td>IBM MQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Administration</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 AUTOMATION</td>
<td>Basic and Advanced Automation</td>
<td>DATE</td>
<td>-- YYYY/MM/DD</td>
</tr>
<tr>
<td>9 PARMS</td>
<td>Dynamic Parameter Manager</td>
<td>TIME</td>
<td>-- 15:28:58</td>
</tr>
<tr>
<td>General Services</td>
<td></td>
<td>USERID</td>
<td>-- BAOMXY1</td>
</tr>
<tr>
<td>C CYCLE</td>
<td>Service Refresh Cycle Setup</td>
<td>MODE</td>
<td>-- ISPF 6.0</td>
</tr>
<tr>
<td>J JOURNAL</td>
<td>Journal Log</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G LOGJRNL</td>
<td>Journal Log (Windows Mode)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M MESSAGES</td>
<td>Messages and Codes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P PERFORMANCE</td>
<td>Performance and History (Windows Mode)</td>
<td>DATE</td>
<td>-- YYYY/MM/DD</td>
</tr>
<tr>
<td></td>
<td>----------------------------</td>
<td></td>
<td>---------------</td>
</tr>
</tbody>
</table>
When you select Option 9, the Dynamic Parameter Manager panel is shown (Figure 8 on page 71).

**Figure 8: Dynamic Parameter Manager—example 1**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Dynamic Parameter Manager</th>
<th>AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND</td>
<td>Add</td>
<td>TGT</td>
</tr>
<tr>
<td>Primary command: Add</td>
<td></td>
<td>BBSYSA</td>
</tr>
<tr>
<td>LC CMDS</td>
<td>(S)elect, (B)rowse w/o Symbolic Substitution</td>
<td>DATE --- YY/MM/DD</td>
</tr>
<tr>
<td>Actv CMDS</td>
<td>(A)ctivate, B(r)owse w/ Symbolic Substitution</td>
<td>TIME --- 12:45:48</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cmd</th>
<th>Member</th>
<th>Actv Lib VV.MM</th>
<th>Created</th>
<th>Changed</th>
<th>Size</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>_</td>
<td>AAOALSDL</td>
<td>1 01.17</td>
<td>04/09/21</td>
<td>08/07/01</td>
<td>07:02</td>
<td>7</td>
</tr>
<tr>
<td>_</td>
<td>AADALSD00</td>
<td>YES 01.76</td>
<td>93/06/08</td>
<td>08/06/04</td>
<td>10:38</td>
<td>22</td>
</tr>
<tr>
<td>_</td>
<td>AAOALSD98</td>
<td>00.03</td>
<td>08/05/21</td>
<td>08/05/22</td>
<td>08:59</td>
<td>6</td>
</tr>
<tr>
<td>_</td>
<td>AAOARP01</td>
<td>01.01</td>
<td>04/09/21</td>
<td>08/03/18</td>
<td>10:23</td>
<td>31</td>
</tr>
<tr>
<td>_</td>
<td>AAOARP00</td>
<td>YES 01.55</td>
<td>03/03/26</td>
<td>07/12/03</td>
<td>10:35</td>
<td>76</td>
</tr>
<tr>
<td>_</td>
<td>AADARPD2</td>
<td>01.05</td>
<td>06/01/20</td>
<td>07/12/05</td>
<td>06:58</td>
<td>36</td>
</tr>
<tr>
<td>_</td>
<td>AAOEXP01</td>
<td>01.02</td>
<td>08/05/20</td>
<td>08/05/21</td>
<td>13:48</td>
<td>38</td>
</tr>
<tr>
<td>_</td>
<td>AAOEXP00</td>
<td>YES 03.08</td>
<td>87/05/12</td>
<td>07/12/05</td>
<td>07:14</td>
<td>35</td>
</tr>
<tr>
<td>_</td>
<td>AAOEXP098</td>
<td>00.01</td>
<td>08/05/21</td>
<td>08/05/21</td>
<td>12:24</td>
<td>26</td>
</tr>
<tr>
<td>_</td>
<td>AAOPRM01</td>
<td>3 01.00</td>
<td>04/09/01</td>
<td>04/09/01</td>
<td>06:52</td>
<td>15</td>
</tr>
<tr>
<td>_</td>
<td>AAOPRMSV</td>
<td>1 01.07</td>
<td>07/05/14</td>
<td>08/05/22</td>
<td>06:52</td>
<td>63</td>
</tr>
<tr>
<td>_</td>
<td>AAOPRM02</td>
<td>YES 02.95</td>
<td>93/03/29</td>
<td>08/08/06</td>
<td>06:17</td>
<td>65</td>
</tr>
<tr>
<td>_</td>
<td>AAOPRM03</td>
<td>1 01.11</td>
<td>06/01/20</td>
<td>07/12/11</td>
<td>06:23</td>
<td>55</td>
</tr>
<tr>
<td>_</td>
<td>AAOPRM04</td>
<td>1 01.05</td>
<td>07/05/21</td>
<td>08/04/01</td>
<td>06:47</td>
<td>64</td>
</tr>
</tbody>
</table>

Use this panel to see a list of all the AAOALS xx, AAOARPxx, AAOEXPxx, and AAOTSPxx members in BBPARM where the

- AAOALSxx members contain parameters that determine how much extended storage MainView AutoOPERATOR ALERTs use on your system
- AAOARPxx members contain parameters that control how the Automation Reporter application operates
- AAOEXPxx members contain parameters and thresholds that determine how efficiently MainView AutoOPERATOR EXECs execute on your system
- AAOPRMxx members contain parameters that determine how certain features of MainView AutoOPERATOR operate
- The AAOTSPxx members contain parameters that control how TapeSHARE operates

From this panel, you can

- use the line command Select in the **Cmd** column to select a member to view and modify the parameters, if you choose
  
  Refer to “Describing the line commands” on page 72 for more information about the line commands.

For more information about selecting
- an AAOALSxx member, refer to “Modifying ALERT thresholds in AAOALSxx” on page 74.

- selecting an AAOARPxx member, refer to “Modifying Automation Reporter application parameters in AAOARPxx” on page 79.

- an AAOEXPxx member, refer to “Modifying EXEC parameters in AAOEXPxx” on page 82.

- an AAOPRMxx member, refer to “Modifying MainView AutoOPERATOR parameters in AAOPRMxx” on page 94.

- an AAOTSPxx member, refer to “Modifying TapeSHARE parameters in AAOTSPxx” on page 109.

- use the primary command ADD to add a new AAOALSxx, AAOARPxx, AAOEXPxx, AAOPRMxx, or AAOTSPxx member. To add new members, type one of the following commands:

  ADD AAOALSxx
  ADD AAOARPxx
  ADD AAOEXPxx
  ADD AAOPRMxx
  ADD AAOTSPxx

  on the COMMAND line, where xx is a new suffix for a new member. For example, to add a new member named AAOEXP02, type

  ADD AAOEXP02

**Describing the line commands**

The following table describes the line commands.

<table>
<thead>
<tr>
<th>Line command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S)elect</td>
<td>selects a member to view and modify the parameters</td>
</tr>
</tbody>
</table>
Line command | Description
--- | ---
(A)ctivate | use the line command Activate with the member to either
- | dynamically change the BBPARM member that your system is currently using
- | implement any changes you might have made to a member
The Activate command displays the Confirm Parameter Activation panel. This function allows you to verify that you want to dynamically change an AAOALSxx, AAOARPxx, AAOEXPxx, AAOPRMxx or AAOTSPxx member. For more information about activating your changes, refer to “Dynamically implementing new settings with the Activate command” on page 115.

(B)rowse without symbolic substitution | allows you to browse the member without viewing the symbolic substitution

B(r)owse with symbolic substitution | allows you to browse the member with any symbolic parameters substituted with the current value displayed

**Describing the fields**

The fields for the Dynamic Parameter Manager panel are as follows:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cmd</td>
<td>enables you to enter the Select or Activating line command</td>
</tr>
<tr>
<td>Member</td>
<td>is the list of members that can be</td>
</tr>
<tr>
<td></td>
<td>- AAOALSxx for ALERTs thresholds</td>
</tr>
<tr>
<td></td>
<td>- AAOARPxx for Automation Reporter operation</td>
</tr>
<tr>
<td></td>
<td>- AAOEXPxx for EXEC thresholds</td>
</tr>
<tr>
<td></td>
<td>- AAOPRMxx for certain MainView AutoOPERATOR features</td>
</tr>
<tr>
<td></td>
<td>- AAOTSPxx for TapeSHARE operation</td>
</tr>
<tr>
<td>Actv</td>
<td>indicates which members are currently active in the target BBI-SS PAS</td>
</tr>
<tr>
<td></td>
<td>YES indicates that the parameter is active for the BBI-SS PAS.</td>
</tr>
<tr>
<td>Lib</td>
<td>is the library volume number</td>
</tr>
<tr>
<td>VV.MM</td>
<td>indicates the version level (VV) and modification level (MM)</td>
</tr>
<tr>
<td>Created</td>
<td>is the date the member was created</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>Changed</td>
<td>is the date and time the member was last edited</td>
</tr>
<tr>
<td>Size</td>
<td>is the size (in number of lines) of the member</td>
</tr>
<tr>
<td>ID</td>
<td>is the user ID of the person who last changed the member or, if the member was last changed with the Dynamic Parameter Managers, the job name of the subsystem used</td>
</tr>
</tbody>
</table>

### Filtering the display of data

You can qualify (or filter) the members shown on the panel by entering a prefix or string of characters in the input field located under the column heading Member. For example, to see all the member names of AAOEXPxx, enter the characters AAOEXP in the input field:

<table>
<thead>
<tr>
<th>Cmd</th>
<th>Member</th>
<th>Lib VV.MM</th>
<th>Created</th>
<th>Changed</th>
<th>Size</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AAOEXP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The output shows only those members with the prefix AAOEXP.

### Modifying ALERT thresholds in AAOALSxx

BBPARM member AAOALSxx contain the parameters that determine how much storage MainView AutoOPERATOR ALERTs can use on your system.

To modify an AAOALSxx member, type **S** in the *Cmd* field of the Dynamic Parameter Manager panel. Figure 9 on page 74 shows an example of the panel.

**Figure 9: Dynamic Parameter Manager panel—example 2**

```
BMC Software ---------------- Dynamic Parameter Manager ----------------- AutoOPERATOR
COMMAND ===>                                TGT ===> BBSYSA
Primary command: Add                       Date --- YY/MM/DD
LC CMDS --- (S)elect, \(B\)rowse w/o Symbolic Substitution
(\(A\)ctivate, \(B\)(r)owse w/ Symbolic Substitution
Cmd Member Lib VV.MM Created Changed Size ID
s AAOALS00 1 01.11 YY/MM/DD YY/MM/DD 20:12 4 JDB1
- AAOALS01 1 00.01 YY/MM/DD YY/MM/DD 13:30 3 JDB1
- AAOARP01 1 00.01 YY/MM/DD YY/MM/DD 13:30 3 JDB1
- AAOEXP00 1 02.26 YY/MM/DD YY/MM/DD 20:12 27 JDB1
- AAOPRM00 1 02.26 YY/MM/DD YY/MM/DD 20:12 27 JDB1
- AAOTSP00 1 02.01 YY/MM/DD YY/MM/DD 20:12 12 JDB1
END OF MEMBERS****************************
```

Use this panel to see a list of all the AAOALSxx, AAOARPxx, AAOEXPxx, AAOPRMxx, and AAOTSPxx members in BBPARM.
From this panel, you can

- use the line commands; refer to “Describing the line commands” on page 72 for more information
- use the primary command ADD to add a new members

To add a new member or members, type the following command on the COMMAND line, where xx is a new suffix for a new member:

```
ADD AAOALSxx
```

### Setting ALERTs storage thresholds

Use the Alerts Storage Thresholds panel to see and modify thresholds that control how much storage ALERTs can use on your system.

Figure 10 on page 75 shows an example of this panel.

**Figure 10: ALERTs Storage Thresholds panel**

This panel provides two columns of data:

- on the left, the value of the parameters in the AAOALS xx member for the target BBI-SS PAS
  You can enter your modifications on this side of the panel.

- on the right, the value of the parameters that are currently in effect for the target BBI-SS PAS; these values cannot be edited

### Describing the fields

Descriptions of the fields on this panel follow:
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXSTOR</td>
<td>maximum amount of BBI-SS PAS extended private virtual storage to be allocated to ALERTs. Specify the storage amount in kilobytes and thousandths of megabytes. The default is zero, which means no checking is done.</td>
</tr>
<tr>
<td>WARNLVL1</td>
<td>first warning threshold level for queued ALERTs. The default is 60%, which means that if MAXSTOR=1K, a warning message is issued when the amount of virtual storage allocated to queued ALERTs reaches .6K. The following is an example of a warning message: AU6100W ALERT STORAGE USAGE 60% OVER 1ST LEVEL (60%)</td>
</tr>
<tr>
<td>WARNLVL2</td>
<td>second warning threshold level for queued ALERTs. When alert storage usage is divided by the value that is specified with the MAXSTOR parameter, warning message AU6100W is issued when the result is greater than or equal to the percent that is specified by WARNLVL2. If you specify 0 or a percentage lower than WARNLVL1, this warning level is ignored. The default is 75%, which means that if MAXSTOR=1K, a warning message is issued when the amount of virtual storage allocated to queued ALERTs reaches .75K. The message looks like the following message: AU6100W ALERT STORAGE USAGE 75% OVER 2ND LEVEL (75%)</td>
</tr>
</tbody>
</table>

When the maximum amount of storage is used (as set by MAXSTOR), two additional messages are produced. The first message is EM0022E ERROR PROCESSING .. DEMOAL22 .. ALERT NOT ADDED QUEUE IS FULL where DEMOAL22 is the name of the ALERT that was not added to the ALERT queue because the maximum storage limit has been reached. The second message is EM0020E - IMFEXEC ALERT DASD5.11:57:28 'RESERVE IN PROGRESS...' where the text of the message is the text of the IMFEXEC ALERT command that attempted to store the ALERT. To solve this problem, use a Rule to interrogate the information fields in the message and delete less important ALERTs. No more ALERTs are queued until the amount of storage used by queued ALERTs drops below MAXSTOR. When the shortage condition is alleviated, the following message is issued: AU6200I ALERT STORAGE USAGE 58%, NO LONGER OVER THRESHOLD. |
ALERTNV

- default setting for the RETAIN keyword for MainView AutoOPERATOR ALERTs
- ALERTs that are retained are written and saved to disk and therefore, exist across BBI-SS PAS restarts and system z/OS IPLs.
- When you are creating ALERTs (using the IMFEXEC ALERT statement, the Rules Processor ALERT panels, the AOEXEC ALERT statement, and so on) and you do not explicitly specify a setting for the RETAIN keyword, the ALERTNV parameter allows you to specify whether the default setting should be RETAIN(NO) or RETAIN(YES).
- For example, for ALERTs that do not specify the RETAIN keyword:
  - when the ALERTNV parameter is not specified or is specified as NO, RETAIN(NO) is used as the default.
  - when ALERTNV=YES is specified, RETAIN(YES) is used as the default.
- Regardless of which setting you specify for the ALERTNV parameter, you can override the setting when you create a specific ALERT, where the RETAIN=YES or RETAIN=NO keyword is explicitly set. When you override this setting, only that ALERT is retained across BBI-SS PAS restarts and z/OS IPLs.
  - **Note:** When an ALERT is targeted to a different BBI-SS PAS, by default, the ALERT retains the ALERTNV setting of the target BBI-SS PAS.
- Changing the ALERTNV setting takes effect only after a BBI-SS PAS warm or cold start.
- The default is NO.
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PUBLISH</strong></td>
<td><strong>whether MainView AutoOPERATOR should (by default) publish events and ALERTs to BMC Impact Manager cells that are configured to receive input from MainView AutoOPERATOR through the General Message Exchange (GME).</strong>&lt;br&gt;Settings for GME are defined in BBPARM member AAOGME00. For more information about how to configure BBPARM member AAOGME00 to send events and ALERTs to BMC Impact Manager cells, refer to “Implementing BMC Event Manager to work with BMC products” on page 209.&lt;br&gt;Valid values are as follows:&lt;br&gt;■ <strong>ADD</strong>: publish, but do not replace a previously published ALERT; does not apply to ADAPTOR connections&lt;br&gt;■ <strong>REPLACE</strong>: publish, but delete a previously published ALERT before adding the new one; does not apply to ADAPTOR connections&lt;br&gt;■ <strong>NO</strong>: do not publish ALERTs&lt;br&gt;The default is ADD.</td>
</tr>
<tr>
<td><strong>AUDIT</strong></td>
<td><strong>specifies options for auditing the life-cycle of an ALERT</strong>&lt;br&gt;The default is NONE. Valid values are as follows:&lt;br&gt;■ <strong>ALL</strong>: all stages of an ALERT’s life cycle will be audited&lt;br&gt;■ <strong>NONE</strong>: no auditing will be performed&lt;br&gt;■ <strong>ADD</strong>: generate an audit message when an ALERT is created with the ADD keyword&lt;br&gt;■ <strong>ESCAL</strong>: generate an audit message when an ALERT’s priority escalates&lt;br&gt;■ <strong>DELETE</strong>: generate an audit message when an ALERT is deleted with the DELETE keyword&lt;br&gt;Audit messages are generated into the BBI Journal. Enter a &quot;?&quot; next to the keyword to display a panel of the current options that you can edit.</td>
</tr>
<tr>
<td><strong>ALERTLOG</strong></td>
<td><strong>specifies whether MainView AutoOPERATOR logs all ALERT activity to the MainView Logger</strong>&lt;br&gt;When ALERTLOG is set to YES, the ALRTLOG view is available in windows-mode for viewing the ALERT log.</td>
</tr>
</tbody>
</table>
Modifying Automation Reporter application parameters in AAOARP

BBPARM members AAOARP contain the parameters that activate and control the data collection and offloading functions of the Automation Reporter application.

To modify an AAOARP member, type `S` in the `Cmd` field of the Dynamic Parameter Manager panel. Figure 11 on page 79 shows an example of the panel.

**Figure 11: Dynamic Parameter Manager panel—example 3**

```
BMC Software ------------- Dynamic Parameter Manager ------------- AutoOPERATOR
COMMAND ===>                                                 TGT ===> BBSYSA
Primary command: Add                                              Date --- YY/MM/DD
Time --- 11:41:36                                                   
LC CMDs --- (S)elect, (B)rowse w/o Symbolic Substitution
            (A)ctivate, B(r)owse w/ Symbolic Substitution

<table>
<thead>
<tr>
<th>Cmd</th>
<th>Member</th>
<th>Lib</th>
<th>VV.MM</th>
<th>Created</th>
<th>Changed</th>
<th>Size</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>_</td>
<td>AAOALS00</td>
<td>1</td>
<td>01.11</td>
<td>YY/MM/DD</td>
<td>YY/MM/DD</td>
<td>20:12</td>
<td>JDB1</td>
</tr>
<tr>
<td>_</td>
<td>AAOALS01</td>
<td>1</td>
<td>00.01</td>
<td>YY/MM/DD</td>
<td>YY/MM/DD</td>
<td>13:30</td>
<td>JDB1</td>
</tr>
<tr>
<td>S</td>
<td>AAOARP01</td>
<td>1</td>
<td>00.01</td>
<td>YY/MM/DD</td>
<td>YY/MM/DD</td>
<td>13:30</td>
<td>JDB1</td>
</tr>
<tr>
<td>_</td>
<td>AAOEXP00</td>
<td>1</td>
<td>02.26</td>
<td>YY/MM/DD</td>
<td>YY/MM/DD</td>
<td>20:12</td>
<td>JDB1</td>
</tr>
<tr>
<td>_</td>
<td>AAOPRM00</td>
<td>1</td>
<td>02.26</td>
<td>YY/MM/DD</td>
<td>YY/MM/DD</td>
<td>13:30</td>
<td>JDB1</td>
</tr>
<tr>
<td>_</td>
<td>AAOTSP00</td>
<td>1</td>
<td>02.01</td>
<td>YY/MM/DD</td>
<td>YY/MM/DD</td>
<td>20:12</td>
<td>JDB1</td>
</tr>
</tbody>
</table>
```

Use this panel to see a list of all the AAOALSxx, AAOARPxx, AAOEXPxx, AAOPRMxx, and AAOTSPxx members in BBPARM.

From this panel, you can

- use the line commands; refer to Table 9 on page 72 for more information
- use the primary command ADD to add a new members

To add a new member or members, type the following command on the COMMAND line, where `xx` is a new suffix for a new member:

```
ADD AAOARPxx
```

**Setting Automation Reporter parameters**

Use the Automation Reporter Parms panel to

- see and modify parameters for the Automation Reporter application
- control how often data is collected
- name a data set to hold the data
- indicate when the collected data is offloaded to the data set
Figure 12 on page 80 shows an example of this panel.

Figure 12: Automation Reporter Parms panel

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Automation Reporter</th>
<th>Parms</th>
<th>AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND :==&gt;</td>
<td></td>
<td></td>
<td>TGT --- BBSYA</td>
</tr>
<tr>
<td>DATE --- YY/MM/DD</td>
<td></td>
<td></td>
<td>TIME --- 13:39:59</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Member</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>New DSN :==&gt; AAO66..&amp;SYSPLEX..&amp;SYSNAME..OFFLOAD</td>
<td></td>
</tr>
</tbody>
</table>

| Disp | SHR | Current ===>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Interval :==&gt; 1439</td>
<td>0</td>
<td>Interval minutes (10 - 1440, or 0)</td>
</tr>
<tr>
<td>Off int :==&gt; 1439</td>
<td>1440</td>
<td>Offload interval minutes (60-1440)</td>
</tr>
<tr>
<td>Alert Queue :==&gt; ARMAIN2</td>
<td>ARALERT</td>
<td>Alert Queue</td>
</tr>
<tr>
<td>IR History :==&gt; NO</td>
<td>YES</td>
<td>Interval Recorder History</td>
</tr>
<tr>
<td>AR Options :==&gt; (Enter ? for AR Options list)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Press END to save changes, CANCEL to cancel changes

This panel provides two columns of data:

- on the left, the value of the parameters in the AAOARP xx member for the target BBI-SS PAS.
  You can enter your modifications on this side of the panel.

- on the right, the value of the parameters that are currently in effect for the target BBI-SS PAS; these values cannot be edited.
  The first time you bring up the Automation Reporter, no current data set values are available. Therefore, the Current ===>
  field and all the values on the right side of the panel are blank.
  After you have used the Automation Reporter once, the next time you access this panel, the Current ===>
  field and all the display fields show the values you previously entered.
  The left side of the panel displays the default values in BBPARM member AAOARPxx, as shown in this panel.

Describing the fields

Descriptions of the fields on this panel follow:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New DSN</td>
<td>specify the name of the sequential data set that you have preallocated. The user-defined data set should be a fixed-block, sequential data set with a maximum record length of 580 bytes. This data set must be preallocated before the Automation Reporter can be activated.</td>
</tr>
<tr>
<td>Field name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| Disp       | specify the disposition of the user-defined sequential data set  
Valid disposition values are:  
- SHR (share): causes the Automation Reporter to overwrite existing data  
- MOD (modify): causes the Automation Reporter to append data to the end of the data set from each offload cycle  
  MOD is the default value. |
| Interval   | specify the collection interval (in minutes) for the Automation Reporter to collect data  
For example, if you specify 60, the Automation Reporter records the data in the subsystem for the resources and activities every 60 minutes.  
Specifying any value greater than zero means that the Automation Reporter will collect data.  
Valid collection intervals range from 10 minutes to 1440 minutes. The default is 10 minutes. |
| Off Int    | specify the offload interval (in minutes) for the Automation Reporter to offload the data it has collected to the data set  
For example, if you specify 120, the Automation Reporter writes the data it has collected to the sequential data set every 2 hours.  
Valid offloading intervals are 60 minutes to 1440 minutes. The default is 1440 (or once a day). |
| Alert Queue| specify the name of an ALERT queue to which any ALERTs created by the Automation Reporter is queued  
For example, if you specify ARALRT, ALERTs are queued to a queue named ARALRT.  
Any valid ALERT queue name can be entered in this field (default queue name is MAIN). You do not have to enter the name of a pre-existing ALERT queue. You can enter the name of a new queue and it is created for you when the Automation Reporter is activated and ALERTs are generated. |
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRHIST</td>
<td>specifies whether the windows mode Automation Reporter will collect interval data that allows you to perform short term and long term analysis of automation processes and performance. This parameter also enables the windows mode Automation Reporter views that show real-time data collected by interval and allows for local or SSI summarization of the automation information. For more information about the windows mode version of the Automation Reporter, refer to the MainView AutoOPERATOR Basic Automation Guide, Volume 2. <strong>Note:</strong> BMC recommends that you migrate your automation data collection processes to use the windows-mode Automation Reporter applications available with MainView AutoOPERATOR version 7.1 and later instead of using the Automation Reporter that is available with MainView AutoOPERATOR version 6.5.xx and earlier. In a future release, BMC Software plans to discontinue supporting the MainView AutoOPERATOR version 6.5.xx (and earlier) version of the Automation Reporter application. The default is NO; valid values are YES or NO.</td>
</tr>
<tr>
<td>AR Options</td>
<td>specifies one or more Automation Reporter collectors that will actively collect data. Default value is ALL; valid values are ALL (all collectors will be active), ACT, ALT, AUS, CPU, EXE, EXC, EVT, HTH, RES, RST, RUL, TOM, TSL, TSP, ELU, and RLU. Enter a question mark (?) next to this keyword to display a list of the current values. ELU includes EXE data collection and RLU includes RUL data collection. If you do not specify ELU, no EXE data collection is performed unless you specify the EXE value. If you do not specify ELU, no RUL data collection is performed, unless you specify the RUL value.</td>
</tr>
</tbody>
</table>

---

### Modifying EXEC parameters in AAOEXP.xx

BBPARM member AAOEXP.xx contains parameters that determine the resource throttle settings and the environment for MainView AutoOPERATOR EXECS.

To modify an AAOEXP.xx member, type **S** in the **Cmd** field next to a member on the Dynamic Parameter Manager panel (refer to **Figure 8 on page 71**).
Figure 13 on page 83 shows an example of the panel that is displayed when you select an AAOEXPxx member called AAOEXP00.

**Figure 13: Dynamic Parameter Manager Member panel**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Dynamic Parameter Manager Member</th>
<th>AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND</td>
<td>Dynamic Parameter Manager Member</td>
<td>AutoOPERATOR</td>
</tr>
<tr>
<td>TGT</td>
<td>BBSYSA</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>YY/MM/DD</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>11:41:44</td>
<td></td>
</tr>
</tbody>
</table>

Member Name: AAOEXP00 - AutoOPERATOR EXEC Specifications

<table>
<thead>
<tr>
<th>LC CMDS --- (S)elect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cmd</td>
</tr>
<tr>
<td>Panel Description</td>
</tr>
<tr>
<td>_ EXEC Thresholds</td>
</tr>
<tr>
<td>_ EXEC Parameters</td>
</tr>
<tr>
<td>_ EXEC Parameters 2</td>
</tr>
</tbody>
</table>

Use this panel to select additional panels that allow you to see and modify either:

- **EXEC Thresholds** — set limits for how all your EXECs execute on your system; for example:
  - the maximum number of EXECs that can run in the Normal or High priority queue
  - the maximum number of EXECs that are allowed to abend before the EXEC Manager is terminated
  
  For more information about selecting EXEC thresholds, refer to “Setting EXEC Thresholds” on page 83.

- **EXEC Parameters** — specify parameters for your EXECs themselves; for example:
  - whether EXEC parameters are translated to uppercase
  - what the maximum number of OSPI ACBs are

  For more information about selecting EXEC parameters, refer to “Setting EXEC Parameters” on page 88.

- **EXEC Parameters 2** — specify whether MainView AutoOPERATOR sends messages from EXECs to the EMLOG or enable the IMFEXEC HMC command to work in EXECs
  
  See “Additional EXEC parameters” on page 92 for more information.
  
  See the *MainView AutoOPERATOR Advanced Automation Guide* for more information about the IMFEXEC HMC command.

---

**Setting EXEC Thresholds**

Use the EXEC Thresholds panel to view or modify values that determine how many EXECs can run in your system in the Normal or high-priority queues, how many EXEC abends that you want to be able to support, and so on.
Figure 14 on page 84 shows an example of the EXEC Thresholds panel.

**Figure 14: EXEC Thresholds panel**

<table>
<thead>
<tr>
<th>Member</th>
<th>Current</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXABCNT</td>
<td>3</td>
<td># of ABENDS allowed before EXEC disabled</td>
</tr>
<tr>
<td>MAXNORM</td>
<td>5</td>
<td>Maximum concurrent Normal EXECs</td>
</tr>
<tr>
<td>MAXHIGH</td>
<td>3</td>
<td>Maximum concurrent High Priority EXECs</td>
</tr>
<tr>
<td>MAXNORMQ</td>
<td>5</td>
<td>Maximum Normal EXECs Queued</td>
</tr>
<tr>
<td>MAXHIGHQ</td>
<td>3</td>
<td>Maximum High Priority EXECs Queued</td>
</tr>
<tr>
<td>MAXHOTQ</td>
<td>10</td>
<td>Maximum Hot/First priority EXECs Queued</td>
</tr>
<tr>
<td>WARN_LVL1</td>
<td>65</td>
<td>Initial threshold (percent)</td>
</tr>
<tr>
<td>WARN_LVL2</td>
<td>80</td>
<td>Secondary threshold (percent)</td>
</tr>
<tr>
<td>SELLM</td>
<td>10</td>
<td>Maximum depth for SELECT with WAIT(YES)</td>
</tr>
<tr>
<td>MAXTPUT</td>
<td>0</td>
<td>Maximum TPUT allowed in each EXEC</td>
</tr>
</tbody>
</table>

Press END to save changes, CANCEL to cancel changes.

This panel provides two columns of data:

- on the left, the value of the parameters in the AAOEXP.xx member for the target BBI-SS PAS

  You can enter your modifications on this side of the panel.

- on the right, the value of the parameters that are currently in effect for the target BBI-SS PAS; these values cannot be edited

**Scheduling EXECs**

With the EXEC Thresholds panel, you can specify the number of EXECs that can be scheduled to the High and Normal priority queues by using MAXNORM and MAXHIGH parameters. When the number of scheduled EXECs matches the values set in these parameters, all scheduling stops until the number of EXECs drops below the thresholds.

You can also specify two thresholds for the High and Normal queues that, when reached, issue warning messages telling you that the threshold has been reached. These thresholds are WARN_LVL1 and WARN_LVL2 and the values represent percentages of MAXNORMQ and MAXHIGHQ.

**Examples**

For example, you can specify

MAXNORMQ ===> 150

MAXHIGHQ ===> 100

and
**WARNLVL1 ===> 50**

**WARNLVL2 ===> 75**

For the Normal queue, warning messages are sent when 75 EXECs are scheduled and again when 112 EXECs are scheduled.

\[\text{WARNLVL1} \times 50\% \text{ of 150} = 75\]

\[\text{WARNLVL2} \times 75\% \text{ of 150} = 112\]

For the High queue, warning messages are sent when 50 EXECs are scheduled and again when 75 EXECs are scheduled.

\[\text{WARNLVL1} \times 50\% \text{ of 100} = 50\]

\[\text{WARNLVL2} \times 75\% \text{ of 100} = 75\]

Scheduling of EXECs stops when the maximum values are reached. EXEC scheduling automatically resumes when the number of EXECs drops below the maximum. Refer to the descriptions of the MAXNORMQ, MAXHIGHQ, WARNLVL1, and WARNLVLV2 fields in “Describing the fields” on page 85 for more information.

**Describing the fields**

A description of the fields on this panel follows.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member</td>
<td>specifies the values for the new AAOEXPxx member</td>
</tr>
</tbody>
</table>
| Current    | specifies the values in the AAOEXPxx member that the BBI-SS PAS is currently using  
These values are stored in CSA.  
The member that the BBI-SS PAS is currently using is one of the following choices:  
- the AAOEXPxx member that was used during the most recent BBI-SS PAS initialization  
- the most recently applied AAOEXPxx member |
| EXABCNT    | is the maximum number of abends that a single EXEC can sustain before it is disabled  
Valid values range from 1 to 99 and the default is 5. For example, if you specify EXABCNT=10 and an EXEC abends 10 times, the EXEC is disabled after the tenth abend. |
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXNORM</td>
<td>is the maximum number of EXECs that can execute concurrently on Normal priority threads. Valid values range from 1 to 99 and the default is 1.</td>
</tr>
<tr>
<td>MAXHIGH</td>
<td>is the maximum number of EXECs that can execute concurrently on high-priority threads. Valid values range from 1 to 99 and the default is 5.</td>
</tr>
</tbody>
</table>
| MAXNORMQ   | is the maximum number of Normal priority EXECs that may be queued for execution. Valid values range from 0 to 999 and the default is 0. A value of 0 means that no checking is done. When the maximum number is reached, no more EXECs are scheduled to the Normal priority queue and a warning message is issued: | \[
\text{EM6100W \ MAXNORMQ EXEC 19/20, OVER LAST LEVEL (95%)}
\]
EXEC scheduling resumes when the queue size drops below the maximum. An informational message is issued to inform you that scheduling has resumed: | \[
\text{EM6200I \ MAXNORMQ EXEC 9/20, NO LONGER OVER THRESHOLD}
\]  
**Note:** EXECs scheduled with WAIT=YES are never queued. They are executed on the thread of the EXEC that selected it. |
| MAXHIGHQ   | is the maximum number of High priority EXECs that may be queued for execution. Valid values range from 0 to 999 and the default is 0. A value of 0 means that no checking is done. When the maximum number is reached, no more EXECs are scheduled to the high priority queue and a warning message is issued: | \[
\text{EM6100W \ MAXHIGHQ EXEC 19/20, OVER LAST LEVEL (95%)}
\]
EXEC scheduling resumes when the queue size drops below the maximum. An informational message is issued to inform you that scheduling has resumed: | \[
\text{EM6200I \ MAXHIGHQ EXEC 9/20, NO LONGER OVER THRESHOLD}
\]  
**Note:** EXECs scheduled with the IMFEXEC command WAIT=YES are never queued. They are executed on the thread of the EXEC that selected it. |
| MAXHOTQ    | is the maximum number of First or Hot priority EXECs that can be queued. Valid values range from 0 to 999 and the default is 10. A value of 0 means that no checking is done and all Hot or First priority EXECs are queued. When the maximum is reached, no more EXECs with First or Hot priority specified are scheduled until the queue level drops below the maximum level set and a warning message is issued: | \[
\text{EM6100W \ MAXHOTQ EXEC 19/20, OVER LAST LEVEL (95%)}
\]
EXEC scheduling resumes when the queue size drops below the maximum. An informational message is issued to inform you that scheduling has resumed: | \[
\text{EM6200I \ MAXHOTQ EXEC 9/20, NO LONGER OVER THRESHOLD}
\]
### Field name | Description
--- | ---
WARNLVL1 | specifies a preliminary percentage of EXECs in a queue that, when met, generates a warning message.
Valid values range from 0% to 99% and the default is 60%.
Use this threshold to set the first percentage that triggers a warning message to be sent. For example, if you set MAXHIGHQ=100, and WARNLVL1=60, a warning message is issued when 60 EXECs are queued to the High-priority queue. The message looks like the following example:

```
EM6100W MAXHIGHQ EXEC 60/ 100, OVER 1ST LEVEL (60%)
```
This threshold is applicable to the High and Normal priority queues.

WARNLVL2 | specifies a secondary percentage of EXECs in a queue that, when met, generates a warning message.
Valid values range from 0% to 99% and the default is 75%.
Use this threshold to set a second percentage that triggers a warning message to be sent.
For example, if you set MAXHIGHQ=100 and WARNLVL2=75, a warning message is issued when 75 EXECs are queued to the high-priority queue. The message looks like the following example:

```
EM6100W MAXHIGHQ EXEC 75/ 100, OVER 2ND LEVEL (75%)
```
In addition, a second message is issued that identifies the first EXEC that was not scheduled because of this queuing condition:

```
EM6150W EXEC-COLORS NOT QUEUED, OVER LAST LEVEL THRESHOLD
```
In this example, Colors is the name of an EXEC that was not scheduled. Additionally, no more EXECs are scheduled to the high-priority queue until the usage percentage of the queue is less than the value set for MAXHIGHQ. This threshold is applicable to the High and Normal priority queues.

SELLIM | is the maximum number of active EXECs on a single thread.
Valid values range from 0 to 99 and the default is 0.
This parameter is designed to control recursive calls (where EXECA schedules EXECB, and EXECB schedules EXECC, and so on) that run out of control, and can potentially fill up the private storage in the BBI-SS PAS and cause automation to stop.
An example of recursive calls is EXECA schedules EXECB with the WAIT=YES parameter and EXECB in turn, calls EXECA.
### Field name | Description
--- | ---
MAXTPUT | is the maximum number of TSO TPUTs that is allowed for each execution of an EXEC

Valid values range from 0 to 999 and the default is 0. TPUTs for EXECs scheduled with the WAIT=YES parameter are counted separately from the scheduling EXEC. If the maximum is exceeded, the following message is issued:

```
EM0026W TPUT JOURNALING SUSPENDED, BACKLOG HAS EXCEEDED MAXIMUM
```

and subsequent TPUTs are ignored until the EXEC terminates.

---

### Setting EXEC Parameters

Use the EXEC Parameters panel to view or modify values that determine how individual EXECs execute in your system.

**Figure 15 on page 88** shows an example of the panel.

**Figure 15: EXEC Parameters panel**

```
BMC Software ---------------- EXEC Parameters ----------------- AutoOPERATOR
COMMAND ===>                                                 TGT  --- BBSYSA
DATE --- YY/MM/DD
TIME --- 09:30:55
Member         Current
TIMEXLIM ===>        1             0  Maximum CPU time limit for an EXEC
PEREXLIM ===>        1             0  Maximum CPU percentage usage for an EXEC
TSOTIME ===>        1          1000  Timeout value for attached TSO commands
ELAPSLIM ===>        0             0  Maximum elapsed time limit for an EXEC
UCPARMS ===> YES           NO        Translate EXEC parms to UPPERCASE
AUDITMSG ===> YES           YES       Write JRNL msg for BLDL and submit EXEC
AUDITEXE ===> NO            NO        Write JRNL msg for EXEC Start and End
OSPINUM ===>        0             2  Maximum # of OSPI ACBs
OSPIPRFX ===> OSPI          J062      4 character OSPI ACB prefix
UNITNAME ===> SYSALLDA      SYSALLDA  UNITNAME used for allocating a dataset
PREFIX   ===>               J071      Prefix used for allocation a dataset
SUBXAUTH ===> DEFER         DEFER     RACF/JOBNAME/DEFER - IMS/DBE security
APIAUTH  ===> RACF          RACF      ORIGIN/RACF - Userid for MVAPI Execs
CICSECID ===> ORIGIN        ORIGIN    ORIGIN/RACF - Userid for CICS Commands
RESAUTH  ===> ORIGIN        ORIGIN    ORIGIN/RACF - Userid for Sysprog Cmds
EXEC     ===>                         (Enter ? to see high priority EXECs)
Press END to save changes. CANCEL to cancel changes
```

Use this panel to continue setting parameters for how single EXECs execute on the system. As on the EXEC Thresholds panel, this panel provides two columns of data:

- on the left, the value of the parameters in the AAOEXP.xx member for the target BBI-SS PAS
  
  You can enter your modifications on this side of the panel.

- on the right, the value of the parameters that are currently in effect for the target BBI-SS PAS; these values cannot be edited
From this panel, you can specify a question mark ( ?) in the EXEC ==> field to display another panel where you can specify the names of EXECs that will always execute on High priority threads.

Refer to “Setting high-priority EXECs” on page 91 for more information about specifying EXEC names for high-priority queuing.

### Describing the fields

Descriptions of the fields on this panel follow:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIMEXLIM</td>
<td>is the maximum CPU time limit allowed for an EXEC to use while executing. Valid values range from 0 to 9999 and the default is 0. Set this parameter to ensure that any EXEC does not execute limitlessly.</td>
</tr>
<tr>
<td>PEREXLIM</td>
<td>is the maximum CPU percentage usage that is allowed for an EXEC to use while executing. Valid values range from 0 to 99 and the default is 0. Set this parameter to ensure that any EXEC does not use a disproportionate amount of the CPU time.</td>
</tr>
<tr>
<td>TSOTIME</td>
<td>is the time-out value (in seconds) for attached TSO commands. Valid values range from 0 to 9999 and the default is 1200. Set this parameter to limit the amount of time that a TSO command associated with an EXEC can use.</td>
</tr>
<tr>
<td>ELAPSIM</td>
<td>is the elapsed time limit for all EXECs. If an EXEC is still running after the number of minutes specified, it will be canceled. Cancellation messages are sent to the BBI Journal that say an EXEC has exceeded the ELAPSIM limit. You can set elapsed time limits for an individual EXEC with the IMFEXEC CNTL ELAPSIM() keyword. Valid values range from 0 to 9999 and the default is 0.</td>
</tr>
<tr>
<td>UCPARMS</td>
<td>specifies that parameters passed to an EXEC must be converted to uppercase characters. Valid values are YES and NO and the default is YES. All CLIST EXECs require uppercase characters for parameters.</td>
</tr>
<tr>
<td>AUDITMSG</td>
<td>specifies whether you want to record in the BBI-SS PAS Journal all occurrences of BLDLs against the SYSPROC data set and all user-initiated EXEC requests. Valid values are YES and NO and the default is YES.</td>
</tr>
<tr>
<td>AUDITEXE</td>
<td>specifies whether you want to record in the BBI-SS PAS Journal audit messages indicating when EXECs start and end. Valid values are YES and NO and the default is NO.</td>
</tr>
<tr>
<td>OSPINUM</td>
<td>specifies the maximum number of OSPI ACBs that is allowed. Valid values range from 0 to 999 and the default is 0.</td>
</tr>
<tr>
<td>Field name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>OSPIPRFX</td>
<td>specifies the default four-character OSPI ACB prefix Valid values are any four-character string and the default is OSPI.</td>
</tr>
<tr>
<td>UNITNAME</td>
<td>is the default UNITNAME that is used for allocating a data set Valid values are any eight-character string and the default is SYSALLDA.</td>
</tr>
<tr>
<td>PREFIX</td>
<td>is the prefix that is used for allocating a data set in an EXEC if the data set name is not enclosed in quotation marks Valid values are any seven-character string and the default is blank, which refers to the subsystem ID.</td>
</tr>
</tbody>
</table>
| SUBXAUTH   | specifies one of the following names:  
|            | - **DEFER**: defers the use of the job name or user ID for the security check to the batch job  
|            | - **RACF**: uses the RACF user ID for all batch job security checks  
|            | - **USER ID**: uses the user ID for all batch job security checks  
|            | - **JOBNAME**: uses the job name for all batch job security checks The default is **DEFER**.  
|            | - **REXX**: shows the current REXX= setting Valid values are **YES** or **NO** and the default is **YES**.  
|            | **Restriction**: The number of EXECs that can be scheduled concurrently is equal to \( \frac{x - 3}{2} \) where \( x \) is the number of REBs. At BBI-SS PAS startup time, the following warning message is issued if the aggregate of MAXNORM and MAXHIGH exceeds this limit:  
|            | **EM0002W EXEC MULTITASKING EXCEEDS SITE LIMITS**  
|            | In addition, further scheduling of EXECs is inhibited if this limit is reached. |
| APIAUTH    | specifies the user ID to be used for MV API calls issued from EXECs with IMFEXEC MV commands Default value is RACF; valid values are as follows:  
|            | **ORIGIN** specifies that the EXEC origin is used as user ID for MV API call. For ALERT-initiated EXECs, the origin is the origin of the ALERT. For Rule-initiated EXECs, the origin is the SSID of the BBI-SS PAS started task. For ALERT escalation-initiated EXECs, the origin is the jobname of the BBI-SS PAS.  
|            | **RACF** the user ID associated with the BBI-SS PAS is used for MV API calls.  
<p>|            | <strong>Note</strong>: This parameter applies only when you are using the IMFEXEC MV API (IMFEXEC MV commands) and does not apply to the MainView Extended API (IMFEXEC MVX commands). |</p>
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICSECID</td>
<td>specifies whether the user ID or the current origin of the EXEC is used for security checking of an IMFEXEC CICS command that is used in a remote EXEC when it is scheduled by a local Rule-initiated EXEC. Default value is ORIGIN, which is used if you do not specify a value for this keyword; valid values are as follows: &lt;br&gt; <strong>ORIGIN</strong> specifies that the EXEC origin is the name of the originating BBI-SS PAS where the EXEC was scheduled and the name of that BBI-SS PAS is used during security checking. &lt;br&gt; <strong>RACF</strong> specifies that the security ID of the BBI-SS PAS is used during security checking. &lt;br&gt;In both cases, the user ID is used for security checking of CICS commands that are used in EXECs scheduled in a remote BBI-SS PAS that are called by a local EXEC.</td>
</tr>
<tr>
<td>RESAUTH</td>
<td>specifies the user ID that will be used when SYSPROG Services are invoked from MainView AutoOPERATOR EXECs. Default value is ORIGIN, which is used if you do not specify a value for this keyword; valid values are as follows: &lt;br&gt; <strong>ORIGIN</strong> specifies that the EXEC origin is used as user ID for Sysprog Services. For ALERT-initiated EXECs, the origin is the origin of the ALERT. An exception occurs when the EXEC origin contains the BBI-SS PAS SSID or the jobname of the BBI-SS PAS. In this case, the user ID that is associated with the BBI-SS PAS started task is used. &lt;br&gt; <strong>RACF</strong> specifies that the user ID associated with the BBI-SS PAS started task is used for SYSPROG Services.</td>
</tr>
<tr>
<td>EXEC</td>
<td>enables you to display another panel where you can define the names of EXECs that will always run on High priority threads. Refer to “Setting high-priority EXECs” on page 91 for more information about this panel.</td>
</tr>
</tbody>
</table>

**Setting high-priority EXECs**

To update the list of high-priority EXECs, enter a question mark ( ?) on the EXEC Parameters panel, next to the EXEC ====> field in the Member column. This displays the Update Non-Unique Parameter List panel.

*Figure 16 on page 91 shows an example of the panel.*

**Figure 16: Update Non-Unique Parameter List**

```
<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Update Non-Unique Parameter List</th>
<th>AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND</td>
<td>TGT --- BBSYSA</td>
<td></td>
</tr>
<tr>
<td>Parameter:</td>
<td>DATE --- YY/MM/DD</td>
<td></td>
</tr>
<tr>
<td>EXEC========</td>
<td>TIME --- 11:27:27</td>
<td></td>
</tr>
<tr>
<td>LC CMDS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--- (D)elete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXEC1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXEC2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXEC3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXEC4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXECABCD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
To add an EXEC for high-priority execution, type

ADD execname

where execname is the one- to eight-character name of the EXEC that you want to execute in the high-priority queue.

To delete an EXEC, use the line command Delete in the Cmd field next to the EXEC name that you want to delete.

Additional EXEC parameters

Use the EXEC Parameters 2 panel (see the following figure) to specify additional EXEC-related parameters.

Figure 17: EXEC Parameters 2 panel

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMLOG</td>
<td>specifies whether the windows-mode EMLOG view is available for viewing, sorting and filtering EXEC messages</td>
</tr>
<tr>
<td></td>
<td>Valid values are YES or NO. The default is NO.</td>
</tr>
<tr>
<td></td>
<td>In the EMLOG view, you can select to view (filter) all the messages from a specific EXEC. This filtering ability is not supported in the MVLOG view.</td>
</tr>
<tr>
<td></td>
<td>For more information about the windows-mode EMLOG view, see the MainView AutoOPERATOR Basic Automation Guide, Volume 2.</td>
</tr>
</tbody>
</table>
### EXHMC

Specifies whether you can use the IMFEXEC HMC command in EXECs. Valid values are YES or NO. The default is NO.

**WARNING:** The IMFEXEC HMC command allows EXECs to issue queries and commands to IBM mainframe hardware. IMFEXEC HMC provides capabilities that are similar to those allowed in the Hardware Management Console (HMC) and mainframe Support Element.

Using IMFEXEC HMC can alter your IBM machine, which might result in outages or incur unexpected financial costs. For example, using the IMFEXEC HFC command and its parameters can increase the machine’s capacity, which might increase costs.

BMC recommends that only users who have proficiency in operating the IBM System z Support Elements and Hardware Management Console (HMC) should use the IMFEXEC HMC command. You should also already be familiar with HMC and System z terminology and properties.

In addition, the user control and confirmation features available on the actual HMC do not apply when using the IMFEXEC HMC command in an EXEC issued from MainView AutoOPERATOR. For example, you can configure the HMC to require a user to enter a password before performing an IPL (issuing a Load command) of the system. With the IMFEXEC HMC command, the Load command executes without requiring any confirmation.

### AUDITRES

Specifies whether BBI Journal message EM0715I is written to the log when an EXEC issues IMFEXEC RES.

Use the message to help identify which of your automation EXECs use the IMFEXEC RES command. This information can be especially helpful when the MainView SYSPROG Services product releases new versions that introduce changes to SYSPROG commands, or services that might impact your automation.

### MSGID

Updates the TSO user profile MainView AutoOPERATOR uses to turn on (and off) the TSO MSGID option.

Use this parameter to specify whether MainView AutoOPERATOR should alter the TSO profile MSGID setting where:

- **Y** (YES): sets a PROFILE MSGID in the TSO user profile
- **N** (NO): sets a PROFILE NOMSGID in the TSO user profile
- **I** (IGNORE): specifies that the EXEC Manager will bypass any setting of the user profile

The default is Yes.
Modifying MainView AutoOPERATOR parameters in AAOPRMxx

BBPARM members AAOPRMxx contain the parameters that control certain functions of MainView AutoOPERATOR.

To modify an AAOPRMxx member, type S in the Cmd field of the Dynamic Parameter Manager panel. Figure 18 on page 94 shows an example of the panel.

You can use the ACTIVATE command or the BBI control command .RESET for BBPARM member AAOPRMxx to reset all of the values specified in AAOPRMxx and to restart all time-initiated Rules that are currently defined. These actions also occur any time the BBI-SS PAS is restarted with WARM or COLD specified.

**Note**

You cannot dynamically reset the values of the AAOPRMxx keywords AOOPTION and RULESET when you use the Activate command in DPM or use the BBI control command .RESET. The only way to reset the values for the AOOPTION and RULESET keywords is to restart the BBI-SS PAS with WARM or COLD specified.

The Dynamic Parameter Manager can update AAOPRMxx with the changes, however the PAS must be restarted (warm or cold) for the changes to take effect.

**Figure 18: Dynamic Parameter Manager panel—example 4**

Use this panel to see a list of all the AAOALSxx, AAOARPxx, AAOEXPxx, AAOPRMxx, and AAOTSPxx members in BBPARM.

From this panel, you can

- use the line commands; refer to Table 9 on page 72 for more information
use the primary command ADD to add a new members

To add a new member or members, type the following command on the COMMAND line, where xx is a new suffix for a new member:

ADD AAOPRMxx

Setting MainView AutoOPERATOR parameters

The Dynamic Parameter Manager member panel (the following figure) lists the categories of MainView AutoOPERATOR functions that you can modify.

Figure 19: MainView AutoOPERATOR primary parameters panel

The following sections describe each of these options from this panel.

Selecting General Parameters

This topic describes selecting general parameters.

When you use the Select line command to select General Parameters from Figure 19 on page 95, Figure 20 on page 95 is displayed.

Figure 20: MainView AutoOPERATOR General Parameters panel
Describing the fields

Descriptions of the fields on this panel follow:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOOPTION</td>
<td>allows you to limit the licensed MainView AutoOPERATOR product components that are started during BBI-SS PAS initialization. When this parameter is not specified, all MainView AutoOPERATOR components with valid BMC passwords are activated at the time of BBI-SS PAS startup. Enter a &quot;?&quot; next to the keyword to display a list of the current options that you can edit. Default value is blank; valid values are ANV, TSH, CAO, IAO, SHA, QAO, MAO. For best results, perform a cold start of the BBI-SS PAS after you make changes to the AOOPTION parameter.</td>
</tr>
<tr>
<td>SUPREPLY</td>
<td>suppresses commands that are sent as replies to WTOR in the BBI Journal. Default value is NO; valid values are YES or NO. Specify YES to have commands suppressed from view in the BBI Journal; for example, if you specify YES, and an EXEC issues a &quot;R 33,XYZ&quot;, the BBI Journal displays this reply as &quot;R 33,<em>SUPPRESSED</em>&quot;. Note that only the local journal will still display the command as &quot;R 33,XYZ&quot;.</td>
</tr>
<tr>
<td>IMFJNUM</td>
<td>specifies whether the variable IMFJNUM will be 5 digits in length with leading zeros or a variable length number without leading zeros. You must specify IMFJNUM=V to support JES job numbers greater than 99,999. Default value is V; possible values are 5 or V. Refer to “IMFJNUM option and seven-digit JES job numbers (z/OS version 1.2 and later)” on page 249 for more information.</td>
</tr>
</tbody>
</table>
### Field name | Description
--- | ---
MAXDLYQ | specifies the maximum concurrent number of delayed requests from Rules The default is 999; valid values are 0 to 999. Specifying zero means no maximum is set.  
The maximum value is 999 or half the value of the MAXPQE parameter that is specified in the BBPARM member BBISSPxx member, whichever value is lower.  
A delayed request may be for a command or message.  
When the number of concurrent DELAYED requests reaches 60 percent of the MAXDLYQ threshold, the following message is issued:  
\[\text{DY6160W} \quad xxxxx \quad \text{Delayed Requests queued}\]  
When the number of concurrent DELAYED requests reaches 75 percent of the MAXDLYQ threshold, the following message is issued:  
\[\text{DY6175W} \quad xxxxx \quad \text{Delayed Requests queued}\]  
If the MAXDLYQ value is reached, no additional requests can be queued until the number drops below the specified threshold and the following message is issued:  
\[\text{DY6199E} \quad \text{Maximum Delay Request queued}\]  
The MAXDLYQ threshold calculations are re-adjusted when the AAOPRMxx member is reset. The threshold is calculated every time a DELAYED request is queued.  
Delayed actions are not retained over a warm or cold restart of the BBI-SS PAS. Therefore if outstanding delayed actions exist when the BBI-SS PAS is restarted, those delayed actions are lost and are not re-queued when the BBI-SS PAS restarts.  
You can use the BBI command .DISPLAY DLY before you restart the BBI-SS PAS to review any delayed actions that have not yet been completed. Determine whether you should delay restarting the BBI-SS PAS or, restart the BBI-SS PAS but note that you need to reissue the delayed actions when the BBI-SS PAS restarts.

MSGJOBNM | specifies how MainView AutoOPERATOR should derive batch, Started Task, and TSO job names for MSG events  
The job name or the IMFOJOB variable can be used in the Rule Processor Selection Criteria panel while defining Rules. The following values are possible:  
ACTUAL: Use the job name as it is derived from the address space that creates the event. There are situations where this value might not match the address space that initiated the event (WTO). For example some JES3 messages will have a value of "CONSOLE".  
ORIGIN: Use the job name as it is derived from the MVS control block (WQE). Specifying ORIGIN would result in a more accurate name of the issuer of the WTO. ORIGIN is the default value.

SPLXVARS | specifies that the BBI-SS PAS supports sysplex variables  
The default value is N; valid values are Y(yes) or N(no).

SPLXPREF | specifies the prefix that denotes that the SHARED variable name is a sysplex variable  
The default value is AOSPLX_; valid values are 1 to 16 alphanumeric characters long.  
If you use the default setting, an example of a sysplex variable name would be AOSPLX_DB2STATUS
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPLXSTR</td>
<td>specifies the name of the XCF Structure that stores the sysplex variables. The XCF Structure resides in a Coupling Facility. Valid values can be 1 to 8 alphanumeric characters long. SPLXSTR uses the default value of the XCFGROUP option in BBISSPxx. BMC recommends that you avoid coding user-specified values for the XCFGROUP= and SPLXSTR= parameters and use the default values. The default value for XCFGROUP= is BMCAB and the default XCF Structure name will be MVAO_BMCAB.</td>
</tr>
<tr>
<td>UCCICTRN</td>
<td>specifies whether CICSTRAN transaction parameters will be translated to uppercase. This specification applies to CICSTRAN when executed from anywhere except the TS Command Execution Application (which supports mixed case specification of parameters). The following values are valid: N: specifies that CICSTRAN parameters are not translated to uppercase Y: specifies that CICSTRAN parameters are translated to uppercase The default value is Y.</td>
</tr>
<tr>
<td>UCVARS</td>
<td>specifies whether the Rules Processor should convert variable names to uppercase before resolving. The following values are valid: NO: do not make variable names uppercase before resolving the variable value YES: convert the variable name to uppercase before resolving Specify UCVARS=YES if you want the Rules Processor to convert variable names to uppercase. Two passes are made when resolving compound variables, which might result in a mixed case name. If the variable name is mixed case, you must specify UCVARS=YES for that variable to be resolved. <strong>Note:</strong> When creating Rules that allow you to send events from MainView AutoOPERATOR to a BMC Impact Manager cell, be aware that the slot fields on the Event Action (BA) or Alert Action (AE) panels are case sensitive. Variable names specified on this panel are converted to uppercase before they are resolved when UCVARS=YES is specified.</td>
</tr>
<tr>
<td>WTOEARLY</td>
<td>specifies that early WTOs, captured by the QAOCNZ15 utility, are available as HWTO events to the Rule Processor. This parameter is valid only when the BMC MainView Automation Solution is active in the MainView AutoOPERATOR BBI-SS PAS.</td>
</tr>
<tr>
<td>WTOINTRM</td>
<td>specifies that WTOs captured during brief MainView AutoOPERATOR outages are available as HWTO events to the Rule Processor. This option is ignored if the MainView AutoOPERATOR BBI-SS PAS abends or does not terminate normally. This parameter is valid only when the BMC MainView Automation Solution is active in the MainView AutoOPERATOR BBI-SS PAS.</td>
</tr>
</tbody>
</table>
Selecting Automation and Rule Set parameters

When you use the Select line command to select Automation and Rule Set from “Setting MainView AutoOPERATOR parameters” on page 95, Figure 21 on page 99 is displayed.

Figure 21: MainView AutoOPERATOR Automation and Rule Set Parms panel

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
</table>
| RULESET      | specifies one or more RULESETs to become active upon initialization of the BBI-SS PAS  
|              | You can specify the Rule Set name by the two character Rule Set suffix only or by specifying the complete Rule Set name.  
|              | Default value is 00.  
|              | Format of the statement is RULESET= xx for one set, or RULESET= (xx,yy,...,zz ) for multiple sets. The two-character code is appended to AAORUL.  
|              | You can enter a question mark (?) to display a list of the current Rule Sets and you can add or delete Rule Set names as needed. |
| RULESCAN     | specifies the automation strategy of F (FIRST), A (ALL), or I (INDIVIDUAL) to determine how Rule Sets and Rules are searched to match events  
|              | Valid values are as follows:  
|              | ■ F indicates only the first Rule with matching selection criteria is fired.  
|              | ■ A indicates Rules with matching selection criteria are fired.  
|              | ■ I indicates that each Rule Set can have its own (individual) automation strategy. Each Rule Set can be set with First or All.  
|              | This automation strategy is used when the BBI-SS PAS is either warm or cold started. |

Describing the fields

Descriptions of the fields on this panel follow:
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HONORMPF</td>
<td>specifies if the Rules Processor application honors MPF suppression. Default is N; valid values are Y(yes) or N(no). The HONORMPF parameter setting determines whether messages are handled by the Rule Processor when MPF suppression is active.</td>
</tr>
<tr>
<td>IMFDATE</td>
<td>specifies the format of the IMFDATE variable. The following list shows valid formats and an example of each:</td>
</tr>
<tr>
<td></td>
<td>■ YYYY/MM/DD; 2050/10/23</td>
</tr>
<tr>
<td></td>
<td>■ YYYYMMDD; 20501023</td>
</tr>
<tr>
<td></td>
<td>■ DD/MM/YYYY; 23/10/2050</td>
</tr>
<tr>
<td></td>
<td>■ DDMYYYYY; 23102050</td>
</tr>
<tr>
<td></td>
<td>■ MMDDYYYY; 10232050</td>
</tr>
<tr>
<td></td>
<td>■ MM/DD/YYYY; 10/23/2050</td>
</tr>
<tr>
<td></td>
<td>■ MMMDDYYYY; OCT232050</td>
</tr>
<tr>
<td></td>
<td>■ DDMMMYYYY; 23OCT2050</td>
</tr>
<tr>
<td>RULESAVE</td>
<td>specifies what happens when you save a Rule Set from the Rule Processor Detail Control panel. Valid options are as follows:</td>
</tr>
<tr>
<td></td>
<td>■ FIRST (the default) automatically saves the Rule Set in the first data set of the BBIPARM DD concatenation, regardless of the data set from which the Rule Set was read.</td>
</tr>
<tr>
<td></td>
<td>■ ORIGINAL saves the Rule Set in the data set from which it was originally read in the BBIPARM DD concatenation.</td>
</tr>
<tr>
<td></td>
<td>■ PROMPT lets you specify the data set (FIRST or ORIGINAL) to which to save the Rule Set.</td>
</tr>
<tr>
<td>RULEPROT</td>
<td>activates serialization on a Rule when it is edited, which prevents multiple users from updating the same Rule in the same BBI-SS PAS at the same time. When the parameter is not specified, the default is NO. The following values are valid:</td>
</tr>
<tr>
<td></td>
<td>■ NO does not activate serialization when a Rule is edited.</td>
</tr>
<tr>
<td></td>
<td>■ YES activates serialization within the scope of each PAS.</td>
</tr>
</tbody>
</table>
Selecting Automation Logger application (AutoLog) parameters

When you use the Select line command to select AutoLog Parameters from “Setting MainView AutoOPERATOR parameters” on page 95, Figure 22 on page 101 is displayed.

Figure 22: MainView AutoOPERATOR Event Logging (AutoLog) parameters panel

BMC Software ---------- Event Logging (AutoLog) ---------------- AutoOPERATOR
COMMAND ===>                                                 TGT --- BBSYSA
DATE --- YY/MM/DD
Member     Current                             TIME --- 08:45:44
AOLOGEV ===>                       (Enter ? for Event Type list)
AOLOGACT ===>  Y          Y         Send Rules Actions to Log (Yes/No)
AOREPLAY ===>  Y          Y         Activate Event Replay Log (Yes/No)
Press END to save changes,  CANCEL to cancel changes

Describing the fields

Descriptions of the fields on this panel follow:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOLOGEV</td>
<td>specifies one or more event types whose activity will be tracked by the Automation Logger application. Default value is NONE; valid values are ALL (all events are logged), ALRM, ALRT, BRC, CICS, CMD, DB2, EOM, EOS, EXT, IMS, JRNL, MQS, NVC, NVM, TIME, TLM, VAR. To activate the Automation Logger application, you must specify at least one event type. Enter a question mark (?) next to this keyword to display a list of the current values where you can add or delete event type names.</td>
</tr>
<tr>
<td>AOLOGACT</td>
<td>specifies whether a record of the actions taken by a Rule are sent to the Automation Logger application. To send a record of all actions taken by a Rule to the Automation Logger application, specify YES. Default value is NO; valid values are YES or NO.</td>
</tr>
<tr>
<td>AOREPLAY</td>
<td>specifies whether to send Event Replay records to the MainView Logger where you can review (replay) them at a later time. Default value is NO; valid values are YES or NO. This keyword is valid only if AOLOGEV is not set to NONE. Event Replay records will be saved only for the event types specified on the AOLOGEV statement. The MVLOGSS parameter in BBPARM member BBISSP00 specifies the SSID of the MainView Logger for the EMLOG records.</td>
</tr>
</tbody>
</table>
Selecting IMS parameters

When you use the Select line command to select IMS parameters from “Setting MainView AutoOPERATOR parameters” on page 95, Figure 23 on page 102 is displayed.

**Figure 23: IMS parameters panel**

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>IMS Parameters</th>
<th>AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member</td>
<td>Current</td>
<td></td>
</tr>
<tr>
<td>IMFORGN</td>
<td>LTERM</td>
<td>NONE</td>
</tr>
<tr>
<td>IMSPLTIM</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>IMSROUTE</td>
<td>ALL</td>
<td>ALL</td>
</tr>
<tr>
<td>IMSMSTIM</td>
<td>120</td>
<td>120</td>
</tr>
<tr>
<td>IMSSHTIM</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>IMSAUTH</td>
<td>USER</td>
<td>USER</td>
</tr>
<tr>
<td>IMSCMDCN</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>IMSPLAUD</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>IMSOMEX</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>DBCTLORG</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>

Press END to save changes, CANCEL to cancel changes

**Describing the fields**

Descriptions of the fields on this panel follow:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
</table>
| IMFORGN    | specifies that the LOCAL variable IMFORGN is set to the IMS LTERM name for IMS command and response  
Default is the VTAM node name. |
| IMSPLTIM   | specifies a user-defined timeout limit for waiting for a response from IMSplex commands that are issued from the Journal  
Default timeout limit is 30 seconds; valid values range from 5 to 9999 seconds.  
The IMSplex command timeout limit represents the maximum time to wait for a response from IMS. If a response is received from all IMS regions of the IMSplex before the timeout is reached, the response is displayed and waiting is terminated.  
If a response is not received from all IMS regions of the IMSplex before the timeout is reached, a partial response is displayed and the IMS Operations Manager terminates execution of the IMSplex command. |
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
</table>
| IMSROUTE | specifies the default routing for IMSplex commands issued from the MainView AutoOPERATOR Journal, Rules or EXECs. Valid values are ALL and LOCAL. Default is ALL. ALL: IMSplex commands are sent to all IMS regions of the IMSplex. LOCAL: IMSplex commands, issued without a ROUTE() keyword are sent only to the IMS region connected to the MainView AutoOPERATOR PAS. This parameter does not apply when:  
  - the ROUTE() parameter is specified with an IMSplex command  
  - the IMSPLEX() parameter on an IMSplex command from EXEC specifies IMSplex that does not include the IMS region connected to the MainView AutoOPERATOR PAS. |
<p>| IMSMSTIM | specifies the maximum time to wait between receiving segments of IMS multisegment messages. Arrival of the final segment terminates the wait and the message is processed by the Rules processor. If the time is exceeded without receiving a final segment from IMS, the wait is terminated and any future segments are ignored. The delay may occur due to system heavy workloads, a lower dispatching priority of the IMS issuing the message, or other IMS processing. Default wait time is 5 seconds; valid values range from 1 to 999 seconds. Specify this parameter if Rules automation is not processing the complete IMS multisegment message, for example if only the first message segment is logged on the Journal. <strong>Note:</strong> Increasing the default value might create a delay of Rules-based automation for IMS multisegment messages. |
| IMSSHTIM | specifies the internal time to wait for termination of the MainView AutoOPERATOR tasks in the IMS Control region address space during IMS shutdown. Default timeout limit is 2 seconds; valid values range from 1 to 999 seconds. Specify this parameter only if your IMS Control region takes unusually long to terminate and frequently experiences abend sA03 at shutdown. <strong>Note:</strong> Increasing the default value reduces the possibility of sA03 abends, but might extend the amount of time required for IMS to shut down. |</p>
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMSAUTH</td>
<td>specifies whether to pass the user ID to IMS with every IMS command that MainView AutoOPERATOR issues. Valid values are as follows:</td>
</tr>
<tr>
<td></td>
<td>- NONE (the default) does not pass the user ID.</td>
</tr>
<tr>
<td></td>
<td>- USER passes the user ID.</td>
</tr>
<tr>
<td></td>
<td>Standard IMS command security rules will be applied. For commands that are issued from the User Address Space (UAS), the user ID is the issuer of the command. For commands that are issued from automation such as Rules and EXECs, the user ID value is the user ID that is associated with the MainView AutoOPERATOR PAS.</td>
</tr>
<tr>
<td></td>
<td>This parameter applies to:</td>
</tr>
<tr>
<td></td>
<td>- Type-2 (IMSplex) commands</td>
</tr>
<tr>
<td></td>
<td>- Type -1 for IMS DBCTL with DBCTLCMD=MVS</td>
</tr>
<tr>
<td></td>
<td>- commands issued from the Journal</td>
</tr>
<tr>
<td></td>
<td>- line commands from IMS OPERATOR displays</td>
</tr>
<tr>
<td></td>
<td>- commands that Rules and EXECs issue</td>
</tr>
<tr>
<td></td>
<td>This parameter does not apply to IMS commands:</td>
</tr>
<tr>
<td></td>
<td>- Type-1 for IMS DBCTL with DBCTLCMD=IMS</td>
</tr>
<tr>
<td></td>
<td>- Type-1 for IMS DB/DC region</td>
</tr>
</tbody>
</table>

**Note:** This parameter is not related to MainView windows-mode or full-screen mode security.
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
</table>
| IMSCMDCN | specifies whether a confirmation panel is displayed when you manually enter an IMS action command from MainView AutoOPERATOR views or from the Journal. Valid values are NO and YES:  
  - NO (the default) does not display a confirmation panel.  
  - YES: displays a confirmation panel for all IMS action commands that you issue manually from MainView AutoOPERATOR views, or from the Journal.  
  **Note:** A confirmation panel is displayed for both Type-1 and Type-2 commands.  
  A confirmation panel is not displayed for non-action commands such as /DISPLAY and QUERY, or for commands issued by automation (such as Rules or EXECs).  
  Ensure that the User Address Space (UAS) is locally connected to the MainView AutoOPERATOR PAS where this parameter is specified. |
| IMSPLAUD | specifies whether MainView AutoOPERATOR and MainView for IMS version 5.1 and later issues audit messages to the Journal for all IMSplex (type-2) commands. Valid values are NO and YES:  
  - NO: (the default) audit messages are not issued.  
  - YES: audit messages, containing the user ID, are issued to the Journal for all IMSplex commands. For commands that originate from the user address space (UAS), the user ID is the issuer of the command. For commands that MainView AutoOPERATOR automation (such as Rules and EXECs) issues, the user ID is the MainView AutoOPERATOR PAS user ID. |
| IMSOMEX | specifies whether this PAS should receive IMSplex (type-2) commands from the AutoOPERATOR IMS OM Input user exit. Valid values are NO and YES:  
  - NO: The AutoOPERATOR IMS OM Input user exit does not broadcast IMS Type-2 commands to this PAS. IMP event type Rules do not fire.  
  - YES: (the default) when the MainView AutoOPERATOR for IMS product is active, the MainView AutoOPERATOR IMS OM Input user exit broadcasts IMS Type-2 commands to this PAS and IMP event type Rules will capture and automate these commands. |
**Field name** | **Description**
---|---
DBCTLORG | specifies whether the Journal logs IMS DBCTL commands with the origin of who issued the commands

Valid values are NO and YES:

- **NO**: (the default) the Journal logs IMS DBCTL commands with only the origin of the DBCTL job name.
- **YES**: the Journal logs IMS DBCTL commands with the origin as the name of the command issuer, in addition to logging the commands with the originating DBCTL job name.

For commands that are issued from the User Address Space (UAS), the origin is the TSO user ID. For automation-issued commands, such as Rules and Rule-initiated EXECs, the origin is the event-originating job name.

This parameter applies to the following types of IMS DBCTL commands that are:

- Issued from the Journal
- Line commands from IMS OPERATOR displays
- Issued by Rules and EXECs
- Issued from MainView for IMS views

---

**Selecting IBM MQ parameters**

When you use the Select line command to select IBM MQ from “Setting MainView AutoOPERATOR parameters” on page 95, Figure 24 on page 106 is displayed.

**Figure 24: IBM MQ panel**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>IBM MQ</th>
<th>AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND =====&gt;</td>
<td></td>
<td>TGT --- AOAO</td>
</tr>
<tr>
<td>DATE --- YY/MM/DD</td>
<td>TIME --- 09:37:21</td>
<td></td>
</tr>
</tbody>
</table>

Describing the fields

Descriptions of the fields on this panel follow:
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
</table>
| MQEV       | specifies that MainView AutoOPERATOR for MQ should automatically enable instrumentation events for a queue manager during connection if it is not already enabled. You can specify that all, none of selected events are enabled. When you specify multiple event types, you must code them consecutively, without using blank spaces or commas (does not apply if you are specifying Y or N). Valid values are as follows:  
  - N: do not enable any events  
  - Y: enable all events  
  - S: enable start/stop (strstpev) events  
  - L: enable local (localev) events  
  - I: enable inhibit (inhibtev) events  
  - P: enable performance (perfcvc) events  
  - R: enable remote (remoteev) events  
  - F: enable configuration (configev) events  
  - M: enable command (cmdev) events  
  - C: enable channel (chlev) events  
  - V: enable ssl (sslev) events  
  - B: enable bridge (bridgeev) events  
  - X: enables non-display command events (CMDEV(NODISPLAY))  
  The default value is NO. |
| MQGINHIB   | specifies the action MainView AutoOPERATOR should take when MainView AutoOPERATOR attempts to listen to a queue which is defined as GET(DISABLED). Possible settings are:  
  - JRN: Issue a message to the BBI Journal stating that MainView AutoOPERATOR cannot listen to the queue.  
  - WTO: Issue a write-to-operator (WTO) message stating that MainView AutoOPERATOR cannot listen to the queue.  
  - IGNORE: Take no action.  
  - ALTER: Alter the queue to GET(ENABLED).  
  The default value is JRN. |
| MQNSHARE   | specifies the action MainView AutoOPERATOR should take when MainView AutoOPERATOR attempts to listen to a queue which is defined as NOSHARE. Possible settings are:  
  - JRN: issue a message to the BBI Journal stating that MainView AutoOPERATOR cannot listen to the queue.  
  - WTO: issue a write-to-operator (WTO) message stating that MainView AutoOPERATOR cannot listen to the queue.  
  - IGNORE: take no action.  
  - ALTER: alter the queue to share.  
  The default value is JRN. |
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
</table>
| MQEVLPRC   | specifies the name of the IBM MQ Event Listener PROC  
Do not use this parameter if the IBM MQ Event Listener is not required. Refer to the MainView AutoOPERATOR for MQ Installation and User Guide for more information about the IBM MQ Event Listener and coexistence of MainView AutoOPERATOR and other BMC products that require the IBM MQ event queues. |
| QALIAS     | specifies whether MainView AutoOPERATOR for MQ should open and listen for messages on QALIAS type queues  
**Note:** If you specify an alias queue name in BBPARM AAOMQL00, in addition to specifying a base queue, any Rules that you have written for the same message for the base queue might also fire for the alias queue.  
For example, if the selection criteria for a Rule for messages going to the base queue does not specifically contain the queue name then the Rule might fire twice; once for the base queue and once for the alias queue. |
| MQALTUSR   | specifies an AlternateUserid to be specified when issuing MQSC or PCF MQ commands from MainView AutoOPERATOR  
The user ID is used when MQ commands are issued from an EXEC or from a Rule.  
There is no default; valid values are as follows  
RACF: specifies that the security ID of the BBI-SS PAS is used as the AlternateUserid for commands  
user ID: specifies that a valid user ID is used as the AlternateUserid  
This global parameter can be overridden with the ALTUSER() | AU() parameter of the IMFEXEC CMD TYPE(MQS) command or as an action from a MQ initiated Rule. When you use this parameter the MQ PUT1 options will include MQOO_ALTERNATE_USER_AUTHORITY, MQPMO_DEFAULT_CONTEXT and MQPMO_ALTERNATE_USER_AUTHORITY and the AlternateUserid will be placed in the object descriptor.  
Refer to the MainView AutoOPERATOR for MQ Installation and User Guide for more information about the IMFEXEC CMD TYPE(MQS) and issuing MQ commands from Rules. |
| MQAUPRMS   | specifies whether individual MQSC or PCF MQ commands that are issued from EXECs or Rules can use the ALTUSER parameter to override the global parameter MQALTUSR (when specified) and specify an AlternateUserid  
When no value is specified for the MQALTUSR parameter, this parameter allows or denies the use of AlternateUserid for MQ command.  
Default value is NO, which means that the ALTUSER value is ignored. |
Modifying TapeSHARE parameters in AAOTSP<sub>xx</sub>

BBPARM members AAOTSP<sub>xx</sub> contain the parameters that you can modify to customize the way that the MainView AutoOPERATOR TapeSHARE component automates the sharing of tape devices among your z/OS images.

Remember, the MainView AutoOPERATOR TapeSHARE component is designed to perform tape sharing automation without you needing to customize any of these parameters. For more information about TapeSHARE and how these parameters affect TapeSHARE, refer to the MainView AutoOPERATOR Options User Guide.

To modify an AAOTSP<sub>xx</sub> member, type <code>S</code> in the <code>Cmd</code> field of the Dynamic Parameter Manager panel.

**Figure 25 on page 109** shows an example of the panel.

**Figure 25: Dynamic Parameter Manager panel—example 5**

<table>
<thead>
<tr>
<th>RMC Software</th>
<th>Dynamic Parameter Manager</th>
<th>AutoOPERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND ==&gt;</td>
<td></td>
<td>TGT ==&gt; BBSYSA</td>
</tr>
<tr>
<td>Primary command: Add</td>
<td>Date --- YY/MM/DD</td>
<td>Time --- 11:41:36</td>
</tr>
</tbody>
</table>

**LC CMDS ---**

- (S)elect, (B)rowse w/o Symbolic Substitution
- (A)ctivate, (B(r))rowse w/ Symbolic Substitution

<table>
<thead>
<tr>
<th>Cmd</th>
<th>Member</th>
<th>Lib</th>
<th>VV.MM</th>
<th>Created</th>
<th>Changed</th>
<th>Size</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>__</td>
<td>AAOALS00</td>
<td>1</td>
<td>01.11</td>
<td>YY/MM/DD</td>
<td>YY/MM/DD</td>
<td>20:12</td>
<td>4</td>
</tr>
<tr>
<td>__</td>
<td>AAOALS01</td>
<td>1</td>
<td>00.01</td>
<td>YY/MM/DD</td>
<td>YY/MM/DD</td>
<td>13:30</td>
<td>3</td>
</tr>
<tr>
<td>__</td>
<td>AAOARP01</td>
<td>1</td>
<td>00.01</td>
<td>YY/MM/DD</td>
<td>YY/MM/DD</td>
<td>13:30</td>
<td>3</td>
</tr>
<tr>
<td>__</td>
<td>AAOEXP00</td>
<td>1</td>
<td>02.26</td>
<td>YY/MM/DD</td>
<td>YY/MM/DD</td>
<td>20:12</td>
<td>27</td>
</tr>
<tr>
<td>S</td>
<td>AAOTSP00</td>
<td>1</td>
<td>02.01</td>
<td>YY/MM/DD</td>
<td>YY/MM/DD</td>
<td>20:12</td>
<td>12</td>
</tr>
</tbody>
</table>

Use this panel to see a list of all the AAOALS<sub>xx</sub>, AAOARP<sub>xx</sub>, AAOEXP<sub>xx</sub>, AAOPRM<sub>xx</sub>, and AAOTSP<sub>xx</sub> members in BBPARM.

From this panel, you can

- use the line commands; refer to Table 9 on page 72 for more information
- use the primary command ADD to add a new members

To add a new member or members, type the following command on the COMMAND line, where <sub>xx</sub> is a new suffix for a new member:

`ADD AAOTSP<sub>xx</sub>`
Customizing TapeSHARE

Use the TapeSHARE Parameters panel to see and modify parameters for the TapeSHARE application and control how TapeSHARE automates the sharing of tape devices among your z/OS images.

Figure 26 on page 110 shows an example of this panel.

**Figure 26: TapeSHARE Parameters panel**

<table>
<thead>
<tr>
<th>Field</th>
<th>Member</th>
<th>Current</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARTNER</td>
<td></td>
<td>* LIST *</td>
<td>(Enter ? for a partner list)</td>
</tr>
<tr>
<td>TIMEOUT</td>
<td>120</td>
<td>120</td>
<td>Subsystem response timeout value</td>
</tr>
<tr>
<td>RETRYCNT</td>
<td>2</td>
<td>2</td>
<td>Times to retry a take request</td>
</tr>
<tr>
<td>RETRYINT</td>
<td>30</td>
<td>30</td>
<td>Interval between take retries</td>
</tr>
<tr>
<td>ACTION</td>
<td>DEFAULT</td>
<td>DEFAULT</td>
<td>Allocation failure action</td>
</tr>
<tr>
<td>NOTAKE</td>
<td></td>
<td></td>
<td>(Enter ? for a device list)</td>
</tr>
<tr>
<td>PREF</td>
<td>NO</td>
<td>NO</td>
<td>Enable / Force preferencing</td>
</tr>
<tr>
<td>PREFDEV</td>
<td></td>
<td></td>
<td>(Enter ? for a device list)</td>
</tr>
<tr>
<td>FREE</td>
<td>DEALLOC</td>
<td>DEALLOC</td>
<td>Deallocation procedure name</td>
</tr>
<tr>
<td>TRACE</td>
<td>NO</td>
<td>NO</td>
<td>Start / Stop TapeSHARE tracing</td>
</tr>
</tbody>
</table>

Press END to save changes, CANCEL to cancel changes.

This panel provides two columns of data:

- on the left, the value of the parameters in the AAOTSP xx member for the target BBI-SS PAS
  You can enter your modifications on this side of the panel.

- on the right, the value of the parameters that are currently in effect for the target BBI-SS PAS; these values cannot be edited

The first time you bring up this panel, the display-only fields match the input fields and default values are shown in both columns.

**Describing the fields**

A description of the fields on this panel follows.
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
</table>
| PARTNER   | specifies a BBI-SS PAS name (that can be associated with either one or a range of tape devices) that is a TapeSHARE PLEX partners You also can include with the BBI-SS PAS name, the device address (or a range of device addresses) that TapeSHARE cannot GIVE to the named partner. See examples below.  
**Example:**  
PARTNER=SYSA  
In this example, SYSA is a partner to this image.  
**Example:**  
PARTNER=SYSC,0123  
In this example, SYSC is a partner to this image and the device whose address is 0123 cannot be GIVEN to partner SYSC.  
**Example:**  
PARTNER=SYSB,0120-0127  
In this example, SYSB is a partner to this image and the devices whose addresses fall between 0120 and 0127 cannot be GIVEN to partner SYSB.  
If no values are associated with this parameter, TapeSHARE assumes that any BBI node defined in BBINOD00 member is a partner within a TapeSHARE PLEX. |
| TIMEOUT   | specifies the amount of time (in seconds) to wait for the partners to respond to a request for devices  
When this time is reached and a successful GIVE has not been completed, an Allocation Failed Event occurs unless you have specified a number of retries on the RETRYCNT parameter. See below.  
**Example:**  
TIMEOUT=20  
Valid values are 0 - 999 seconds and 120 seconds is the default. |
| RETRYCNT  | specifies the number of times that TapeSHARE attempts to satisfy a request for devices after an initial attempt has failed  
For example, suppose this system’s request for devices is not satisfied and three retries are specified. TapeSHARE attempts three more times to satisfy this request for devices.  
**Example:**  
RETRYCNT=3  
This parameter is used in conjunction with the RETRYINT parameter that specifies how long TapeSHARE waits between retry attempts.  
Valid values are 0-10 retries and 2 is the default. |
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RETRYNT</td>
<td>specifies the number of seconds that TapeSHARE waits for devices between attempts after an initial attempt has failed. For example, suppose:</td>
</tr>
<tr>
<td></td>
<td>■ the RETRYCNT= parameter is set to 3 retries</td>
</tr>
<tr>
<td></td>
<td>■ the RETRYINT= parameter is set to 45 seconds</td>
</tr>
<tr>
<td></td>
<td>■ this system’s request for devices is not satisfied</td>
</tr>
<tr>
<td></td>
<td>TapeSHARE waits 45 seconds after the initial attempt fails and retries to satisfy the request (retry attempt 1). If this attempt fails, TapeSHARE waits another 45 seconds and retries (attempt 2). If this attempt also fails, TapeSHARE waits another 45 seconds and retries (attempt 3). Should all three attempts fail, an Allocation Failed Event occurs and the specified action is taken.</td>
</tr>
<tr>
<td>REExample:</td>
<td>RETRYCNT=3</td>
</tr>
<tr>
<td></td>
<td>Valid values are 0 - 300 seconds and 30 is the default.</td>
</tr>
<tr>
<td>ACTION</td>
<td>specifies the action that TapeSHARE should take when an Allocation Failed Event occurs because a successful GIVE cannot be performed. Possible values and their definitions are as follows:</td>
</tr>
<tr>
<td></td>
<td>■ DEFAULT: specifies that the installation's default action is to occur</td>
</tr>
<tr>
<td></td>
<td>■ WTOR: issues a WTOR to the operator</td>
</tr>
<tr>
<td></td>
<td>■ If the site does not have a defined default action, this action is the default.</td>
</tr>
<tr>
<td></td>
<td>■ CANCEL: cancels the job</td>
</tr>
<tr>
<td></td>
<td>■ NOHOLD: specifies that the job waits without holding resources</td>
</tr>
<tr>
<td></td>
<td>■ HOLD: specifies that the job waits while holding resources</td>
</tr>
<tr>
<td>Example:</td>
<td>ACTION=WTOR</td>
</tr>
<tr>
<td>Field name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| NOGIVE     | specifies a list of device addresses (or a range of addresses) that should not be GIVEN to other request devices to TAKE  
*Note:* This action does not prevent you from manually GIVING a NOGIVE device.  
If a device address is not associated with this parameter, TapeSHARE assumes that no devices are dedicated to this image.  
*Example:*  
NOGIVE=0123  
or  
NOGIVE=0120-012F  
You also can use this parameter in conjunction with the NOTAKE parameter. By specifying a device address on both these parameters, the device will be a NOGIVE-NOTAKE device and it will be excluded completely from TapeSHARE control.  
*Example:*  
NOGIVE=0123  
NOTAKE=0123  
If the device address 0123 is specified on both the NOGIVE and NOTAKE parameters, the 0123 is completely excluded from TapeSHARE control.  
TapeSHARE does not attempt to GIVE or TAKE this device.  
The default value is none. |
| NOTAKE     | specifies a list of device addresses (or a range of addresses) that this image should not TAKE when a request for resources is made  
TapeSHARE on this image will not accept (TAKE) the devices specified. If a value is not associated with this parameter, TapeSHARE assumes that it can accept any tape devices when a request for resources is made.  
*Example:*  
NOTAKE=0123  
or  
NOTAKE=0120-012F  
You also can use this parameter in conjunction with the NOGIVE parameter. By specifying a device address on both these parameters, the device will be a NOGIVE-NOTAKE device and it will be excluded completely from TapeSHARE control.  
*Example:*  
NOGIVE=0123  
NOTAKE=0123  
If the device address 0123 is specified on both the NOGIVE and NOTAKE parameters, the 0123 device is completely excluded from TapeSHARE control.  
TapeSHARE does not attempt to GIVE or TAKE this device.  
The default value is none. |
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
</table>
| PREF       | specifies whether Preferencing is activated  
If activated, TapeSHARE uses the devices identified on the PREFDEV parameter  
for only the operation (input or output) identified for that device.  
This parameter must be used in conjunction with the PREFDEV parameter.  
Specifying YES turns on Preferencing in Guideline mode. For example, in  
Guideline mode, if SYSA needs an output-only device but only input-only devices  
are available, the input-only device is GIVEN to SYSA.  
Specifying FORCE turns on Preferencing in Force mode. If Force mode were in  
place, the input-only device would not be GIVEN to SYSA and an Allocation  
Failure Event would occur.  
Example: |
|            | PREF=YES  
or  
PREF=NO  
or  
PREF=FORCE |
| PREFDEV    | specifies a tape device address (or a range of tape device addresses) followed by  
either an I for input-only or an O for output-only; see examples  
This parameter must be used in conjunction with the PREF parameter.  
With the PREF activated, when a device is needed for an output-only operation,  
TapeSHARE attempts to choose from those devices identified by this parameter as  
output-only.  
If a value is not associated with this parameter, TapeSHARE will not invoke device  
preferencing.  
Example: |
|            | PREFDEV=01A0-01A3,I  
or  
PREFDEV=0130,O  
or  
PREFDEV=0150-015F,O  
Default value is none. |
| FREE       | For use with MVS SP4.x only  
specifies a 1- to 8-character deallocation procedure which starts after TapeSHARE  
issues the VARY OFFLINE command to GIVE a device to another image  
Scheduling this procedure triggers deallocation processing in the system GIVING  
the device and places the device in an OFFLINE state.  
The default deallocation name is DEALLOC. |
Dynamically implementing new settings with the Activate command

The Dynamic Parameter Manager Activate line command can be used to dynamically change the BBPARM members currently in use by a BBI-SS PAS without restarting the BBI-SS PAS.

The command is specified on the main Dynamic Parameter Manager panel (see examples in “Accessing the Dynamic Parameter Manager” on page 70 and “Modifying ALERT thresholds in AAOALSxx” on page 74.)

Figure 27 on page 115 shows the Confirm Parameter Activation panel that is displayed after the Activate line command is entered.

Figure 27: Confirm Parameter Activation panel

Automation is suspended briefly while the BBPARM member is read and new control blocks are built. Other products operating in the target BBI-SS PAS continue to operate.

Describing the fields

Descriptions of the fields on this panel follow:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRACE</td>
<td>specifies that diagnostic information be written to the BBITST DD statement</td>
</tr>
</tbody>
</table>

Example:

```
TRACE=NO
```

If YES is specified, the BBITST DD statement must appear in the BBI-SS PAS startup JCL for information to be written to it (see BBSAMP member SSJCL).
### Field name | Description
--- | ---
Current Target | is the ID of the current target
Current SSID | is the ID of the current BBI-SS PAS
Current Parm | is the name of the member of BBPARM that was used to build the current parameters in the BBI-SS PAS
Replaced by | is the name of the BBPARM member you want to use to build the new set of parameters in the BBI-SS PAS
Last update | is the date and time of the last change to this member
User | is the user ID that made the last change to this member
Sending Aux Log actions to MainView Logger

The MainView Logger is a stand-alone logging facility that allows any BMC product, or the MainView Infrastructure itself, to log important messages or events in a centralized repository. MainView AutoOPERATOR uses the MainView Logger in a variety of ways. For example, MainView AutoOPERATOR can write event records to Aux Log if warranted when a Rule fires for an event.

Enabling logging to MainView Logger

Edit the MVLOGSS parameter in BBPARM member BBISSPxx to write Aux Log events to the Mainview Logger.

Note

If you are using the default logger in the LPAR, you do not need to edit the MVLOGSS parameter. However, if you want to write records to a specific logger in the LPAR other than the default logger, you must use the MVLOGSS parameter to specify the SSID that you want to use.

Aux Log uses a product-specific logspace to hold the Aux Log data. The default name for this logspace is AOAUXLOG.

To enable Aux Log logging

1. Create the product-specific logspace in the MainView Logger task.

   For more information about this process and creating the required datasets for the logspace, see the MainView Customization Reference. To review a sample logspace definition, see BBPARM member LOGGERAO.
Tip

Use the default name AOAUXLOG for the product-specific logspace. Using AOAUXLOG can minimize the number of changes that you need to make. Using the default also gives you more flexibility in managing data retention separately from other MainView Logger data.

However, if you have multiple MainView AutoOPERATOR product-specific logspaces, use a logspace name other than the default. An example is if you use a separate logspace for each BBI-SS PAS that writes data; specifying a logspace name other than the default enables you to manage data retention separately.

Only one instance of a product-specific logspace name can be active in an LPAR at one time. To specify a name other than the default, you must code the AUXLOGNM= keyword in the BBI-SS PAS BBPARM member AAOPRMxx.

2 Start (or restart) MainView Logger and verify that the MainView AutoOPERATOR product-specific logspace is initialized.

For more information about setting up, enabling, starting, and using MainView Logger, see the MainView Administration Guide and the MainView Customization Reference.

Accessing the AUXLOG view

The following procedure describes accessing the AUXLOG view.

1 Enter option P on the MainView AutoOPERATOR primary option menu to enter windows-mode.

2 Select AUXLOG from the EZAO menu or type AUXLOG on the COMMAND line, and press Enter.

The following is an example of the AUXLOG view:

<table>
<thead>
<tr>
<th>Time</th>
<th>Date</th>
<th>Event Type</th>
<th>Message</th>
<th>Origin</th>
<th>Job</th>
</tr>
</thead>
<tbody>
<tr>
<td>21:28:07</td>
<td>2012/08/08</td>
<td>MSG</td>
<td>IST663I</td>
<td>CNMNETD</td>
<td>AAO</td>
</tr>
<tr>
<td>21:29:47</td>
<td>2012/08/08</td>
<td>MSG</td>
<td>IST663I</td>
<td>CNMNETD</td>
<td>AAO</td>
</tr>
<tr>
<td>21:30:10</td>
<td>2012/08/08</td>
<td>MSG</td>
<td>IST663I</td>
<td>CNMNETD</td>
<td>AAO</td>
</tr>
<tr>
<td>21:30:40</td>
<td>2012/08/08</td>
<td>MSG</td>
<td>IST663I</td>
<td>CNMNETD</td>
<td>AAO</td>
</tr>
<tr>
<td>21:30:43</td>
<td>2012/08/08</td>
<td>MSG</td>
<td>IST663I</td>
<td>CNMNETD</td>
<td>AAO</td>
</tr>
<tr>
<td>21:30:45</td>
<td>2012/08/08</td>
<td>MSG</td>
<td>IST663I</td>
<td>CNMNETD</td>
<td>AAO</td>
</tr>
<tr>
<td>21:30:51</td>
<td>2012/08/08</td>
<td>MSG</td>
<td>IST663I</td>
<td>CNMNETD</td>
<td>AAO</td>
</tr>
<tr>
<td>21:30:53</td>
<td>2012/08/08</td>
<td>MSG</td>
<td>IST663I</td>
<td>CNMNETD</td>
<td>AAO</td>
</tr>
<tr>
<td>21:30:56</td>
<td>2012/08/08</td>
<td>MSG</td>
<td>IST663I</td>
<td>CNMNETD</td>
<td>AAO</td>
</tr>
<tr>
<td>21:31:04</td>
<td>2012/08/08</td>
<td>MSG</td>
<td>IST663I</td>
<td>CNMNETD</td>
<td>AAO</td>
</tr>
<tr>
<td>21:31:08</td>
<td>2012/08/08</td>
<td>MSG</td>
<td>IST663I</td>
<td>CNMNETD</td>
<td>AAO</td>
</tr>
<tr>
<td>21:31:10</td>
<td>2012/08/08</td>
<td>MSG</td>
<td>IST663I</td>
<td>CNMNETD</td>
<td>AAO</td>
</tr>
<tr>
<td>21:31:13</td>
<td>2012/08/08</td>
<td>MSG</td>
<td>IST663I</td>
<td>CNMNETD</td>
<td>AAO</td>
</tr>
<tr>
<td>21:31:51</td>
<td>2012/08/08</td>
<td>MSG</td>
<td>IST663I</td>
<td>CNMNETD</td>
<td>AAO</td>
</tr>
</tbody>
</table>
For more information about the AUXLOG view, see the online Help.
MVS console considerations

This chapter describes MVS console considerations.

Overview

MVS commands can be issued by MainView AutoOPERATOR terminal session (TS) users, EXECs, or Rules. When MVS commands are issued by TS users and from EXECs, the TS user or the EXEC can also wait for and inspect the response from the command.

Extended MCS (EMCS) consoles allow MainView AutoOPERATOR to issue MVS command and inspect the responses from these commands.

The use of consoles is not a consideration for MVS commands issued from Rules because MainView AutoOPERATOR Rules never expect a response. However, some commands issued by TS users and EXECs require a response to the command and therefore, require a MVS console.

For more information about how EXECs use consoles, refer to the MainView AutoOPERATOR Advanced Automation Guide where the IMFEXEC CMD statement is documented.

What extended MCS consoles (EMCS) are

EMCS consoles are identified by an 8-character name and a 4-byte ID.

Note

With versions of z/OS earlier than 1.8, EMCS also could have an optional 1-byte console ID known as a Migration ID. With z/OS release 1.8 and later, EMCS consoles do not support 1-byte Migration IDs.

Upon startup, MainView AutoOPERATOR dynamically allocates EMCS consoles named with the format
SSIDqqnn

where

SSID

is the subsystem ID (SSID) of the BBI-PAS subsystem

qq

is the 2-character value from the BBPARM member BBISSP00 parameter CONSQUAL=qq

The default value is 00. When there are BBI-PASs with the same SSID in a sysplex, the CONSQUAL parameter allows you to create unique EMCS console names for MainView AutoOPERATOR PASs that use identical SSIDs. Supported characters for the CONSQUAL value include letters A through Z, numbers 0 through 9, and the characters at sign (@), pound sign (#) and dollar sign ($).

nn

is a number from 1 to 0099 that represents the total number of EMCS consoles created

The nn value is derived from the 2-digit value from the BBPARM member BBISSP00 parameter CONSOLES=nn.

Example

SSID=ABCD, CONSQUAL=not-specified, CONSOLES=2

results in the following three EMCS consoles being activated:

ABCD0000
ABCD0001
ABCD0002

SSID=DEFG, CONSQUAL=34, CONSOLES=2

result in the following three EMCS consoles being activated:

DEFG3400
DEFG3401
DEFG3402
WARNING
You must ensure no other application allocates consoles with the same names or MainView AutoOPERATOR will encounter an error when attempting to activate these consoles.

Ensure that all BBI-PASs in a sysplex must have unique SSID$q$q$n names. It is recommended all BBI-PASs have unique SSID$s$s$s$s even across sysplexes.

MainView AutoOPERATOR uses the first EMCS console allocated, SSID0000, to issue MVS commands without response, and therefore console SSID0000 is known as the no-response console.

To allocate the next group of EMCS consoles, you must enter the number of additional no-response consoles in BBPARM member BBISSP00 with the CONSOLES parameter.

Refer to the MainView Customization Reference for more information about specifying values for the CONSOLES and MIGCONS parameters in BBPARM member BBISSP00.

Determining how many consoles to allocate

This section describes how to determine how many consoles MainView AutoOPERATOR needs to allocate to function effectively.

EMCS consoles are a shared resource. MainView AutoOPERATOR allocates an EMCS console to an EXEC when it is scheduled and also when a TS user issues an MVS command, an EMCS console is allocated for the duration of the MVS command. In both these cases, the console is in use until the command response has been received or the time out period expires.

Generally a small number of EMCS consoles can satisfy the needs of many EXECs and TS users. If an EXEC requires an EMCS console and all of the consoles are currently in use, the EXEC waits until one becomes available.

For best performance

1  Determine the maximum number of EXECs that will issue MVS commands concurrently.

   Note
   The optimum number of EXECs threads is typically the lowest number needed to provide adequate throughput for your workload.

2  Determine the maximum number of TS users that are likely to issue MVS commands concurrently.
3 Add these two numbers.

4 Specify the total number in the CONSOLES parameter in member BBISSP00.

For example if you anticipate total of 5 EXECs and 2 TS users might all issue a MVS command simultaneously, in BBPARM member BBISSP00 specify

\[ \text{CONSOLES}=7 \]

\[ \textbf{Note} \]
The Continuous State Manager (CSM) application requires a minimum of three EMCS consoles.

---

**CICS considerations**

If you plan to issue commands to CICS from a MVS console, that console must be defined to CICS by using the Resource Definition Facility (RDF) or by using the CICS Autoinstall process.

When you dynamically define MVS consoles to CICS using the CICS Autoinstall Facility, in most cases the CICS Autoinstall Facility removes any need to use the CMDCONS parameter in BBPARM member BBISSP00.
Propagating enqueues to other MVS systems

Using the IMFEXEC command VENQ in REXX EXECs and CLISTs results in MVS enqueues for resources with a major name of BBIUSER and a minor name of the user-specified value. Therefore, you must ensure that SYSTEMS enqueues with the major name BBIUSER are propagated to all members of the Global Resource Serialization (GRS) complex.

In particular, if you use the Computer Associates (CA) product MIM (Multi Image Manager), you must verify that SYSTEMS enqueues with the major name BBIUSER are propagated. Because you might have CA-MIM configured to downgrade SYSTEMS enqueues automatically, which prevents the BBIUSER enqueues from being propagated to other MVS systems.

If you use the CA-MIM product, CA-MIM customer support can help you verify that SYSTEMS enqueues with the major name BBIUSER are being correctly propagated to all MVS systems.
Implementing TapeSHARE

This chapter describes how to implement the TapeSHARE for MainView AutoOPERATOR component. Some steps are optional.

Overview

TapeSHARE runs in a BBI-SS PAS and an image can have only one BBI-SS PAS where TapeSHARE is installed. Steps 1 and 2 are required and they must be performed on every BBI-SS PAS within the TapeSHARE PLEX.

Note

To use TapeSHARE, you must make sure the z/OS Allocated/Offline Device Exit is not specified on any image where TapeSHARE will run. TapeSHARE is defined dynamically as an Allocated/Offline Device Exit to z/OS when it initializes.

Prerequisites

To use TapeSHARE, the only prerequisites are that you have a valid MainView AutoOPERATOR for z/OS password key and that you have a password key for the TapeSHARE for MainView AutoOPERATOR component. See “Step 1: Entering the password key for TapeSHARE” on page 127 for more information about password keys.

Step 1: Entering the password key for TapeSHARE

Use the following procedure to enter the password key for TapeSHARE.
1 TapeSHARE requires that you have a password key to use it. Refer to the *Installation System User Guide* for information about how to install and activate the passwords.

2 Go to “Step 2: Activating TapeSHARE” on page 128.

3 Repeat these steps for every BBI-SS PAS that is a partner in the TapeSHARE PLEX as part of the TapeSHARE PLEX installation.

### Step 2: Activating TapeSHARE

After the password key is implemented, all you need to do is restart the BBI-SS PAS and TapeSHARE is activated for that image. *You do not have to restart z/OS.*

For every image that you want to include in a TapeSHARE PLEX, you must add a password key for that image and restart the BBI-SS PAS.

### Using TapeSHARE in its default state

TapeSHARE is designed for you to install and use in its out-of-box default state.

If you choose to use TapeSHARE in its default state, every BBI node defined in the BBINOD00 member with a TapeSHARE installed will be considered a partner in the TapeSHARE PLEX.

For more information about what values are used in the default state, refer to “Modifying TapeSHARE parameters in AAOTSPxx” on page 109.

### Step 3: Securing TapeSHARE (optional)

Complete this optional step if you decide to secure who has access to the TapeSHARE for MainView AutoOPERATOR component.

Security for all MainView AutoOPERATOR options is described in the BMC Software document *MainView Security Reference Guide*.

Securing access to TapeSHARE is the same as securing access to any of the MainView AutoOPERATOR applications, which is described in the *MainView Security Reference Guide* manual.

The resource name for securing access to the TapeSHARE component is `prefix.ssid.AAO.target.TAPSHR`. 
After TapeSHARE is installed

After TapeSHARE is installed, it can perform all of your tape device sharing without ever performing any customization steps. Refer to the *MainView AutoOPERATOR Options User Guide* for information about how to customize and use TapeSHARE.
Implementing Automatic Restart Manager (ARM)

This chapter describes implementing Automatic Restart Manager (ARM).

Overview

The Automatic Restart Manager (ARM) is an IBM facility that allows the IBM MVS/ESA product to automatically restart Started Tasks (excluding TSO users) after abnormal termination.

By implementing support for ARM, you can ensure that MainView AutoOPERATOR will be restarted quickly and automatically after an abend with no operator intervention required.

Previously, to ensure MainView AutoOPERATOR availability, you had to use a second MainView AutoOPERATOR to monitor the primary MainView AutoOPERATOR subsystem or use an outboard processing facility. If the primary MainView AutoOPERATOR abended, user-defined automation routines in the secondary MainView AutoOPERATOR (or outboard facility) would attempt to restart MainView AutoOPERATOR.

ARM eliminates the need for a secondary MainView AutoOPERATOR system (or outboard) to monitor MainView AutoOPERATOR availability and can simplify MainView AutoOPERATOR restarts by allowing MVS to monitor and restart MainView AutoOPERATOR.

Implementation considerations

To implement ARM support for MainView AutoOPERATOR, you must modify the ARMPOLCY parameter in BBPARM member BBISSP00 (see “Enabling ARM support” on page 133). The following choices are available:
- Do not implement ARM support.
- Implement ARM support with the default ARM policy.
- Implement ARM support with your own user-designed policy.

Implementing ARM support: default ARM policy

The default ARM restart policy is that ARM attempts to restart an abended Started Task up to three times in a five-minute interval. After three attempts and the Started Task has still failed to restart, ARM stops attempting restarts and a message is written to SYSLOG.

Therefore, if you choose to use the default ARM policy for MainView AutoOPERATOR, ARM will attempt to restart an abended MainView AutoOPERATOR subsystem up to three times in a five-minute interval.

The default ARM policy uses the same start command that was used to start the MainView AutoOPERATOR subsystem originally. After the third attempt, if MainView AutoOPERATOR has not successfully restarted, a message is written to SYSLOG and restart attempts stop.

If you decide that ARM should perform more restarts or perform them over a different period of time, you can write your own ARM restart policy. Refer to “Implementing ARM support: creating a user-defined policy” on page 132.

Implementing ARM support: creating a user-defined policy

You might already have an ARM restart policy in place or you can create your own ARM restart policy.

To create your own policy, refer to the IBM publication Setting Up a Sysplex. In either case, MainView AutoOPERATOR provides support for a user-defined ARM restart policy. “Enabling ARM support” on page 133 describes how to enter the name of a user-defined ARM restart policy in BBPARM member BBISSP00.

Note

Only one ARM policy can operate in an z/OS image at one time.
Enabling ARM support

To enable ARM support, edit the BBPARM member BBISSP00 and specify

\[
\text{ARMPOLCY=\{NO|YES|x\ldots\}}
\]

where valid values are

- **NO**: specifies that the BBI-SS PAS will not use the ARM facility; the default setting
- **YES**: specifies that the BBI-SS PAS will use the ARM default restart policy
  The default ARM restart policy is that ARM will restart an abended address space up to three times within five minutes. If the address space fails to restart after the third attempt, a message is written to the SYSLOG and the address space remains unstarted.
- **x\ldots**: Specifies a 1- to 8-character name of a user-created policy to be used to control BBI-SS PAS restarts

When you have completed modifying the ARMPOLCY parameter, restart the BBI-SS PAS. Either a warm start or a cold start will cause the new ARM policy to be recognized.

The *MainView Customization Reference* describes how to implement ARM support during manual customization.

Stopping MainView AutoOPERATOR when ARM is enabled

You might find that you need to stop MainView AutoOPERATOR when ARM is enabled and you do not want ARM to restart MainView AutoOPERATOR. To stop (or cancel) MainView AutoOPERATOR without ARM attempting a restart:

- Use the \texttt{STOP\{P} command for a normal shutdown of MainView AutoOPERATOR; for example:
  \[
  \texttt{P ao_ssid}
  \]
Use the `CANCEL | C` or `FORCE` command if you unexpectedly need to cancel MainView AutoOPERATOR; for example:

```
C ao_ssid
or
FORCE ao_ssid
```

where `ao_ssid` is the subsystem ID of the MainView AutoOPERATOR that you want to stop or cancel.
Implementing support for MPF

This chapter describes how MainView AutoOPERATOR can coexist with the MVS Message Processing Facility (MPF). If your site does not use MPF, you can skip this chapter.

Overview

The MVS Message Processing Facility (MPF) allows you to modify the attributes of write-to-operator messages (WTOs). For example, with MPF you can modify the appearance of a WTO (such as its color and highlighting) and suppress messages from consoles.

Previous releases of MainView AutoOPERATOR provided the HONORMPF=(YES | NO) parameter, which determined how MainView AutoOPERATOR automated WTOs with the MPF SUP keyword setting.

- If HONORMPF is set to the default value of NO:
  HONORMPF=NO
  all WTOs are eligible for MainView AutoOPERATOR automation (regardless of how the SUP keyword is set on the WTO).

- If HONORMPF is set to YES:
  HONORMPF=YES
  only WTOs with the SUP keyword set to NO are eligible for MainView AutoOPERATOR automation.

You can specify that MainView AutoOPERATOR will determine which WTOs are eligible for automation by checking the WTO for both the SUP and the AUTO keyword setting.

The possible values of the MPF AUTO keyword are

AUTO=(YES|NO|token)
where token is an up to 8-character user-specified token name. The token name can contain wildcards and blanks.

The possible values of the MPF SUP keyword are

\[
\text{SUP} = (\text{YES} | \text{NO})
\]

**Note**
For more information about MPF, refer to the IBM publication *MVS/ESA Initialization and Tuning Reference*.

To check a WTO for its AUTO and SUP settings, use the MPFFILTR= parameter in BBPARM member AAOPRM00. For more information, refer to “Implementation considerations” on page 136.

### Implementation considerations

Use the MPFFILTR parameter when you want MainView AutoOPERATOR to check the MPF AUTO and SUP settings of a WTO to determine if the WTO should be eligible for MainView AutoOPERATOR automation.

**Note**
You cannot use the MPFFILTR parameter if HONORMPF=YES is set. If HONORMPF=YES is set, MainView AutoOPERATOR ignores any settings on the MPFFILTR= parameter.

When the MPFFILTR parameter is set, MainView AutoOPERATOR actively checks the AUTO and SUP settings and only WTOs that match the MPFFILTR criteria are eligible for automation.

For more information, refer to “Setting the MPFFILTR parameter” on page 136.

### Setting the MPFFILTR parameter

The MPFFILTR parameter allows you specify how MainView AutoOPERATOR handles WTOs that have the MPF AUTO and SUP keywords associated with them.

The syntax for MPFFILTR parameter is

\[
\text{MPFFILTR} = (\text{AUTO=} \text{YES} | \text{NO} | \text{token}, \text{SUP=} \text{YES} | \text{NO})
\]
The AUTO and SUP keywords always must be used together on the MPFFILTR statement. You cannot use one keyword without the other.

The possible combinations of the AUTO and SUP keywords are

<table>
<thead>
<tr>
<th>MPFFILTR</th>
<th>AUTO=NO,SUP=YES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUTO=NO,SUP=NO</td>
</tr>
<tr>
<td></td>
<td>AUTO=YES,SUP=YES</td>
</tr>
<tr>
<td></td>
<td>AUTO=YES,SUP=NO</td>
</tr>
<tr>
<td></td>
<td>AUTO=*,SUP=NO</td>
</tr>
<tr>
<td></td>
<td>AUTO=*,SUP=YES</td>
</tr>
<tr>
<td></td>
<td>AUTO=NO,SUP=*</td>
</tr>
<tr>
<td></td>
<td>AUTO=YES,SUP=*</td>
</tr>
<tr>
<td></td>
<td>AUTO=<em>,SUP=</em></td>
</tr>
<tr>
<td></td>
<td>AUTO=<em>,SUP=</em></td>
</tr>
<tr>
<td></td>
<td>AUTO=TOKEN,SUP=YES</td>
</tr>
<tr>
<td></td>
<td>AUTO=TOKEN,SUP=NO</td>
</tr>
<tr>
<td></td>
<td>AUTO=TOKEN,SUP=*</td>
</tr>
</tbody>
</table>

You can select up to four of the possible combinations. Once you use any of the combinations, MainView AutoOPERATOR checks all WTOs for a match between the WTO’s settings and the MPFFILTR setting.

If the WTO’s setting matches the MPFFILTR setting, the WTO is eligible for automation by MainView AutoOPERATOR. The following examples describe some of the possible MPFFILTR settings.

**Example**
If the MPFFILTR parameter is set
MPFFILTR=(AUTO=NO,SUP=NO)
only WTOs that have both AUTO and SUP set to NO are eligible for MainView AutoOPERATOR automation. MainView AutoOPERATOR ignores all other messages that have different AUTO and SUP settings; they are not eligible for MainView AutoOPERATOR automation.

**Example**
If the MPFFILTR parameter is set
MPFFILTR=(AUTO=NO,SUP=YES)
only WTOs that have these AUTO and SUP settings are eligible for automation by MainView AutoOPERATOR. MainView AutoOPERATOR ignores all other messages that have different AUTO and SUP settings; they are not eligible for MainView AutoOPERATOR automation.
Example

If the MPFFILTR parameter is set

MPFFILTR=(AUTO=BOOLE, SUP=NO)

only WTOs with an AUTO setting of BOOLE and a SUP setting of NO are eligible for automation by MainView AutoOPERATOR.

Example

If you want all WTOs that have a MPF token setting of BOOLE and you do not want to check the SUP setting, use

mpffiltr=(AUTO=BOOLE, SUP=*)

All WTOs with an AUTO setting of BOOLE are eligible for automation by MainView AutoOPERATOR, regardless of its SUP setting. So, for example, if a WTO has any of the following settings, they are not eligible for automation:

AUTO=YES, SUP=NO
AUTO=NO, SUP=NO
AUTO=YES, SUP=YES
AUTO=NO, SUP=NO
AUTO=BOOLE, SUP=NO
AUTO=NETVIEW, SUP=YES

Example

To use more than one set of AUTO and SUP settings

MPFFILTR=((AUTO=YES, SUP=YES), (AUTO=YES, SUP=NO))

Now all WTOs with an auto setting of YES are eligible for automation regardless of the setting. You get the same result when you set MPFFILTR to

mpffiltr=(AUTO=YES, SUP=*)

MainView AutoOPERATOR ignores all other WTOs that have different AUTO and SUP settings; they are not eligible for MainView AutoOPERATOR automation.
Supporting IMFEXEC CMD TYPE(ZLINUX) and TYPE(VM)

This chapter describes how to implement support for using IMFEXEC CMD TYPE(ZLINUX) and IMFEXEC CMD TYPE(VM) commands.

Prerequisites

You can use the IMFEXEC CMD TYPE(ZLINUX) command to access the Linus Torvalds Linux environment from any MainView AutoOPERATOR BBI-SS PAS that has CAS connectivity to a MainView for Linux - Servers product, and is monitoring a zLinux system.

Similarly you can use IMFEXEC CMD TYPE(VM) to access VM environments that are being monitored by MainView for Linux - Servers.

Install MainView for Linux — Servers

You must install and configure MainView for Linux - Servers version 1.4.00 or later on the target Linux and VM systems.

See the MainView for Linux -- Servers Customization Guide for information about installing and customizing the MainView for Linux -- Servers product.

Configure BBPARM member BBISSP00

You must configure the following BBPARM member BBISSP00 parameter:

   CASID=
Ensure that you specify a CAS name that has direct access (or indirect access through CAS connectivity) to the context specified on the TARGET parameter.

**Security considerations**

IMFEXEC CMD TYPE(ZLINUX) commands are processed through the view FSLEXEC in the MainView *for Linux* — Servers product.

IMFEXEC CMD TYPE(VM) commands are processed through the view VMFEXEC in the MainView *for Linux* -- Servers product.

You can specify additional security by using the MainView *for Linux* — Servers security definitions for the FSLEXEC or VMFEXEC view. Review the SERDEF view for the resource entry associated with Execute file. By securing these entries, you can prevent access to the FSLEXEC or VMFEXEC views and therefore, prevent access to zLinux and VM targets being monitored by the MainView *for Linux* -- Servers PAS.

Also, note that this command utilizes the MainView API and requires READ access to the resources TERMINAL(VBAPI) and PROGRAM(BBM0IA10). If you do not allow access to these resources, the IMFEXEC CMD TYPE(ZLINUX) or TYPE(VM) command fails with an IMFCC code of 8.

**Identifying zLinux systems**

To identify all of the Linux and VM systems that are currently connected to MainView *for Linux* — Servers, use the windows mode Plex Manager view, PLEXPROD.

Refer to “Identifying zLinux systems To identify all of the Linux and VM systems that are currently connected to MainView *for Linux* — Servers, use the windows mode Plex Manager view, PLEXPROD.” and “Identifying zLinux systems To identify all of the Linux and VM systems that are currently connected to MainView *for Linux* — Servers, use the windows mode Plex Manager view, PLEXPROD.” for examples of this view.

The PLEXPROD view summarizes the status of local and remote MainView products. The PLEXPROD view lists all of the active and inactive targets, grouped by area, and shows the number of instances of a particular product on any connected system.

The System column shows the Linux or VM system name and the last column (Server) shows the MainView *for Linux* -- Servers server name. After you have identified the systems, you can enter the system name on the TARGET keyword of the IMFEXEC CMD TYPE(ZLINUX) or TYPE(VM) command.
Refer to "CMD TYPE(LINUX) or CMD TYPE(VM)" in the MainView AutoOPERATOR Advanced Automation Guide for complete information about how to use the command.

Figure 28: PLEXPROD view of MainView for Linux with a MVLNX Linux context and targets

![Table showing MainView view with MVLNX context and targets]

Figure 29: PLEXPROD view of MainView for Linux with a VM context and targets

![Table showing MainView view with VM context and targets]
Implementing support for sysplex variables

The following sections describe how to implement support for sysplex variables.

What sysplex variables are

Sysplex variables are an optional feature that you can use to store changing values that need to be synchronized among all MainView AutoOPERATOR BBI-SS PASs within a sysplex.

For more information about sysplex variables, refer to the chapter "Using variables in REXX EXECs" in the MainView AutoOPERATOR Advanced Automation Guide.

Enabling support for sysplex variables

The following sections describe the options in BBPARM member AAOPRM xx that allow you to enable support for sysplex variables.

You can manually specify these values by editing BBPARM member AAOPRMxx or you can enter them in the Dynamic Parameter Manager application that allows you to edit and activate certain BBPARM members. Refer to the “Continuous operation with the Dynamic Parameter Manager” on page 69 for more information about how to use the Dynamic Parameter Manager application.

Note

Sysplex variables are not supported in BBPARM member BBIVAR00. Do not specify sysplex variables in BBPARM member BBIVAR00 or unpredictable results might occur.
Activating support for sysplex variables

To enable support for sysplex variables in MainView AutoOPERATOR, in BBPARM member AAOPRMxx, specify the following parameter and then warm start MainView AutoOPERATOR:

SPLXVARS=Y

The default specification is SPLXVARS=N, which means support for sysplex variables is not enabled.

**WARNING**

Changing an active MainView AutoOPERATOR BBI-SS PAS from SPLXVARS=Y to SPLXVARS=N causes any existing sysplex variables to become ordinary SHARED variables, which operate independently in each MainView AutoOPERATOR. You can clear the whole SHARED pool by starting the MainView AutoOPERATOR BBI-SS PAS with START=COLD, VPOOL=RESET however, you should be aware that there may be unpredictable results from taking this action. For example, the CSM application might perform as if the system was IPLed and CSM will start Transient Objects.

Specifying the default sysplex variable prefix (optional)

The default prefix is `AOSPLX_`. BMC recommends that you use the default prefix unless you have a specific need to specify a different prefix value.

1. To specify the prefix that identifies which variables are sysplex variables, specify a value for following parameter in BBPARM member AAOPRMxx:

   SPLXPREF=(1 to 16 user_specified_prefix)

   Valid user-specified prefixes must be alphanumeric and cannot start with a number. Some special characters are permitted, such as dollar sign ($), pound sign (#), and at sign (@).

   BMC also recommends that if you must specify a sysplex prefix value, that you ensure that all the MainView AutoOPERATOR subsystems use the same sysplex prefix, which will make both maintenance and problem diagnosis easier to do.

2. Warm start the MainView AutoOPERATOR BBI-SS PAS for the change to take effect.
Specifying the name of the coupling facility

This procedure specifies the name of the coupling facility.

1. To specify the name of the coupling facility structure where the sysplex variables are stored, specify a value for the following parameter in BBPARM member AAOPRMxx:

   SPLXSTR=coupling_facility_name

   The default value comes from the specified value on the XCFGROUP= parameter in BBPARM member BBISSPxx.

2. Warm start the MainView AutoOPERATOR BBI-SS PAS for the change to take effect.

Table 10 on page 145 shows examples based on different settings for the XCFGROUP and SPLXSTR parameters. Column 2 lists whether AOAnywhere commands can be sent from one BBI-SS PAS to the other BBI-SS PASs and column 3 lists if sysplex variables can be shared. Column 4 shows the resulting coupling facility structure.

Table 10: Examples of how parameter settings XCFGROUP and SPLXSTR affect sysplex variables

<table>
<thead>
<tr>
<th>AAOPRMxx parameter settings for MainView AutoOPERATOR system 1 (MAO1) and MainView AutoOPERATOR system 2 (MAO2)</th>
<th>AOAnywhere communications available?</th>
<th>Sysplex variables are shared</th>
<th>CF structure name</th>
</tr>
</thead>
<tbody>
<tr>
<td>(MAO1) XCFGROUP=default, SPLXSTR=default</td>
<td>Yes</td>
<td>Yes</td>
<td>MVAO_BMCAB</td>
</tr>
<tr>
<td>(MAO2) XCFGROUP=default, SPLXSTR=default</td>
<td></td>
<td></td>
<td>MVAO_BMCAB</td>
</tr>
<tr>
<td>(MAO1) XCFGROUP=xxx, SPLXSTR=default</td>
<td>No</td>
<td>No</td>
<td>MVAO_xxx</td>
</tr>
<tr>
<td>(MAO2) XCFGROUP=yyy, SPLXSTR=default</td>
<td></td>
<td></td>
<td>MVAO_yyy</td>
</tr>
<tr>
<td>(MAO1) XCFGROUP=xxx, SPLXSTR=sss</td>
<td>No</td>
<td>Yes</td>
<td>MVAO_sss</td>
</tr>
<tr>
<td>(MAO2) XCFGROUP=yyy, SPLXSTR=sss</td>
<td></td>
<td></td>
<td>MVAO_sss</td>
</tr>
<tr>
<td>(MAO1) XCFGROUP=default, SPLXSTR=aaa</td>
<td>Yes</td>
<td>No</td>
<td>MVAO_aaa</td>
</tr>
<tr>
<td>(MAO2) XCFGROUP=default, SPLXSTR=bbb</td>
<td></td>
<td></td>
<td>MVAO_bbb</td>
</tr>
</tbody>
</table>
Additional considerations

MainView AutoOPERATOR does not provide support for a BBPARM member that allows you to pre-initialize sysplex variables because the design assumes that sysplex variables contain dynamic values that cannot be unconditionally set.

For example, all of the MainView AutoOPERATOR subsystems could run the same startup EXEC or Rule that tests the environment and conditionally use the VPUT function to specify sysplex variables with different values.

You should carefully examine your automation needs and environment to first determine if your automation even requires the use of sysplex variables. As previously noted, static values or values used by only one MainView AutoOPERATOR subsystem should be stored in SHARED, PROFILE, or TSO variables.

If you decide to enable support for sysplex variables, and you have three MainView AutoOPERATOR systems (AO1, AO2, and AO3) in the sysplex, you must ensure that automation works correctly no matter which MainView AutoOPERATOR BBI-SS PAS starts first or if they all start at the same time.

To initialize your sysplex variables correctly, do not design automation where one BBI-SS PAS must start before subsequent BBI-SS PASs.

You should design and test any automation that uses sysplex variables so that any of the MainView AutoOPERATOR BBI-SS PAS can be started first.

Also test that all of the systems, (in this case, AO1, AO2, and AO3), can be started simultaneously and the sysplex variables have the correct values.

You can use EXECs with the IMFEXEC VENQ SCOPE(Global) command to protect or serialize operations across MainView AutoOPERATOR BBI-SS PASs in the sysplex.

Using the coupling facility

The following sections describe the coupling facility and how to configure and manage it to enable and sustain MainView AutoOPERATOR sysplex variables.
Coupling facility hardware requirements

The use of sysplex variables in MainView AutoOPERATOR requires access to a coupling facility. The coupling facility must be available to all LPARs in the sysplex where MainView AutoOPERATOR runs.

For high availability, through both planned and unplanned outages, you can define a duplex structure where the primary and secondary copies of the structure reside in different coupling facilities on separate machines. This way, an outage of one coupling facility (or the machine that it runs on) does not affect the availability of sysplex variables.

Rebuilding the coupling facility

The four types of Rebuilds, what they are, and why you would want to do them, are described in the MVS manual, *Sysplex Services Guide*.

The following list describes additional limitations:

- User-Managed Rebuild is not supported by MainView AutoOPERATOR.
- User-Managed Duplexing Rebuild is not supported by MainView AutoOPERATOR.
- System Managed Rebuild is supported by MainView AutoOPERATOR, however, sysplex variables will not be available during the time when the structure is not functional during the rebuild.
- System-Managed Duplexing Rebuild is supported by MainView AutoOPERATOR and offers the highest level of availability.
- MVS supports duplexing a structure as one method of improving reliability and availability.

If the structure is unavailable, for example due to some failure or during a Rebuild, then sysplex variable operations will fail during this outage period (for example, reads of a sysplex variable will behave as though the variable did not exist).

You can define a duplex structure where copies of the structure reside in different coupling facilities as a means of providing high availability through both planned and unplanned outages.

For example, you have three physical z9 machines where the MainView AutoOPERATOR BBI-SS PASs and your applications run on machine1 while the
coupling facility with the primary copy of your structure resides on machine2 and
the secondary copy of your structure resides on machine3.
With this duplex structure arrangement, the loss of one coupling facility, whether
planned or unplanned, should not disrupt sysplex variables operation.

**Coupling facility structure sizing**

The maximum number of sysplex variables for your structure size can be seen by
issuing the .D V command in the BBI Journal or the following MVS command to
display the structure:

```
D XCF,STR,STRNM=mvao_bmcab
```

For example, a 2 megabyte (2,000 KB) structure on a 2817 (z196) machine can contain
approximately 1,100 sysplex variables. A 5,000 KB structure can contain
approximately 4,600 sysplex variables. It is difficult to estimate the number of long
variables that can be stored because long variables are internally implemented as
invisible chains of short variables.

The amount of overhead storage consumed in a structure is related to the CFLEVEL.
Higher CF Levels (versions) with increased features tend to have more overhead and
will therefore support a fewer number of sysplex variables for the same size
structure.

IBM typically provides notice to customers when new CF levels incur more
overhead, alerting customers to evaluate if existing structures need to be made
larger.

You might choose to allocate a smaller or larger structure to suit their needs. The
optimum-sized structure is one that supports the maximum number of sysplex
variables needed with a safety margin for unexpected variables or an unexpected
increase in coupling facility overhead.

When planning your number of sysplex variables, you should note that sysplex
variables are slower and consume more resources than SHARED variables. Variables
that are referenced in only one MainView AutoOPERATOR subsystem, or those that
have static values, are best stored as SHARED variables.

You may need to slightly increase the size of the coupling facility structure when
migrating to a new machine or upgrading the CFLEVEL on the same machine, if the
amount of overhead storage consumed by the coupling facility is more than before.
WARNING
Do not create overly large structures or structures that are less than 60% utilized. For example, do not create a 20,000 KB structure or allocate a structure for 20,000 sysplex variables when you will need only a maximum of 200 sysplex variables. Such large or under utilized structures will consume significant amounts of CPU and storage. Based on the size of the structure, the internal code in the coupling facility will allocate an equal number of control blocks, and also search or scan that same number of blocks even though very few sysplex variables have been created.

Coupling facility structure definition

Your MVS system programmer needs to define the coupling facility structure to use sysplex variables.

If both the BBPARM member BBISSPxx XCFGRP parameter and the AAOPRMxx member SPLXSTR parameter use the defaults, the default structure name is as follows, where BMCAB is the default value for the XCFGROUP and MVAO_ is a hard-coded prefix:

MVAO_BMCAB

Run the appropriate IBM utility to define the coupling facility structure. For example, this utility might be the IXCMIAPU program that is also known as the Administrative Data Utility.

An example input statement follows for the utility to create a 1 Megabyte (1,000 KB) structure.

STRUCTURE NAME(MVAO_BMCAB)

SIZE(1000)

If XCFGROUP=YOURCOMP and SPLXSTR=MYCOMP is specified, the effective structure name is BMCAB_MYCOMP. Refer to Specifying the name of the coupling facility on page 145 for more examples.

Modifying the coupling facility structure

To change the size of the structure or modify its properties (for example, to enable duplexing), you must ensure all BBI-SS PASs are down before manually deleting and redefining the structure.
1 Stop all the BBI-SS PASs connected to the structure.

2 To verify that all connections to the structure have been stopped, issue the MVS command:

   D XCF STR,STRNM= structure name

   The default structure name is MVAO_BMCAB.

3 To delete the structure, issue the MVS command:

   SETXCF FORCE,STR,STRNM= structure name

   At this point the structure is deleted and all sysplex variables are lost.

4 Run your job to redefine the structure with the desired properties.

5 Issue the MVS command to display the structure and verify that it has the new desired properties.

6 Re-start MainView AutoOPERATOR to resume automation.

### Coupling facility structure connections

Each MainView AutoOPERATOR subsystem creates one connection to the coupling facility structure.

The maximum number of connections allowed for a structure is controlled by a CFRM parameter and this parameter directly controls the maximum number of MainView AutoOPERATOR subsystems that can connect or share a structure.

Use the Format utility for couple data sets to change this parameter. An example of the statement follows:

   ITEM NAME(CONNECT) NUMBER(nnn)

For z/OS version 1.7 and later, the default for nnn is 32, the minimum is 1 and maximum is 255.

Refer to the IBM manual, *z/OS V1R7.0 MVS Setting Up a Sysplex*, appendix C, "CFRM Parameters for Administrative Data Utility" for more information.

If you set the maximum number of connections to 255, as many as 255 MainView AutoOPERATOR subsystems can connect to the structure and share sysplex variables.
To display the current maximum number of connections allowed to a structure, issue the following MVS command:

\texttt{D XCF,COUPLE,TYPE=CFRM}

### Coupling facility structure persistence

The coupling facility is a volatile entity and only exists in memory.

MainView AutoOPERATOR allocates the structure so that it persists and retains sysplex variables even when all MainView AutoOPERATOR subsystems have shut down.

When the coupling facility is reset or re-IPLed, all structures that reside in the coupling facility are destroyed and subsequently all sysplex variables are lost.

Individual structures might also be manually destroyed or deleted by using an MVS command such as:

\texttt{SETxCF FORCE,STRUCTURE,STRNAME=(structure name)}

Do not destroy any structure while it is in use by MainView AutoOPERATOR.

---

**Note**

MVS supports duplex structures, with each structure in a separate coupling facility, as a means to improve reliability and availability.
Using the coupling facility
Defining OSPI virtual terminals

This chapter describes defining Open Systems Procedural Interface (OSPI) virtual terminals.

To use the OSPI interface, you must be running VTAM 3.0 or later.

Overview

MainView AutoOPERATOR provides OSPI the as an interface to VTAM-based products. OSPI provides a means for REXX- or CLIST-based automation procedures to interface with any LU2 VTAM application that uses full screens to communicate with users.

With OSPI, MainView AutoOPERATOR has logon capabilities and complete access to any VTAM application’s data stream. In this way, MainView AutoOPERATOR can interact with the application by analyzing the output data and issuing the VTAM application’s own commands.

By automatically interfacing with critical VTAM applications and simulating a user at a VTAM terminal, OSPI can communicate with various data center software products and decrease the number of VTAM-based terminals.

This terminal emulation requires that some number of OSPI virtual terminals be defined to VTAM. In addition, some applications, such as CICS and IMS, might require local definitions for OSPI terminals. Refer to “OSPI terminal definitions for CICS and IMS” on page 343 for more information.

Using the OSPI ACB pool

When OSPI attempts to establish a session with a VTAM application, a VTAM ACB is required to represent the OSPI virtual terminal. If a specific ACB name is not requested, the OSPI ACB pool is searched for an available and usable ACB.
See the chapters that describe the OSPI Scripting facility and the IMFEXEC LOGON command in the *MainView AutoOPERATOR Advanced Automation Guide* for information about requesting a specific ACB.

### Naming conventions

The pool consists of a group of ACBs named using the following conventions:

\[
PPPPnnnn
\]

where

<table>
<thead>
<tr>
<th>PPPP</th>
<th>is the 4-character pool prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnn</td>
<td>is a sequential number from 0000 to 9999</td>
</tr>
</tbody>
</table>

The default pool prefix is OSPI. You can specify a different prefix by using the OSPIPRFX parameter in BBPARM member AAOEXP00.

**Note**

The prefix *must* be four characters.

OSPI attempts to locate the first usable ACB beginning with OSPI0000. The search continues sequentially until a usable ACB is found or all OSPI ACBs have been exhausted.

### Defining the ACB pool

A VTAM APPL statement is required for each ACB that you want to define in the OSPI ACB pool. You can add new APPL statements to an existing member or create a new member in SYS1.VTAMLST. The statements should be similar to the following example:

```verbatim
OSPI VBUILD TYPE=APPL
OSPI0000 APPL AUTH=(ACQ,NOTSO,VPACE)
OSPI0001 APPL AUTH=(ACQ,NOTSO,VPACE)
```

The number of ACBs defined determines the maximum number of concurrent OSPI sessions MainView AutoOPERATOR can support unless ACBs outside the OSPI pool are also used.

ACBs must be activated before they can be used. They may be activated automatically at VTAM start-up or by issuing a VTAM VARY command.
If LOGMODE entries other than the entries defined in the default MODETAB will be used by OSPI sessions, add the MODETAB= keyword to each APPL statement.

### Defining the ACB pool to MainView AutoOPERATOR

When OSPI attempts to locate a usable ACB in the pool, the search begins with OSPI0000 and continues sequentially until a usable ACB is found or all OSPI ACBs have been exhausted.

If the number of ACBs in the OSPI pool has not been defined to MainView AutoOPERATOR, OSPI assumes that the highest numbered (last) ACB has been used when an open is unsuccessful against an ACB that is not in use. The search will not continue even if higher numbered ACBs might be usable.

BMC Software recommends that you define the number of ACBs in the pool using the OSPINUM parameter in BBPARM member AAOEXP00. This causes a pool search to continue until all OSPI ACBs have been examined. The number specified using the OSPINUM parameter should match the number of ACBs defined to VTAM.

### OSPI application definitions

Some applications, such as CICS and IMS, require local definitions for OSPI terminals.

Refer to “OSPI terminal definitions for CICS and IMS” on page 343 for information about how to define virtual terminals to CICS and IMS.

When defining OSPI virtual terminals to applications other than CICS and IMS, BMC Software recommends that the terminals be defined as 3278 model 2.
Considerations for OMVS and \textit{TCP/IP}

The MainView AutoOPERATOR BBI-SS PAS might use TCP/IP services. If the BBI-SS PAS uses TCP/IP services, the BBI-SS PAS must have access to an OMVS segment with superuser authority.

Some examples of processing within the MainView AutoOPERATOR BBI-SS PAS that might use TCP/IP services include the General Message Exchange (GME), TCP-based solutions (such as EXECs QAOMTP2, QAOSNPP2, QAOALMP2, QAOALMP5, and QAOSNMP2), or any user written automation that uses the commands NETSTAT or PING. When TCP/IP services are in use a basic UNIX System Services segment is required. In addition, it is required that TCP/IP is up and running before the MainView AutoOPERATOR BBI-SS PAS requires TCP/IP services.

Refer to the chapter "Managing cross system communication" in the \textit{MainView Administration Guide} for information about how to define a TCP/IP connections and additional information about OMVS.
Implementing BBI-SS PAS to CICS communication

This chapter describes the steps required to implement BBI-SS PAS to CICS communication.

Overview

For sites implementing the MainView for CICS product at the same time as MainView AutoOPERATOR for CICS, use the MainView for CICS Customization Guide and perform the steps described in "Standard Implementation Procedures."

This chapter discusses implementation procedures if you are installing MainView AutoOPERATOR for CICS only.

Certain MainView AutoOPERATOR for CICS functions require BBI-SS PAS to CICS communication. However, numerous MainView AutoOPERATOR for CICS functions are available to you if you do not implement BBI-SS PAS-to-CICS communication. See “MainView AutoOPERATOR for CICS functions (with BBI-SS PAS-to-CICS communication)” on page 171 for additional information.

---

**Note**

A CICS target can only connect to, and be automated by, one MainView AutoOPERATOR BBI-SS PAS at a time.

The following steps are required to implement BBI-SS PAS to CICS communication:

- Review implementation considerations (“Implementation considerations” on page 160).
- Specify resources to CICS (“Specifying resources to CICS" on page 161).
- Modify CICS startup JCL (“Modifying CICS startup JCL” on page 170).
- Restart CICS target regions (“Restarting CICS target regions” on page 170).
To begin implementing BBI-SS PAS-to-CICS communication, go to “Implementation considerations” on page 160.

**Note**

If you are running MainView AutoOPERATOR for CICS and in BBPARM member BBISSP00 you specified ACONCICS=ONLY or ACONCICS=YES with no static or dynamic target definition, all CICS commands that can be used in MainView AutoOPERATOR Rules and EXECs are dependent, which means they will not be available until the CICS region target has established communications with the BBI-SS PAS.

Refer to the *MainView for CICS Customization Guide* for more important information about the ACONCICS parameter in BBPARM member BBISSP00.

---

**Implementation considerations**

This section provides information you should consider when you are implementing MainView for CICS or MAINVIIEW AutoOPERATOR for CICS or both products together.

**Site-specific security**

Transactions FST2, JNL2, FIC2, FCM1, and FCD2 run asynchronously. The TRANSEC parameter of the transaction definition must have a value of 1 when CICS security is used. If your site uses an external security package, such as IBM RACF or Computer Technologies CA-ACF2, that package must be updated to let these transactions run asynchronously.

Transaction SMN2 can be RACF-secured.

**Temporary Storage Table (TST) compatibility**

To avoid conflicts between the BMC Software temporary storage prefix and any existing recoverable temporary storage definitions, perform the following steps for each CICS that will communicate with a BBI-SS PAS:

1. Determine if a Temporary Storage Table (TST) is now in use.

   If a TST is not being used, there is no conflict.
2 For an existing TST, determine if the DATAID parameter for any of the DFHTST TYPE=RECOVERY or TYPE=REMOTE macros specifies a value that conflicts with the following string:

CMRI

If there is a conflict, see “Changing the use of CICS recoverable temporary storage” on page 271.

Specifying resources to CICS

The following sections describe the steps necessary to make the MainView AutoOPERATOR transactions and programs available to CICS.

Making MainView AutoOPERATOR transactions and programs available to CICS and CTS

Use the auto maintenance facility to install and apply maintenance for all supported releases of CICS and CTS.

For more information, see the section “Using auto maintenance” on page 163. Also, be sure to review the section “Update the CSD” on page 166 for information about applying CSD updates.

Modifying the Program List Table (PLT)

The PLT defines programs to be executed in the post-initialization phase of CICS startup.

Completing the steps in this section automates the initiation of BBI-SS PAS-to-CICS communication at CICS startup.

Use of the PLT is optional. However, if you do not use the PLT, manually start BBI-SS PAS-to-CICS communication using either the SMN2 transaction from within the CICS region or the FST2 transaction from the z/OS console.

Refer to “Manual activation of BBI-SS to CICS communication” on page 279 for more information.
Note
To assemble the table changes, concatenate BBSAMP with SYSLIB in your site's CICS table assembly JCL (normally, DFHAUPLE).

To edit an existing PLT

1. Edit the PLT and find the following statement at the end of the table:

   \texttt{DFHPLT TYPE=FINAL}

2. Just before the \texttt{DFHPLT TYPE=FINAL} statement, insert the following macro invocation:

   \texttt{CMRPLT MODE=AUTO}

   The parameters for the CMRPLT macro are described in the following table.

\begin{table}[ht]
\centering
\begin{tabular}{|l|l|l|}
\hline
\textbf{Term} & \textbf{Parameters} & \textbf{Description} \\
\hline
MODE & AUTO & Required parameter to use auto maintenance, which is also required. \\
SUFFIX & BI & Specifies the suffix for the new PLT. Specify a suffix that adheres to the naming conventions at your site. BI is the default. If you do not use the default, create an entry for DFHPLTxx (where xx is the suffix) in the PPT table of the CICS system that uses the PLT.
\multicolumn{2}{|l|}{When SUFFIX is used, TYPE=INITIAL and TYPE=FINAL statements are generated automatically.} \\
\hline
\end{tabular}
\end{table}

3. Assemble the newly revised PLT.

   The PLT is modified.

To create a new PLT

1. Create a new PLT in your CICS table library using a member name that adheres to the naming conventions at your site. The new member should contain the following macro invocation:

   \texttt{CMRPLT MODE=AUTO}

   The parameters for this macro are described in Table 11 on page 162.
2. Assemble the newly created PLT.

3. Add the following parameter to the CICS initialization override parameters:

   \[ \text{PLTPI}=xx \]

   where \( xx \) is the suffix specified in Step 1 above.

   The PLT is created and enabled.

   Go to “Modifying CICS startup JCL” on page 170.

### Using auto maintenance

The MainView AutoOPERATOR for CICS auto maintenance facility enables you to apply upgrades to the product software without restarting target CICS systems.

The auto maintenance facility provides INIT and TERM functions that completely remove the MainView agents from a running CICS system and re-initialize the MainView agents without affecting the CICS system itself.

The TERM function terminates all active MainView AutoOPERATOR functions and removes the MainView AutoOPERATOR for CICS resources in CICS, with the exception of those resources necessary to redefine the discarded resources. The TERM function stores the MainView processing environment in a status record and writes it to a temporary storage queue.

### TERM requests

For TERM requests, MainView AutoOPERATOR for CICS

The Transient Data Exit, CMRTDPXn is also released from storage. TERM requests can come from the following sources:

1. discards all created resources
2. stops the Transient Data Exit
3. closes the BBICOM01 DD subsystem data set
4. frees the BBICOM01 DD
The Transient Data Exit, CMRTDPXn is also released from storage. TERM requests can come from the following sources:

- the MainView AutoOPERATOR PAS terminates
- the MVCICS CRegion View commands RES, TRM or AOF
- the invocation of transaction FST2 TERM
- the invocation of transaction FST2 SYSM OFF

**INIT requests**

The INIT function recreates all the MainView AutoOPERATOR for CICS resources and restarts all their functions according to the status record. The auto maintenance facility replaces the code within a running CICS system. The function can be initiated after you apply the necessary upgrade and maintenance jobs.

For INIT requests MainView AutoOPERATOR for CICS creates all the necessary resources with the exception of program OLTVCAO. This program is defined in the CICS CSD and is the root program for MainView AutoOPERATOR for CICS. OLTVCAO determines if it is running in a supported CICS release and allows or disallows the continuation of MainView AutoOPERATOR for CICS initialization.

INIT requests can come from the following sources

- the CICS system starts and MainView AutoOPERATOR for CICS is defined in the PLT and the MainView AutoOPERATOR PAS is already running
- CICS is already running and the MainView AutoOPERATOR PAS starts
- the MVCICS CRegion View commands RES, INI or AON
- the invocation of transaction FST2 INIT
- the invocation of transaction FST2 SYSM ON

**ON requests**

For ON requests that are generated by the FST2 SYSM ON command or when MainView AutoOPERATOR for CICS initialization starts with auto maintenance not active, MainView AutoOPERATOR for CICS creates the program resource OLTCAORL to check for CICS release compatibility. If auto maintenance finds that any of the required resources are not available, OLTSCAO dynamically creates them.
**OFF requests**

For OFF requests that are generated by the FST2 SYSM OFF command, the CMRTDPXn exit program is disabled but BBICOM01 is not closed or freed. Messages are no longer be processed for automation.

Implementing auto maintenance requires modifications to the CICS CSD, CICS PLT and SIT, and to the BBI-SS JCL. These procedures are described in this chapter.

**Before you begin: auto maintenance checklist**

Before initiating auto maintenance, verify that

- the appropriate load libraries are updated with the modules for the PTF(s) and the data sets do not expand into additional extents
- all prerequisite PTFs can be applied by using the auto maintenance facility

In some cases, a cold or warm restart of the BBI-SS PAS may be required.

**Initiating auto maintenance**

Auto maintenance can be started in the following ways

- an action command from the CREGAGT view
  
  The RES command is used when updating MainView for CICS modules that run in the CICS address space only. The RES command initiates the TERM and the INIT functions for the target CICS systems. In addition, the TRM and INT commands can be used individually in place of the RES command.

  **Note**

  This facility is not related specifically to MainView AutoOPERATOR for CICS but is a part of the MainView for CICS product. If you are installing MainView AutoOPERATOR for CICS without also installing MainView for CICS, you will not be able to use this facility.

- a restart of the BBI-SS PAS restart

  When the BBI-SS PAS terminates, it is detected by common functionality in MainView for CICS and by MainView AutoOPERATOR for CICS in the CICS system. When the termination is detected, a TERM function is initiated.

  When the BBI-SS PAS restarts, an INIT request for MainView for CICS and MainView AutoOPERATOR for CICS is sent to the target CICS systems through
the External CICS Interface facility (EXCI). The EXCI facility sends the action commands from the CREGAGT view to the CICS system.

- an FST2 TERM/INIT sequence is issued from the console, a CICS terminal, or from a MainView AutoOPERATOR for CICS EXEC

The EXCI facility enables auto maintenance to initiate the MainView for CICS and MainView AutoOPERATOR for CICS agents without a BBI-SS PAS communications link. It also enables the BBI-SS PAS to initiate the link to a CICS system.

### Auto maintenance messages

MainView AutoOPERATOR for CICS messages are documented in members of the BBMLIB data set and can be viewed by using the MESSAGES General Services option from the MainView AutoOPERATOR for CICS Primary Option Menu.

**Note**

CICS issues DFHXM and DFHPG messages when the CREATE and DISCARD commands are issued. These messages are directed to the CSPL transient data queue by default. MainView AutoOPERATOR for CICS issues error messages to the CICS job log for any CREATE and DISCARD failures.

### Update the CSD

The CICS CSD data set must be updated with the transaction definitions and program resources that are used by MainView AutoOPERATOR for CICS. The transaction definitions become part of the internal CICS Program Control Table (PCT); the program definitions become part of the internal CICS Processing Program Table (PPT).

This procedure modifies or adds resource groups and group lists to the CICS CSD. The resource groups contain the resource definitions for the PCT and PPT which identify MainView AutoOPERATOR for CICS transactions and programs to a CICS system. The group lists identify the resource groups that are installed at CICS initialization.

The CSDs for each CICS region that communicate with a BBI-SS PAS must be modified.

The BBSAMP members CMRCSDLAL, CMRCSDLAL, CMRCSDLB, and CMDCSDEL contain sample JCL with online program and transaction resource definitions. Use the JCL to define the resource groups that required for MainView AutoOPERATOR for CICS. Each BBSAMP member contains documentation for customizing the JCL.
To modify resource groups and lists

1 Read the comments in BBSAMP member CMRCSDAL and modify the JCL accordingly.

CMRCSDAL creates the BMCMVOLT and BMCMVAAO groups. These groups contain the following resources:

<table>
<thead>
<tr>
<th>Group Name</th>
<th>Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCMVOLT</td>
<td>Common resources:</td>
</tr>
<tr>
<td></td>
<td>■ OLTFSET program</td>
</tr>
<tr>
<td></td>
<td>■ FST2 transaction</td>
</tr>
<tr>
<td></td>
<td>■ BMCE transaction</td>
</tr>
<tr>
<td>BMCMVAAO</td>
<td>MainView AutoOPERATOR for CICS resources</td>
</tr>
<tr>
<td></td>
<td>■ OLTVCAO program</td>
</tr>
</tbody>
</table>

The CMRCSDAL job submits the CSDcccDF members based on your modifications, where ccc is OLT or AAO.

For more information about the members in the BBSAMP data set, refer to “Sample data set members” on page 335.

2 Submit the JCL.

3 Check the output for a return no higher than 4.

4 Modify and run the BBSAMP member CMRCSDLA.

The CMRCSDAL member referred to in Step 1 on page 167 explains which member to use if you are installing MainView for CICS along with MainView AutoOPERATOR for CICS. In this case, refer to the MainView for CICS Customization Guide for installation information. These members modify the group list that is used at CICS initialization that contains the BMC Software resource groups.

5 Remove members from groups BOOLEPCT and BOOLEPPT.

Review the instructions in BBSAMP member CMRCSDEL to modify and run the CMRCSDEL job. The CMRCSDEL job deletes obsolete resource definitions from the CICS CSD by removing resource definitions from the existing BOOLEPPT and BOOLEPCT groups.
This job also inserts the correct CSDcccrr members based on the specifications that you enter where

- **ccc**: corresponds to parameter PRODUCT setting in BBSAMP member CMRCSDEL
- **rr**: corresponds to the CICS version and release numbers that are defined in BBSAMP member CMRCSDEL

6 Implement EXCI connection and session definitions

Program, transaction, connection, and session definitions are required to use the EXCI function. Ensure that all CICS systems have the appropriate EXCI connection and session definitions installed and in service. MainView for CICS provides the necessary program and transaction definitions with BMCE (transaction) and OLTFSET (program).

You must create the connection and session definitions yourself. The connection definitions must specify the EXCI protocol, IRC access method, and CONNTYPE(GENERIC).

Refer to the IBM document *CICS External Interfaces Guide* for more information about the EXCI interface.

7 Modify the Program List Table (PLT)

The PLT specifies which programs are executed in the post-initialization phase of CICS startup. This procedure modifies the PLT to initiate BBI-SS PAS to CICS communication at CICS startup. Using the PLT to modify MainView AutoOPERATOR for CICS is optional. However, if you do not use the PLT, you must manually start BBI-SS PAS to CICS communication by using the FST2 transaction from the MVS console.

There are two different procedures in this section:

- “To modify an existing PLT” on page 168
- “To create a new PLT” on page 169

You must complete one of these procedures based upon whether you have an existing PLT or are creating a new PLT. Either procedure automates the initiation of BBI-SS PAS to CICS communication.

**To modify an existing PLT**

Complete this procedure if you have an existing PLT that you want to modify. Modify the PLT by inserting the appropriate program or using the CMRPLT macro. BMC Software recommends that you manually insert the program name.
1. To insert the program name manually, find the following statement:
   ```plaintext
   TYPE=ENTRY,PROGRAM=DFHDELIM
   ```

2. Insert the following program name immediately after the `PROGRAM=`
   `DFHDELIM` statement:
   ```plaintext
   DFHPLT TYPE=ENTRY,PROGRAM=OLTFSET
   ```

3. To use the CMRPLT macro, find the following statement at the end of the table:
   ```plaintext
   DFHPLT TYPE=FINAL
   ```

4. Insert the CMRPLT macro invocation immediately before the `DFHPLT`
   `TYPE=FINAL` statement:
   ```plaintext
   CMRPLT MODE=AUTO
   ```

5. Assemble the newly revised PLT.

   You must include the BBSAMP data set in the assembler SYSLIB concatenation if
   you use the CMRPLT macro.

6. Add the PLTPI parameter to the SIT to identify this updated PLT to CICS:
   ```plaintext
   PLTPI=xx
   ```
   where `xx` is the suffix of this PLT.

   When you have completed the above steps, you have modified the PLT for
   automatic start up of BBI-SS PAS to CICS communication.

**To create a new PLT**

Complete this procedure if you need to create a new PLT. Create a new PLT in your
CICS table library using a member name that matches the naming conventions used
at your site. The new member should contain either a CMRPLT macro invocation or
the appropriate program name:

1. To use the program name, add this statement after the `PROGRAM=DFHDELIM`
   statement:
   ```plaintext
   DFHPLT TYPE=ENTRY,PROGRAM=OLTFSET
   ```

2. To use the CMRPLT macro, add this macro invocation:
   ```plaintext
   CMRPLT MODE=AUTO
   ```

3. Assemble the newly revised PLT.

   You must include the BBSAMP data set in the assembler SYSLIB concatenation if
   you use the CMRPLT macro.
4 Add the PLTPI parameter to the SIT to identify this updated PLT to CICS:

```
PLTPI=xx
```

where xx is the suffix of this PLT.

When you have completed the above steps, you have modified the PLT for automatic startup of BBI-SS PAS to CICS communication.

### Modifying CICS startup JCL

Do the following steps for each CICS that will communicate with a BBI-SS PAS:

1. Add the following DD statement to the CICS execution step:

```
//BBIPARM DD DSN=prefix.UBBPARM,DISP=SHR
```

2. Add the following DD statement to the DFHRPL library concatenation in the CICS execution step:

```
//   DD DSN=prefix.BBLINK,DISP=SHR
```

The CICS startup JCL is modified.

### Restarting CICS target regions

To complete the implementation of BBI-SS PAS to CICS communication, do the following steps:

1. Start each BBI-SS PAS that will communicate with a CICS region (if it is not already active).

2. Shut down each CICS region that will communicate with a BBI-SS PAS that is already active.

3. Start each CICS region that will communicate with a BBI-SS PAS.

4. For each CICS region that communicates with a BBI-SS PAS but for which the procedure in “Modifying the Program List Table (PLT)” on page 161 was not performed, manually start BBI-SS PAS-to-CICS communication; use either the SMN2 transaction from within the CICS region or the FST2 transaction from the z/OS console.

Refer to “Manual activation of BBI-SS to CICS communication” on page 279 for more information.
Ensure that the following messages are issued from each CICS region that will communicate with a BBI-SS PAS:

+FT431I SESSION WITH BBI-SS PAS ESTABLISHED  
+FT428I CICS MESSAGE INTERCEPT COMPLETED

The following messages are issued from each CICS region that communicates with a BBI-SS PAS:

+FT431I SESSION WITH BBI-SS SSID ESTABLISHED  
+FT806I AutoOPERATOR for CICS initialization started

+FT803I CICS TRANSIENT DATA INTERCEPT STARTED  
+FT428I CICS MESSAGE INTERCEPT START COMPLETED  
+FT809I AutoOPERATOR for CICS initialization complete

The implementation of BBI-SS PAS to CICS communication is now complete.

**MainView AutoOPERATOR for CICS functions (with BBI-SS PAS-to-CICS communication)**

If you choose to implement BBI-SS PAS-to-CICS communication, you will be able to

- use Rules to invoke CICS-dependent commands without using an EXEC
- use the Rule Processor to capture messages from the CICS transient data queue
- use hyperlinks to invoke MainView for CICS applications
- use the BROADCAST application from the CICS Operator Workstation in the BBI-TS
- use the CICS-dependent IMFEXEC CICS commands from an EXEC

The CICS-dependent IMFEXEC CICS commands are as follows:

<table>
<thead>
<tr>
<th>IMFEXEC CICS command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACQUIRE TERMINAL</td>
<td>Acquire a VTAM-supported terminal.</td>
</tr>
<tr>
<td>ALLOC</td>
<td>Allocate a data set to a CICS region.</td>
</tr>
<tr>
<td>CEMT</td>
<td>Issue a CICS master terminal command.</td>
</tr>
<tr>
<td>CHAP</td>
<td>Change a task’s priority in a CICS region.</td>
</tr>
<tr>
<td>CLOSE</td>
<td>Close a file in a CICS region.</td>
</tr>
<tr>
<td>DISABLE</td>
<td>Disable a CICS resource.</td>
</tr>
</tbody>
</table>
MainView AutoOPERATOR for CICS functions (without BBI-SS PAS-to-CICS communication)

If you choose not to implement BBI-SS PAS-to-CICS communication, you can still use MainView AutoOPERATOR for CICS to

- use the STATUS application from the CICS Operator Workstation in the BBI-TS
- use Rules to invoke CICS-independent commands without needing an EXEC
- use the CICS-independent IMFEXEC CICS commands from an EXEC

The CICS-independent IMFEXEC commands are as follows:

<table>
<thead>
<tr>
<th>IMFEXEC CICS command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLOC (LOCAL only)</td>
<td>Allocate a data set to the BBI-SS.</td>
</tr>
<tr>
<td>ALTER</td>
<td>Change CICS task-related throttles.</td>
</tr>
<tr>
<td>IMFEXEC CICS command</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>ALTERVS</td>
<td>Change virtual storage in the CICS region.</td>
</tr>
<tr>
<td>DISABLE(TRAN, DEST, PROGRAM)</td>
<td>Disable a CICS resource.</td>
</tr>
<tr>
<td>ENABLE (TRAN, DEST, PROGRAM)</td>
<td>Make a CICS resource available for use.</td>
</tr>
<tr>
<td>FREE (LOCAL only)</td>
<td>Deallocate a data set from the BBI-SS.</td>
</tr>
<tr>
<td>KILL TASK (KILL, WITH DUMP, FORCE)</td>
<td>Terminate a task in a CICS region by task number.</td>
</tr>
<tr>
<td>QUERY</td>
<td>Invoke a MainView for CICS service.</td>
</tr>
<tr>
<td>SPURGE</td>
<td>Change the spurge value for CICS transactions.</td>
</tr>
</tbody>
</table>
MainView AutoOPERATOR for CICS optional tailoring

This section describes optional steps for tailoring MainView AutoOPERATOR for CICS to your site’s requirements.

Overview

All steps described in this section are optional, and the full functionality of MainView AutoOPERATOR for CICS is available whether or not they are performed.

The MainView AutoOPERATOR for CICS optional tailoring steps are as follows:

- “Customizing terminal types eligible for broadcast messages (CAOTTAB)” on page 175
- “Limiting CICS transient data queues eligible for Rule processing (CAODTAB)” on page 177

If customized to do so, MainView AutoOPERATOR for CICS can intercept any data written to the Intrapartition Transient Destinations as well.

Customizing terminal types eligible for broadcast messages (CAOTTAB)

When the MainView AutoOPERATOR for CICS BROADCAST application is invoked, a list of all terminals eligible for the BROADCAST function is displayed. The following criteria must be met before a terminal is eligible for broadcasting:

- It must have a CICS status of INSERVICE.
- If it is a VTAM terminal, it must be currently acquired.

- Its model specification must match the specifications of the supplied terminal table.

Because terminal models can differ from installation to installation, you may need to customize the MainView AutoOPERATOR for CICS terminal table so that it includes the terminal types used at your site.

The CAOTTAB macro is used to generate the MainView AutoOPERATOR for CICS terminal table. The default table is distributed by BMC Software using the following CAOTTAB macro invocation:

```
CAOTTAB (L3277,R3277)
```

Any terminal type that can be specified for the CICS terminal definition operand TERMMODL can be specified for CAOTTAB.

The member TTABJCL in `prefix.BBSAMP` contains sample JCL that can be used to assemble and link CAOTTAB as follows:

**To customize terminal types eligible for broadcast messages (CAOTTAB)**

1. Copy TTABJCL from `prefix.BBSAMP` to `prefix.UBBSAMP`. Make sure the new member name conforms to site naming conventions.

2. Edit the member created in Step 1. Read the comments in the member and then modify the JCL accordingly.

3. Modify the sample invocation of the CAOTTAB macro so that all the terminal types required are eligible for BROADCAST messages.

4. Submit the newly created JCL; check the output to insure that all steps complete with a return code of 0.

**Note**

S106 abends can occur if the newly link-edited module causes the BBLINK data set to enter secondary extents and the active BBI-SS PAS attempts to load the module. If this happens, stop BBI-SS PAS and restart it.
Limiting CICS transient data queues eligible for Rule processing (CAODTAB)

By default, MainView AutoOPERATOR for CICS intercepts any data written to any CICS extra partition transient data destination.

If customized to do so, MainView AutoOPERATOR for CICS also can intercept any data written to intrapartition destinations. It then runs its Rule Processor definitions to determine if a particular message should be intercepted and acted upon.

The default may be acceptable in a CICS system that does not send large amounts of user data to its transient data destinations. But if there is a large volume of data traffic, performance can be impacted.

If your CICS regions have a large volume of transient data traffic and you want to limit the names of the CICS transient data destinations that are eligible for the Rule Processor message interception, use the CAODTAB macro to generate a CICS transient data name table that can define which queue names are part of the table and that can

- include only selected intrapartition transient data queue destinations
- include only selected extrapartition transient data queue destinations
- include both selected extrapartition and intrapartition transient data queue destinations
- exclude only selected intrapartition transient data queue destinations
- exclude only selected extrapartition transient data queue destinations
- exclude both selected extrapartition and intrapartition transient data queue destinations

Refer to “Using the CAODTAB macro” on page 177 for more information about how the CAODTAB macro works.

Any CICS extrapartition or intrapartition transient data destination name defined at your site can be specified for CAODTAB.

Using the CAODTAB macro

The member DTABJCL in prefix.BBSAMP contains sample JCL which you can use to assemble and link CAODTAB.
1 Copy DTABJCL from `prefix.BBSAMP` to `prefix.UBBSAMP`. Make sure the new member name conforms to site naming conventions.

2 Edit the member created in Step 1. Read the comments in the member and then modify the JCL accordingly.

3 Modify the sample invocation of the CAODTAB macro so that it uses applicable parameters from “Describing CAODTAB macro parameters” on page 178.

4 Submit the newly created JCL; check the output to ensure that all steps are completed with a return code of 0.

**Describing CAODTAB macro parameters**

The following list describes input parameters for the CAODTAB macro:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Possible values</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXTRA</td>
<td>(YES</td>
</tr>
<tr>
<td>INTRA</td>
<td>(YES</td>
</tr>
<tr>
<td>QUEUES</td>
<td>[(name1),(name2),(namen)...]</td>
</tr>
</tbody>
</table>

The QUEUES= parameter is optional. Use the QUEUES= parameter when you want to explicitly include or exclude specific queues by queue name.

**Examples**

The following examples show how to use different combinations of the parameters on the CAODTAB macro invocation.

**Example**

To specify that only extrapartition transient data messages will be eligible for processing by the Rule Processor:

```
EXTRA=(YES,INCLUDE),INTRA=(NO),QUEUES=(QUE1,QUE2,QUE3)
```

The CICS transient data name table will be an inclusion table for extrapartition transient data messages from queues named QUE1, QUE2, and QUE3. If no queue names are specified (for example, if QUEUES=( ) or if the queues parameter is not used), all extrapartition transient data messages are included.
**Example**

To specify that only intrapartition transient data messages will be eligible for processing by the Rule Processor:

```plaintext
EXTRA=(NO), INTRA=(YES, INCLUDE), QUEUES=(QUE4, QUE5, QUE6)
```

The CICS transient data name table will be an inclusion table for intrapartition transient data messages from queues named QUE4, QUE5, and QUE6. If no queue names are specified (for example, if QUEUES=() or if the queues parameter is not used), all intrapartition transient data messages are included.

**Example**

To specify that both intrapartition and extrapartition transient data messages will be eligible for processing by the Rule Processor:

```plaintext
EXTRA=(YES, INCLUDE), INTRA=(YES, INCLUDE), QUEUES=(QUE7, QUE8)
```

The CICS transient data name table will be an inclusion table for both extrapartition and intrapartition transient data messages for queues named QUE7 and QUE8. If no queue names are specified (for example, if QUEUES=() or if the queues parameter is not used), all extrapartition and intrapartition transient data messages are included.

**Example**

To specify that no transient data messages from any queue will be eligible for processing by the Rule Processor:

```plaintext
EXTRA=(NO), INTRA=(NO)
```

**Example**

To specify that only intrapartition transient data messages will be eligible for processing by the Rule Processor:

```plaintext
EXTRA=(NO), INTRA=(YES, INCLUDE)
```

The CICS transient data name table will be an inclusion table for all intrapartition transient data messages from all queues.

**Example**

To specify that only intrapartition transient data messages will be eligible for processing by the Rule Processor:

```plaintext
EXTRA=(NO), INTRA=(YES, EXCLUDE)
```

The CICS transient data name table will be an exclusion table for intrapartition transient data messages. In this example, since the QUEUES parameter is not used, no intrapartition queues are explicitly excluded. Thus, all intrapartition queues are included.
Implementing MainView products in IMS

This chapter includes the following sections and describes how to install and implement BMC modules, which are required for

- MainView AutoOPERATOR for IMS
- MainView for IMS (MVIMS) Online

Follow the procedures in this chapter if you are customizing one or both one or both of these products and you have not performed MainView Customization. You can refer to the manual customization steps in this chapter if you need help during MainView Customization.

**Note**

If you have more than one IMS, you might want to allocate a BBPARM data set that is unique to a IMS, as described in “Setting up BBPARM data sets” on page 182 and “Modifying the IMS control region JCL” on page 186.

For information about how to use product libraries, including parameter libraries (BBPARM and UBBPARM) and sample libraries (BBSAMP and UBBSAMP), see “Using MainView Product Libraries” in the MainView Customization Reference or “Using Product Libraries” in the MainView Administration Guide.

Deleting modules from a prior release

You can skip this section if you are installing MainView AutoOPERATOR for IMS for the first time.

If a previous release of MainView for IMS, MainView for DBCTL, or MainView AutoOPERATOR for IMS was installed on the target IMS, use the information in Table 12 on page 182 and follow the instructions below.

If you copied BBLINK members to an authorized STEPLIB data set using ICOPY, you can use the following BBSAMP jobs to delete the old modules:
Table 12: BBSAMP jobs to delete old modules

<table>
<thead>
<tr>
<th>IMS release</th>
<th>MVIMS 4.6 MAO 7.2</th>
<th>MVIMS 5.1 MAO 7.3.xx</th>
<th>MVIMS 5.2 MAO 7.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS 11.1</td>
<td>IDEL45$5</td>
<td>IDEL46$5</td>
<td>IDEL51$5</td>
</tr>
<tr>
<td>IMS 12.1</td>
<td>n/a</td>
<td>IDEL46$6</td>
<td>IDEL51$6</td>
</tr>
<tr>
<td>IMS 13.1</td>
<td>n/a</td>
<td>n/a</td>
<td>IDEL51$7</td>
</tr>
<tr>
<td>IMS 14.1</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

To delete the old modules

1. From Table 12 on page 182, select the delete job that corresponds to the BMC product (or products) and IMS release installed at your site.

2. Edit the delete job you selected to change all &RESLIBs to the name of the data set where the old modules reside.

3. Run the delete job.

   If you added BBLINK to the IMS STEPLIB concatenation, replace the BBLINK data set in the current IMS STEPLIB concatenation with the new BBLINK data set.

   If you included BBLINK in the link list concatenation, replace the BBLINK data set in the current link list concatenation with the new BBLINK data set. This data set replacement will affect all the IMS address spaces running in that IBM OS/390 system.

Setting up BBPARM data sets

The customization instructions in this chapter refer to the following data sets:

- `hilevel.ibbparm`
- `hilevel.UBBPARM`
- `hilevel.BBPARM`

<table>
<thead>
<tr>
<th>Data set name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hilevel</td>
<td>is the high-level data set name qualifier used at your site</td>
</tr>
<tr>
<td>Data set name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ibbparm</td>
<td>is a user-defined parameter data set unique to this IMS. You can allocate a separate ibbparm data set to contain any members that you want to make unique to the IMS, such as IMFSYS00, IMFECPO00, and AAOPRM00. The ibbparm data set must be allocated. <strong>Note:</strong> If the only members that require customization for an IMS system are IMFSYS00, IMFECPO00, or both, an ibbparm data set is not required. Instead, you can create a renamed (imsidSYS) version of the IMFSYS00 member, a renamed (imsidECP) version of the IMFECPO00 member, or both, include them in UBBPARM, and edit them (as explained below in UBBPARM). If you require customized member AAOPRMxx, specify the suffix using the AAOPRM=xx parameter in the IMFSYS00 (or imsidSYS) member that the IMS system is using.</td>
</tr>
<tr>
<td>UBBPARM</td>
<td>is a parameter data set that is copied and customized from the distributed BBPARM data set and shared by all IMS systems. <strong>Note:</strong> If you used MainView Customization, you can edit and customize the UBBPARM data set that MainView Customization creates. If you did not use MainView Customization, allocate a UBBPARM data set, copy the distributed BBPARM data set to it, and edit the UBBPARM data set as needed. In the UBBPARM data set, you can include a copied, renamed version of IMFSYS00 to customize the system parameters for an IMS system, including the SUBSYS parameter, which establishes communication between the IMS and the BBI-SS PAS. The name of an IMS-specific system parameter member must be in the following format: <strong>imsidSYS</strong> (where imsid is the four-character IMS ID code) You can also include a copied, renamed version of IMFECPO00 to customize the Event Collector parameters for an IMS. The name of an IMS-specific Event Collector parameter member must be in the following format: <strong>imsidECP</strong> (where imsid is the four-character IMS ID code) You can include the AAOPRMxx member by specifying the suffix with the AAOPRM=xx parameter in the IMFSYS00 (or imsidSYS) member that the IMS system is using. If you do not need to customize additional UBBPARM members for an IMS system, you do not need to create and allocate an ibbparm member for that system.</td>
</tr>
<tr>
<td>BBPARM</td>
<td>is the target BBPARM data set distributed by BMC</td>
</tr>
</tbody>
</table>
Throughout the MainView AutoOPERATOR documentation, parameter library members are normally referred to as BBPARM members, even though customized versions of the members may reside in the UBBPARM data set or in an IBBPARM data set.

The system parameter member is referred to as BBPARM member IMFSYS00, and the Event Collector parameter is referred to as BBPARM member IMFEC0P00, even though the members might also reside in a parameter data set with a member name in the format imsidSYS or imsidECP.

Customizing BBPARM members for an IMS system

This section is provided primarily for new installations of MainView AutoOPERATOR for IMS. If you have a prior version installed, you can use your existing parameter settings, but this would be a good time to review them.

This following procedure customizes the parameter members required to establish communication between an IMS region and the BBI-SS PAS

Set the system parameters in an imsidSYS member in the UBBPARM data set or in an IMFSYS00 member in an ibbparm data set. The parameters and their uses are as follows:

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBSYS</td>
<td>Use this parameter to identify the subsystem name of the BBI PAS that the IMS region should communicate with. The subsystem should be the same as the one specified in the BBJNT00 member of the UBBPARM data set or an ibbparm data set.</td>
</tr>
<tr>
<td>MSGLVL1</td>
<td>Set the message level to MTO, WTO, BOTH, or NONE.</td>
</tr>
<tr>
<td>KEYWARN</td>
<td>Use this parameter to set a minimum number of days before expiration warning messages are issued for product keys. The default is 45 days. (The KEYWARN parameter applies to MVIMS Online, MVIMS Offline, and MainView for DBCTL only, and it is used only in the IMS control region, not the BBI-SS PAS.)</td>
</tr>
<tr>
<td>AOEINIT</td>
<td>This parameter is used with the DFSAOE00 initialization call. (See BBPARM member IMFSYS00 for more information.) This is ignored if you define your user exits in an IMS DFSDFxxx PROCLIB member. If you do, you must also define the BMC exit, DFSAOE00, as the last one in EXITDEF list.</td>
</tr>
</tbody>
</table>
### Activating MainView AutoOPERATOR for IMS

To activate MainView AutoOPERATOR for IMS, you must enable product initialization and product authorization.

#### Enabling product initialization

To enable MainView AutoOPERATOR for IMS initialization, in BBPARM member BBISSP00, specify

```
PRODUCT=AAO
```

#### Enabling product authorization

You must have a valid password to enable MainView AutoOPERATOR for IMS product authorization.

If you are licensed for the MainView AutoOPERATOR for IMS option, refer to the *Installation System User Guide* for information about how to install and activate the passwords for MainView AutoOPERATOR for IMS.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOEEXIT and AOIEXIT</td>
<td>Use these parameters to specify the names of user-written AO exit routines. You can also use AOEEXIT and AOIEXIT to specify the order in which AO exit routines get control and whether a return code is padded. See “Enabling AO exit routines” on page 187 for more information. These are ignored if you define your user exits in an IMS DFSDFxxx PROCLIB member. If you do, you must also define the BMC exit, DFSAOE00, as the last one in EXITDEF list. As distributed, IMFSYS00 activates all MVIMS components in IMS. You can use imsidSYS in the UBBPARM data set or IMFSYS00 in an ibbparm data set to temporarily deactivate one or more components. BBPARM member IMFSYSBB contains information about how to deactivate components. Copy what you need from IMFSYSBB to your UBBPARM imsidSYS member or ibbparm IMFSYS00 member.</td>
</tr>
</tbody>
</table>
Modifying the IMS control region JCL

The following sections describe how to allocate a BBPARM data set and provide IMS access to the BBLINK modules.

Allocating a BBPARM data set

To establish the parameters for MVIMS Online and MainView AutoOPERATOR for IMS, allocate the BBPARM data sets by adding the following statements to the IMS control region startup procedure:

```plaintext
//IMFPARM DD DSN=hilevel.ibbparm
// DD DSN=hilevel.UBBPARM
// DD DSN=hilevel.BBPARM
```

Giving IMS access to MainView AutoOPERATOR for IMS

The IMS control region must be able to access the BBLINK modules to enable execution of MVIMS Online and MainView AutoOPERATOR for IMS.

If the BBLINK data set is in the LNKLST concatenation, MainView AutoOPERATOR for IMS access is already established. If not, you can provide access by modifying your IMS control region JCL or by copying the required modules to the site authorized library (such as IMS RESLIB).

To modify the IMS control region JCL, add the BBLINK data set to the IMS STEPLIB concatenation. (BBLINK must be authorized.)

To copy the modules used in IMS, copy the individual BBLINK members to an authorized STEPLIB data set. Select one of the following jobs in BBSAMP:

- ICOPY5 for IMS 11.1
- ICOPY6 for IMS 12.1
- ICOPY7 for IMS 13.1
- ICOPY8 for IMS 14.1

Then edit the JCL and run the job. This job copies the appropriate BMC modules from the BBLINK library to the site-authorized library, such as IMS RESLIB. You must rerun the JCL each time you apply BMC service.
Enabling AO exit routines

You can skip this section if you have a prior version of MainView AutoOPERATOR for IMS installed and will use the same method for defining your exits in the new version.

IMS gives control to AO exit routines to do initialization processing and to do message processing. BMC supplies AO exit routine to do initialization processing and message processing. You can define your own exits with the AOEINIT, AOEXIT, and AOIEXIT parameters, in an imsidSYS member in the UBBPARM dataset or in an IMFSYS00 member in an ibbparm dataset. For IMS 13.1 and above, you can alternatively define your exits, and the BMC exits, directly to IMS. However, they should not be defined in both places. Unless you want to define them in DFSDFxxx, no changes are needed when using IMS 13.1 or above.

BMC supplies two AO exit routines to do message processing: a type-1 AO exit routine, DFSAOUE0, and a type-2 AO exit routine, DFSAOE00.

Note

See the IBM publication IMS Operations Guide for an explanation of how the DFSAOUE0 and DFSAOE00 routines differ.

The DFSAOUE0 and DFSAOE00 routines capture MTO messages and IMS commands and pass them to MainView AutoOPERATOR for IMS. They also perform the following functions:

- provide an interface with your AO exit routines (unless you use DFSDFxxx)
- automatically start up MainView for IMS monitors when IMS starts up
- provide an interface between the Event Collector and the BBI-SS PAS for workload data collection
- initialize the Event Collector

If you use DFSDFxxx to define your own AO initialization or message processing exits, the BMC DFSAOE00 message exit must also be defined as the last exit in the DFSDFxxx TYPE=AOI,EXITDEF= list. In that case, you should also remove any AOEINIT, AOEXIT, and AOIEXIT parameters from your BMC parameter library to avoid warning messages. Type-1 exits are not used in this case.

The following describes how BMC manages the AO exits and the features it provides. It does not apply if you define them directly to IMS through DFSDFxxx. In that case IMS manages and dispatches them.
On entry to DFSAOE00 with AOE0FUNC=1, MainView for IMS loads and executes any user exit specified in AOEINIT=xxxxxxxx one time only. The DFSAOE00 exit does not forward control to the user’s DFSAOE01 exit on the initialization call.

If you require a DFSAOE00 or DFSAOUE0 exit routine in addition to the exit provided by BMC Software, perform Step 1 and Step 2 below.

**To enable AO exit routines**

1. If you have
   a. One DFSAOE00 exit: Rename it DFSAOE01, or choose a different name and use the control statements described in Step B.
   b. Multiple DFSAOE00 exits in addition to the BMC Software-supplied exit, or one exit not named DFSAOE01: You must add one or more of the following AOEEXIT control statements to an imsid SYS member in UBBPARM or an IMFSYS00 member in an ibbparm data set.

   **Example**
   
   AOEEXIT=MYNAME
   AOEEXIT=DFSAOE02
   
   In the example above, both exits are loaded and executed by the BMC Software DFSAOE00 exit. Program MYNAME executes first.

2. If you have
   a. one DFSAOUE0 exit: Rename it DFSAOUE1, or choose a different name and use the control statements described in Step B.
   b. multiple DFSAOUE0 exits in addition to the BMC Software-supplied exit, or one or more exits not named DFSAOUE1: You must add one or more of the following AOIEXIT control statements to the imsidSYS member in UBBPARM or in an IMFSYS00 member in an ibbparm data set.

   **Example**
   
   AOIEXIT=MYNAME
   AOIEXIT=DFSAOUE2
   
   In the example above, both exits are loaded and executed by the BMC Software DFSAOE00 exit. Program MYNAME executes first.
Flow of control

This discussion does not apply if you defined your exits directly to IMS, in a DFSDFxxx member.

By default, the BMC Software DFSAOE00 exit invokes the BMC Software AO exit routine (IELOAD) before it invokes your user exits. The return code set by the last user exit processed is passed to IMS. You can change the default processing in an imsidSYS member in the UBBPARM data set or an IMFSYS00 member in an ibbparm data set by specifying

- the order that exits are to be processed
- the exit return code

**Example**

AOEEXIT=(DFSAOE01,RC)
AOEEXIT=IELOAD

In the example above, the DFSAOE01 user exit assumes control before the IELOAD exit and the DFSAOE01 return code is passed to IMS.

IMS then executes the DFSAOUE0 exit unless indicated otherwise by the return code set by the DFSAOE00 exit. The BMC Software DFSAOUE0 exit calls your exits in the order you specified with the AOIEXIT control statements. By default, the return code set by the last user exit processed is passed to IMS. You can pass the return code from another exit by specifying the RC parameter with the AOIEXIT control statement as shown in the following example:

**Example**

AOIEXIT=(USEMINE,RC)
AOIEXIT=IGNOREME

**Enhancement to the IMS DFSAOE00 interface**

If you defined your exits directly to IMS, in a DFSDFxxx member these enhancements are not available.

BMC Software enhances the programming interface when your DFSAOE00 exit is invoked in the following ways:

- Register 11 contains the address of the IMS SCD.
Register 13 contains the address of 15 prechained save areas.

MainView AutoOPERATOR for IMS considerations

If you install your user exit directly to IMS, you should also define the BMC exit, DFSAOE00, as the last exit in the IMS EXITDEF= list. If it is not defined in the EXITDEF list, MainView for IMS and MainView for AutoOPERATOR will not be available. This specification should be last in the list to ensure proper automation with MainView AutoOPERATOR for IMS Rules.

When you install your user exit with a BMC Software exit, you should consider the following information:

- If the BMC exit is executed before the user exit and the user exit’s return code cancels the processing of additional message segments of a multi-segment message, the message segments are also canceled for MainView AutoOPERATOR. The result is that incomplete IMS messages are passed to the MainView AutoOPERATOR and LAST SEG LOST messages from BBI. The MainView AutoOPERATOR AO exit holds the first segment for a certain length of time while waiting for additional segments, which can cause the messages to be processed out of time stamp sequence by MainView AutoOPERATOR.

- When the user exit is executed before the BMC exit, changes to the messages made by the user exit are received by the MainView AutoOPERATOR AO exit.
  
  — If the user exit sets the length code of a message (or a segment of a multi-segment message) to zero, MainView AutoOPERATOR does not process the message (or segment).
  
  — If the user exit sets the length code of the first segment to 0 and the return code to 4, all further segments are canceled for MainView AutoOPERATOR also.

**Note**

If any modules are specified with the AOIEXIT parameter in IMFSYS00 or imsidSYs, but the BMC exit is not specified, the exit still executes after all specified exits.

The MainView AutoOPERATOR AO exit and any other AO exit you use are under ESTAE protection when invoked by the BMC routines. If an abend occurs in one of these exits, only that routine is disabled; the other routine and IMS itself are not affected.
**Debug EXEC IAODEBUG**

You can schedule the EXEC %IAODEBUG from the MainView AutoOPERATOR Journal. Use this EXEC to find and debug issues for MainView AutoOPERATOR for IMS.

The IAODEBUG EXEC examines the connection between the PAS and IMS and gathers information about the following items:

- The IMS region
- The IMS-related parameter settings in the parmlib
- The status of IMS command processing
- Additional diagnostics data

After EXEC discovers problems, it returns information making suggestions about you might be able to correct the issues.

**Stopping message queuing to the IMS master terminal**

Two methods of stopping message queuing to the IMS master terminal are available:

- Physically turn off the master terminal (requires BBI Logging).

  MainView AutoOPERATOR detects that the master terminal is not operable and will cancel any messages that are queued.

- Create a MainView AutoOPERATOR IMS-initiated Rule to suppress messages from the master terminal. Refer to the section on Rules in the MainView AutoOPERATOR Basic Automation Guide, Volume 1: Using Rules.

If the master LTERM is defined as NON-SNA (in other words, as a local 3270), IMS **does not** indicate to the AOI exit that the terminal is inoperable.

To avoid this problem, you can

- change the definition of the master LTERM
- prevent the master LTERM from connecting to VTAM
- dynamically force the VTAM node inactive by issuing the command
Dynamic installation and uninstallation

Before version 4.1.00 of MainView for IMS (MVIMS) and version 6.5.00 of MainView AutoOPERATOR, the only way to install MVIMS Online and MainView AutoOPERATOR for IMS was to use the IMS control region JCL to install the products during IMS initialization.

MVIMS version 4.1.00 and MainView AutoOPERATOR version 6.5.00 have added a dynamic installation option. For these versions of the products and later, you have the following installation choices:

- the standard JCL method (described on “Modifying the IMS control region JCL” on page 186)
- the dynamic install function described in this section

If you use the dynamic install function to install the products, you do not need to modify the standard IMS JCL. MVIMS and MainView AutoOPERATOR for IMS data sets are not required in STEPLIB, and the IMFPARM DD and BMCPSWD DD statements are not necessary.

However, the recommended procedure is to use the standard JCL method to install the products, so that MVIMS Online and MainView AutoOPERATOR for IMS are always available when IMS is initialized. Then you can use the dynamic uninstall and install functions to implement MVIMS Online maintenance while IMS is still up and running.

If MVIMS backs itself out because of a software failure, you can use the dynamic uninstall and install functions to reinstall the product (after applying corrective maintenance, if appropriate) without having to stop and restart IMS.

If you plan to continue using the IMS control region JCL method to install MVIMS and MainView AutoOPERATOR for IMS as IMS initializes and you want to use dynamic uninstall and install to apply maintenance, follow these steps:

**To use dynamic uninstall and install to apply maintenance**

1. Run the ICOPYAOI job, which is provided in the BBSAMP data set. The job will copy members DFSAOE00 and DFSAOUE0 from your target library into a separate data set.
2 Change the IMS control region JCL:

a Replace the BBLINK data set in the STEPLIB concatenation with the new data set created by ICOPYAOI.

b Add an IMFLOAD DD pointing to the BBLINK data set.

DFSAOE00 and DFSAOUE0 will be loaded from STEPLIB. The JCL install process loads MVIMS and MainView AutoOPERATOR for IMS modules from IMFLOAD. IMFLOAD is deallocated after installation is completed so that you can apply maintenance while IMS is up. After maintenance is applied, you can uninstall and reinstall MVIMS and MainView AutoOPERATOR for IMS to incorporate the new maintenance, without stopping and restarting IMS.

### Dynamic install JCL

The sample dynamic install JCL, shown in Figure 30 on page 194, is used to install MVIMS and MainView AutoOPERATOR for IMS into an active IMS system.

The JCL is provided in BBSAMP member IBINSTL.

If the dynamic installation process is successful, the following messages are issued.

<table>
<thead>
<tr>
<th>From the IMS control region:</th>
<th>IB1133I INSTALL OF MAINVIEW FOR IMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND AUTOOPERATOR FOR IMS STARTED</td>
<td>--IMFLOAD DATASETS--</td>
</tr>
<tr>
<td></td>
<td>dsname.BBLINK</td>
</tr>
<tr>
<td></td>
<td>--IMFPARM DATASETS--</td>
</tr>
<tr>
<td></td>
<td>dsname.UBBPARM</td>
</tr>
<tr>
<td></td>
<td>--BMCPSWD DATASETS--</td>
</tr>
<tr>
<td></td>
<td>dsname.BMCPSWD</td>
</tr>
<tr>
<td></td>
<td>IB1129I OLT AOE EXIT DISPATCHING</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ENABLED</th>
<th>IM0202I MAINVIEW FOR IMS DD</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUNCTION ENABLED FOR FF AND FP IBX</td>
<td>IM0100I IMF/EC COMPONENT VERSION</td>
</tr>
<tr>
<td>4.1.00 ACTIVE IBX</td>
<td>IM0109I CICS=YES IN EFFECT IBX</td>
</tr>
<tr>
<td>4.1.00 ACTIVE</td>
<td>IM0113I MAINVIEW FOR IMS OFFLINE</td>
</tr>
<tr>
<td>4.1.00 ACTIVE</td>
<td>IM0113I MAINVIEW FOR IMS ONLINE</td>
</tr>
<tr>
<td>INSTALLED</td>
<td>IM0766I IMS AOI</td>
</tr>
<tr>
<td>(ssid) INTERFACE INITIALIZED</td>
<td>IE1110I IMS AO/IEINIT TO BBI/SS</td>
</tr>
<tr>
<td>INTERFACE INITIALIZED</td>
<td>IM0759I IMS AO/IMF TO BBI/SS (ssid)</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>IB1115I OLT VERSION 1.1.0</td>
</tr>
<tr>
<td>From the dynamic install batch job:</td>
<td>IB1148I ECAO INSTALL IS PROCEEDING</td>
</tr>
<tr>
<td>SUCCESSFULLY</td>
<td>IB1140I ECAO INSTALL COMPLETED</td>
</tr>
</tbody>
</table>
If the installation process fails, check the messages that are issued by the batch job to determine what went wrong. Related messages could also be written in the IMS control region. If you are not able to correct the problem that is indicated by the messages, contact BMC Customer Support.

If the control region issues message IB1136E, try to correct the indicated problem and then try the dynamic installation again. If the control region issues message IB1138E, you must run the uninstallation job first to clean up the partial installation attempt. Then you can try to correct the problem that caused the installation failure and try to install again.

**Note**
The BMCPSWD DD statement is not necessary if you are installing only the ‘AO’ component.

---

**Figure 30: Dynamic Install JCL**

```
// INSTALL JOB (XXXX),'USER NAME',CLASS=X,MSGCLASS=X,NOTIFY=&SYSUID
/*-----------------------------------------------*/
/* MainView FOR IMS AND AUTOOPERATOR FOR IMS DYNAMIC INSTALL JOB */
/* THIS JOB CAN BE USED TO INSTALL MainView FOR IMS AND AUTOOPERATOR */
/* FOR IMS IN AN ACTIVE IMS SYSTEM. */
/* THE STEPLIB, IMFLOAD, AND IMSRESL DATASETS MUST BE APF AUTHORIZED. */
/* PARAMETERS: */
/* COMP 'ECAO' - INSTALL THE MainView FOR IMS EVENT COLLECTOR */
/* AND MainView AUTOOPERATOR FOR IMS. */
/* 'EC' - INSTALL THE MainView FOR IMS EVENT COLLECTOR. */
/* 'AO' - INSTALL MainView AUTOOPERATOR FOR IMS. */
/* IMSID - IMS TARGET OF THE INSTALL */
/* INSTLDSN - THE DATA SET CONTAINING THE INSTALL PROGRAM */
/* (IBDYN00). THIS DATA SET WILL BE BOTH THE */
/* STEPLIB AND IMFLOAD. */
/* IMFPARMX - THE MainView FOR IMS AND AUTOOPERATOR FOR */
/* IMS PARAMETER DATA SETS. YOU MAY HAVE UP TO */
/* 5 DATA SETS. TO USE ADDITIONAL DATA SETS, */
/* UNCOMMENT AND SPECIFY A VALID DATA SET NAME */
/* FOR THE IMFPARM2= OR IMFPARM3= PARAMETERS */
/* AND UNCOMMENT THE APPROPRIATE LINES AFTER */
/* THE IMFPARM DD STATEMENT. IF YOU NEED MORE */
/* THAN 3 DATA SETS, MODIFY THE PROCEDURE. */
/* BMCPSWD - THE BMC PASSWORD DATA SET */
/*-----------------------------------------------*/

/DYNINSTL PROC COMP=,
// IMSID=
// INSTLDSN=
// IMFPARM1=
// IMFPARM2=
// IMFPARM3=
// BMCPSWD=
// IMSRESL=

/IBDYN00 EXEC PGM=IBDYN00,
// PARM='INSTALL,&COMP,&IMSID',
// TIME=1439,REGION=1024K
/* STEPLIB: DATASET CONTAINING INSTALL PROGRAM (IBDYN00) */
// STEPLIB DD DISP=SHR,DSN=&STGLDSN
```
The sample dynamic uninstall JCL, shown in Figure 31 on page 196, is used to uninstall MVIMS and MainView AutoOPERATOR for IMS from an active IMS system.

The JCL is provided in BBSAMP member IBUNINST.

When the dynamic uninstallation process is successful, the following messages are issued.

From the IMS control region:

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IE1134I</td>
<td>UNINSTALL OF MAINVIEW AUTOOPERATOR FOR IMS STARTED</td>
</tr>
<tr>
<td>IE1135I</td>
<td>UNINSTALL OF MAINVIEW AUTOOPERATOR FOR IMS ENDED</td>
</tr>
<tr>
<td>IB1134I</td>
<td>UNINSTALL OF MAINVIEW FOR IMS STARTED</td>
</tr>
<tr>
<td>IB1135I</td>
<td>UNINSTALL OF MAINVIEW FOR IMS ENDED</td>
</tr>
</tbody>
</table>

From the dynamic uninstall batch job:

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IB1149I</td>
<td>ECAO UNINSTALL IS PROCEEDING</td>
</tr>
</tbody>
</table>
If the uninstallation process fails, check the messages that are issued by the batch job to determine what went wrong. Related messages might also be written in the IMS control region. If you are not able to correct the problem that is indicated by the messages, contact BMC Customer Support.

Figure 31: Dynamic Uninstall JCL

```plaintext
//UNINSTALL JOB (XXXX),'USER NAME',CLASS=X,MSGCLASS=X,NOTIFY=&SYSUID
/*---------------------------------------------*/
/* MainView FOR IMS DYNAMIC UNINSTALL JOB
/* **
/* THIS JOB CAN BE USED TO UNINSTALL MainView FOR IMS AND/OR
/* MainView AUTOOPERATOR FOR IMS FROM AN ACTIVE IMS SYSTEM.
/* **
/* THE STEPLIB DATASETS MUST BE APF AUTHORIZED.
/* **
/* PARAMETERS:
/* ** COMP     'EC'    -UNINSTALL THE MVIMS EVENT COLLECTOR'
/* **       'AO'    -UNINSTALL MainView AUTOOPERATOR FOR IMS
/* **       'ECAO'  -UNINSTALL EC AND AO
/* **    NOTE: WHEN THIS JOB IS SUBMITTED WITH 'ECAO' OR
/* **        'AO', AUTOOPERATOR SHARED VARIABLE QAOIMSMUN
/* **    WILL BE SET TO 'YES', WHILE THE VARIABLE
/* **    IS SET TO 'YES', THE QIMS VARIABLES WILL
/* **   NOT BE MAINTAINED.
/* **    IMSID            -IMS TARGET OF THE UNINSTALL
/* **   UNSTLDSN         -THE DATA SET CONTAINING THE UNINSTALL PROGRAM
/* **                     (IBDYN00). THIS DATA SET WILL BE STEPLIB
/* **---------------------------------------------*/
//DYNUNSTL PROC COMP=,
//             IMSID=,
//             UNSTLDSN=
/*---------------------------------------------*/
//IBDYN00 EXEC PGM=IBDYN00,
//             PARM='UNINSTALL,&COMP,&IMSID',
//             TIME=1439,REGION=1024K
/*---------------------------------------------*/
/* STEPLIB: DATASET CONTAINING THE UNINSTALL PROGRAM (IBDYN00)
/*---------------------------------------------*/
//STEPLIB DD DISP=SHR,DSN=UNSTLDSN
/*---------------------------------------------*/
//SYSPRINT DD SYSOUT=*
//IBDYN00 PEND
/*---------------------------------------------*/
//EXEDYNAM EXEC DYNUNSTL
/*---------------------------------------------*/
```

Note
Time-initiated EXECs with QIS=YES specified are not quiesced when MainView AutoOPERATOR for IMS is dynamically uninstalled because the IMS region is still active. The shared variable, QAOIMSMUN is set to YES when MainView AutoOPERATOR for IMS is uninstalled. Use automation (Rules or EXECs) to query the QAOIMSMUN variable to determine if MainView AutoOPERATOR for IMS has been uninstalled.
Implementing MainView AutoOPERATOR Access NV

This chapter describes how to install the MainView AutoOPERATOR Access NV option into NetView, how to define the default NetView system to MainView AutoOPERATOR.

Overview

To implement the MainView AutoOPERATOR Access NV option to NetView and define the default NetView system to MainView AutoOPERATOR, you must complete the steps in the following table. When the steps are described in this chapter, the page number of the discussion appears in the right column.

<table>
<thead>
<tr>
<th>Task description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>modify NetView JCL</td>
<td>“Modifying NetView JCL” on page 199</td>
</tr>
<tr>
<td>create member NAIOPT00 in the NetView DSIPARM data set</td>
<td>“Creating DSIPARM member NAIOPT00” on page 199</td>
</tr>
<tr>
<td>create user-modifiable copies of the IBM DSIPARM members DSICMDU, DSIDMNU, and DSIOPFU</td>
<td>NA a</td>
</tr>
<tr>
<td>modify DSIPARM member DSICMDU</td>
<td>“Modifying DSIPARM member DSICMDU” on page 200</td>
</tr>
<tr>
<td>modify DSIPARM member DSIDMNU</td>
<td>“Modifying DSIPARM member DSIDMNU” on page 201</td>
</tr>
<tr>
<td>if you are using NetView 5.1, modify DSIPARM member CNMSTYLE</td>
<td>“Modifying DSIPARM member CNMSTYLE” on page 201</td>
</tr>
<tr>
<td>copy members NAIINIT and NAISVAR into the NetView DSICLD data set</td>
<td>“Copying BBSAMP members to the NetView DSICLD data set” on page 202</td>
</tr>
<tr>
<td>modify member DSIOPFU in DSIPARM</td>
<td>“Modifying DSIPARM member DSIOPFU” on page 202</td>
</tr>
<tr>
<td>verify operator profiles</td>
<td>“Verifying operator profiles” on page 203</td>
</tr>
</tbody>
</table>
Allocating SYSUDUMP data sets

This procedure describes allocating SYSUDUMP data sets.

*Note*
Beginning with version 6.4, MainView AutoOPERATOR Access NV creates SVC dumps by default. BMC Software recommends the use of SVC dump data sets. If you want your default to be SVC dumps, you can skip the steps in this section. However, if you want to obtain SYSUDUMP's, you must code DUMPTYPE=SYSOUT in DSIPARM member NAIOPT00. Additionally, if NetView is running SUB=MSTR, you must perform the 3 steps listed below.

1. Allocate a minimum of three dump data sets with RECFM=VBA and LRECL=125, BLKSIZE=1632, DSORG=PS. Make each one large enough to accommodate the largest dump that you expect.

2. Add the following allocations to your NetView address space JCL, pointing to the dump data sets that were created in the previous step.

```
//BBDUMP01 DD DISP=SHR, DSN=dumpdsn1, VOL=SER=...., UNIT=....
//BBDUMP02 DD DISP=SHR, DSN=dumpdsn2, VOL=SER=...., UNIT=....
//BBDUMP03 DD DISP=SHR, DSN=dumpdsn3, VOL=SER=...., UNIT=....
```
3 Catalog all data sets that are referenced in the NetView address space JCL PROC in the master catalog or change all allocations to UNIT/VOLSER allocations.

Modifying NetView JCL

To make MainView AutoOPERATOR programs and parameters available in the NetView address space, modify the NetView JCL:

1. Add the MainView AutoOPERATOR BBLINK library to the NetView STEPLIB concatenation.

2. Add the following new DD statements to the NetView JCL:
   
   ```
   //BBIPARM DD ... DSN=hiLV1.BBPARM,DISP=SHR
   //BBILOAD DD ... DSN=hiLV1.BBLINK,DISP=SHR
   ```

Creating DSIPARM member NAIOPT00

Create member NAIOPT00 in the NetView DSIPARM data set to define

- BBI-SS PAS with which NetView will communicate
- NetView operator ID's that are used to execute IMFEXEC NETVIEW commands from MainView AutoOPERATOR EXECs and NetView commands from Rules

Modifying NetView JCL

Chapter 20  Implementing MainView AutoOPERATOR Access NV
3 Add up to 19 OPERxx statements.

The OPERxx= statements specify operator IDs to NetView that are used to run commands and EXECs which originate from

- EXECs running in the MainView AutoOPERATOR BBI-SS PAS that execute the IMFEXEC NETVIEW command
- MainView AutoOPERATOR Rules that specify a NV command as part of the defined Action Specification for the Rule

You must specify at least one OPERxx statement; multiple statements do not need to be sequential. Specifying multiple OPERxx statements allows multiple NetView commands (and EXECs) to run at the same time.

The maximum number of OPERxx statements is 19.

The operator IDs specified should be used solely for MainView AutoOPERATOR Access NV. A user should not log on to NetView with any of these IDs.

Refer to “Reviewing NetView security for Access NV operator IDs” on page 203 for more information about creating security for the operator IDs that you specify in this step.

Modifying DSIPARM member DSICMDU

Modifying DSIPARM member DSICMDU

MainView AutoOPERATOR Access NV uses two command processors that must be defined to NetView in the DSICMDU member in the DSIPARM data set.

Modify the DSIPARM member DSICMDU by appending the contents of BBSAMP member NAICMDS to the end of your existing DSICMDU member.

This action adds definitions for the three NetView command processors that are used by MainView AutoOPERATOR Access NV, NAIEXEC, AOEXEC and NAISTUB.

BMC Software recommends using AOEXEC as a more efficient alternative to using NAIEXEC for performing automation tasks from the NetView address space. Using this command does not require the use of CSA storage which NAIEXEC does. It also allows greater portability because the same NetView EXEC that is used to invoke AOEXEC can also be invoked from TSO or other AutoOPERATOR address spaces.

**Note**

Do not attempt to use REXX LINKMVS to invoke AOEXEC. The AOEXEC fails and you will receive a return code of 54.
For more information about the functions of AOEXEC, refer to the *MainView AutoOPERATOR Advanced Automation Guide.*

## Modifying DSIPARM member DSIDMNU

If you are running NetView version 5.1 or later, skip this section and proceed to “Modifying DSIPARM member CNMSTYLE” on page 201.

Some Access NV functions run as an optional subtask (OPT) under NetView. Modify the DSIPARM member DSIDMNU to define this OPT by inserting the contents of BBSAMP member NAITASK to your existing DSIDMNU member.

This action adds a definition for the NetView optional subtask (OPT), NATASK, that is used by Access NV.

INIT=Y is included on the NATASK definition so that Access NV automatically initializes when NetView starts. If this parameter is modified, the task may also be started by using the NetView START TASK=NATASK command.

The default NATASK definition does not have a priority specification. If high activity in the NetView address space prevents acceptable response time from Access NV, you might want to specify a priority for NATASK.

## Modifying DSIPARM member CNMSTYLE

If you are running any version of NetView earlier than NetView version 5.1, skip this section and refer to “Modifying DSIPARM member DSIDMNU” on page 201.

Some Access NV functions run as an optional subtask (OPT) under NetView. Modify the DSIPARM member CNMSTYLE to define this OPT by inserting the contents of BBSAMP member NAITASK5 to your existing DSIDMNU member.

This action adds a definition for the NetView optional subtask (OPT), NATASK, that is used by Access NV.

INIT=Y is included on the NATASK definition so that Access NV automatically initializes when NetView starts. If this parameter is modified, the task may also be started by using the NetView START TASK=NATASK command.
The default NATASK definition does not have a priority specification. If high activity in the NetView address space prevents acceptable response time from Access NV, you might want to specify a priority for NATASK.

**Copying BBSAMP members to the NetView DSICLD data set**

To make the three CLISTs NAIINITX, NAIINIT, and NAISVAR, available to NetView, copy BBSAMP members NAIINITX, NAIINIT, and NAISVAR into the DSICLD data set.

**Modifying DSIPARM member DSIOPFU**

Modifying the DSIPARM member DSIOPFU is the default method for defining NetView operator IDs. If you maintain your NetView operator IDs in an external security manager (ESM) such as RACF, you need to define these operator IDs in your ESM.

**To modify member DSIOPFU in DSIPARM**

1. Add the new operator IDs that were previously specified in NAIOPT00.

   The profile specified for these IDs must be similar to the example profile in BBSAMP member NAIPROFX. See the following section for more information.

2. Add NetView operator IDs that will use the MainView AutoOPERATOR NetView Operator Workstation.

   You must define a NetView operator ID for each TSO user ID and each MainView Alternate Access user ID that will use the MainView AutoOPERATOR NetView Operator Workstation application.

   Specify a NetView profile that is similar to the example profile in BBSAMP member NAIPROF. Refer to “Verifying operator profiles” on page 203 for more information.
Verifying operator profiles

All Access NV controlled operator station tasks (OSTs) must execute an initialization CLIST that contains the NAISTUB command. This requirement applies to TS users as well as the OPERxx IDs used for IMFEXEC NETVIEW commands. If you use an ESM such as RACF, you must make the appropriate changes to the user definition for the IC parameter in your user definition.

To ensure that NAISTUB is executed

1. Copy BBSAMP members NAIPROF and NAIPROFX into the NetView DSIPRF data set.

2. If no profile specification already exists for an operator ID, specify either NAIPROFX or NAIPROF and use
   - NAIPROFX for OPERxx IDs that were specified in NAIOPT00
   - NAIPROF for TS operator IDs

3. If a profile is already specified for an operator ID, browse the profile in the NetView DSIPRF data set to determine which initialization CLIST is being used.

4. Edit this CLIST in the NetView DSICLD data set to include the statements from either NAINITX or NAINIT and use
   - NAINITX for OPERxx IDs that were specified in NAIOPT00
   - NAINIT for TS operator IDs

Reviewing NetView security for Access NV operator IDs

In Step 3 “Creating DSIPARM member NAIOPT00” on page 199, you added OPER01= through OPERxx= statements, which are operator IDs defined to NetView and that NetView uses to run commands and EXECs that originate from

- EXECs running in MainView AutoOPERATOR BBI-SS PAS that include the IMFEXEC NETVIEW command
- MainView AutoOPERATOR Rules that specify a NV command as part of the defined Action Specification for the Rule
You must authorize these operator IDs that are defined to NetView to run any commands or EXECs that originate from MainView AutoOPERATOR. You must also authorize the operator IDs to use the STOP and START commands because these commands may be used internally by MainView AutoOPERATOR without specifying them as a NetView Command or that may be used within a NetView EXEC.

In other words, these operator IDs must be defined to have enough NetView authority to perform any command requested from MainView AutoOPERATOR.

**Note**

When NetView is accessed from a TS session, the NetView operator ID is the same as the TS user ID. Standard NetView security applies to the operator ID.

When NetView is accessed from a MainView AutoOPERATOR EXEC (IMFEXEC NetView), the NetView operator ID is chosen by using a least-recently used algorithm from the pool of OPERxx IDs defined in DSIPARM member NAIOPT00. Standard NetView security applies to the operator ID.

---

### Reviewing the use of NetView exit DSIEX02A

MainView AutoOPERATOR Access NV uses the NetView exit DSIEX02A.

If you do not have a DSIEX02A exit, skip step 2 and go to step 3.

If you already use this exit, perform the following steps to ensure that your existing DSIEX02A exit and the MainView AutoOPERATOR Access NV DSIEX02A exit can coexist:

1. Rename your existing DSIEX02A load module to DSIEX02B.

2. Link the two exits by using the following linkage editor statements:

   ```
   //OLDLIB DD DSN=your.user.exit.library
   //SYSLMOD DD DSN=AutoOPERATOR.BBLINK
   //SYSIN DD *
   CHANGE DSIEX02A(DSIEX02B)
   INCLUDE OLDLIB(DSIEX02A)
   INCLUDE SYSLMOD(DSIEX02A)
   ORDER DSIEX02A
   ENTRY DSIEX02A
   NAME DSIEX02A(R)
   /*
   ```

MainView AutoOPERATOR Access NV first passes control to DSIEX02B. If DSIEX02B has modified the incoming message, MainView AutoOPERATOR Access NV processes the modified message. If the exit indicates that the message should be discarded, MainView AutoOPERATOR Access NV returns control to NetView with a return code 4.
3 For NetView version 5.1 and later, you must modify the DSIPARM member CNMSTYLE to activate this exit.

See the *Tivoli NetView for z/OS Administration Reference Guide* for more information about loading NetView exits (LOADEXIT).

---

### Reviewing the use of NetView exit DSIEX03

MainView AutoOPERATOR Access NV uses the NetView exit DSIEX03.

If you do not have a DSIEX03 exit, no other action is required for this step.

If you already use this exit, perform the following steps to ensure that your existing DSIEX03 exit and the MainView AutoOPERATOR Access NV DSIEX03 exit can coexist:

1. Rename your existing DSIEX03 load module to DSIEX03B.

2. Link the two exits by using the following linkage editor statements:

   ```
   //OLDLIB DD DSN=your.user.exit.library
   //SYSLMOD DD DSN=AutoOPERATOR.BBLINK
   //SYSIN DD *
   CHANGE DSIEX03(DSIEX03B)
   INCLUDE OLDLIB(DSIEX03)
   INCLUDE SYSLMOD(DSIEX03)
   ORDER DSIEX03
   ENTRY DSIEX03
   NAME DSIEX03(R)
   /*
   MainView AutoOPERATOR Access NV first processes the MainView AutoOPERATOR exit and control is passed to the NetView DSIEX03 exit. The Registers from NetView are passed with the exception of the return address and R13. The return codes are passed to NetView.

3 For NetView version 5.1 and later, you must modify the DSIPARM member CNMSTYLE to activate this exit.

---

### Reviewing the use of NetView exit DSIEX14

MainView AutoOPERATOR Access NV uses the NetView exit DSIEX14.

If you do not have a DSIEX14 exit, no other action is required for this step.
If you already use this exit, perform the following steps to ensure that your existing DSIEX14 exit and the MainView AutoOPERATOR Access NV DSIEX14 exit can coexist:

1. Rename your existing DSIEX14 load module to DSIEX14B.

2. Link the two exits by using the following linkage editor statements:
   ```
   //OLDLIB DD DSN=your.user.exit.library
   //SYSLMOD DD DSN=AutoOPERATOR.BBLINK
   //SYSIN   DD *
   CHANGE DSIEX14(DSIEX14B)
   INCLUDE OLDLIB(DSIEX14B)
   INCLUDE SYSLMOD(DSIEX14)
   ORDER DSIEX14
   ENTRY DSIEX14
   NAME DSIEX14(R)
   ```

MainView AutoOPERATOR Access NV first passes control to DSIEX14B with Register 1 pointing to the User Service Block (DSIUSE). Upon return from DSIEX14B, DSIEX14 will process the LOGOFF request and then return to NetView.

3. For NetView version 5.1 and later, you must modify the DSIPARM member CNMSTYLE to activate this exit. Edit the DSIEX014=NO to YES.

---

Reviewing the use of NetView exit DSIEX16

MainView AutoOPERATOR Access NV uses the NetView exit DSIEX16.

If you do not have a DSIEX16 exit, no other action is required for this step.

If you already use this exit, perform the following steps to ensure that your existing DSIEX16 exit and the MainView AutoOPERATOR Access NV DSIEX16 exit can coexist:

1. Rename your existing DSIEX16 load module to DSIEX16B.

2. Link the two exits by using the following linkage editor statements:
   ```
   //OLDLIB DD DSN=your.user.exit.library
   //SYSLMOD DD DSN=AutoOPERATOR.BBLINK
   //SYSIN   DD *
   CHANGE DSIEX16(DSIEX16B)
   INCLUDE OLDLIB(DSIEX16B)
   INCLUDE SYSLMOD(DSIEX16)
   ORDER DSIEX16
   ENTRY DSIEX16
   NAME DSIEX16(R)
   ```

MainView AutoOPERATOR Access NV first processes the MainView AutoOPERATOR exit and control is passed to the NetView DSIEX16 exit. The
Registers from NetView are passed with the exception of the return address and R13. The return codes are passed to NetView.

3 For NetView version 5.1 and later, you must modify the DSIPARM member CNMSTYLE to activate this exit.

Modifying BBPARM member BBISSP00

Add a NETVIEW specification to BBPARM member BBISSP00. This specification defines the default NetView target for the BBI-SS PAS. The BBI-SS PAS must be restarted for the parameter change to take effect.

The default NetView target is used when initially displaying the Access NV workstation and as the target for any IMFEXEC NETVIEW commands that do not specify a target. However, you can access other NetView systems by overtyping the NetView target on the Access NV workstation or specifying a job name on the IMFEXEC NETVIEW command.
Implementing BMC Event Manager to work with BMC products

This chapter describes how to implement a MainView AutoOPERATOR connection with BMC Event Manager by using BMC Impact Integration for z/OS (BMC II for z/OS).

For information about BMC Impact Integration for z/OS, refer to the MainView AutoOPERATOR BMC Impact Integration for z/OS User Guide.

Overview of bidirectional communications

You can implement MainView AutoOPERATOR to BMC Event Manager communication by implementing a TCP/IP-based interface between MainView AutoOPERATOR and BMC Event Manager.

Refer to “Establishing a connection to a BMC Event Manager” on page 210 for more information.

Communication between MainView AutoOPERATOR and BMC Event Manager is bidirectional:

- Using BMC Impact Explorer, you can view MainView AutoOPERATOR ALERTs and events and close the ALERTs on MainView AutoOPERATOR from the Explorer.

- MainView AutoOPERATOR users can send MainView AutoOPERATOR ALERTs and events to BMC Event Manager for cross-platform automation or business system modeling.

- BMC Event Manager users can send requests to run commands and EXECs in MainView AutoOPERATOR.
BMC Event Manager users can send requests to issue write-to-operator messages (WTOs) with MainView AutoOPERATOR.

To implement the bidirectional communication, you must establish a connection to each BMC Event Manager cell and specify the BMC Impact Integration Server in MainView AutoOPERATOR which is described in “Establishing a connection to a BMC Event Manager” on page 210.

Establishing a connection to a BMC Event Manager

Use the following procedure to enable MainView AutoOPERATOR to communicate with BMC Event Manager through a TCP/IP connection.

Before you begin

To establish a connection between MainView AutoOPERATOR and BMC Event Manager, you must have

- at least one BBI-SS PAS where MainView AutoOPERATOR 6.5 or later is running
- at least one BMC Event Manager system that has been configured to communicate with MainView AutoOPERATOR 6.5 or later

For more information, refer to the MainView AutoOPERATOR BMC Impact Integration for z/OS User Guide.

- the z/OS TCP/IP product

To establish a connection to a BMC Event Manager

1. In BBPARM member BBISSP00, specify

   \[ GTS=00 \]

   If default TCP specifications are insufficient for your installation, specify a two-character qualifier instead of YES and create a BBPARM member BBTTCP00.
   (Refer to "BBPARM Member BBTTCP00" in the MainView Customization Reference for more information about the BBTTCP00 member.)

2. Create a BBPARM member AAOGME00 (or edit an existing one) to identify the BMC Event Manager cells that are to be connected.
The BBPARM member AAOGME00 allows MainView AutoOPERATOR to use the General Messages Exchange (GME) facility to share MainView AutoOPERATOR ALERTs and events with BMC Event Manager.

This member name can be overridden by using the AAOGME=xx card in the BBCFG file, where xx is the suffix to use for the AAOGME member. See “BBPARM data set members for MainView AutoOPERATOR” on page 281 for complete details about this member.

3 (required) In BBPARM member AAOGMExx, specify

\[ \text{GMEACT}=\text{YES} \]

4 (optional) To override the name of the local GME node or to uniquely identify this instance of MainView AutoOPERATOR in the enterprise, in BBPARM member AAOGMExx, specify

\[ \text{HOSTNAME}=\text{value} \mid \text{SSID} \]

where value is the name of the local GME node. If you do not specify a value, the default is the SSID of the MainView AutoOPERATOR BBI-SS PAS and it must be unique among all instances of MainView AutoOPERATOR in the enterprise.

**Note**

This value should be specified in the mcell.dir to specify with which GME node the BMC Event Manager cell will communicate with. For more information, refer to the BMC Impact Manager Installation Guide.

It should also be specified in the IIMDIR00 configuration file; for more information; refer to Chapter 4, "Configuring BMC Impact Integration for MainView" in the MainView AutoOPERATOR BMC Impact Integration for z/OS User Guide.

5 (optional) To specify the name of the host where the MainView Explorer server is running and the port number that it is listening on, in BBPARM member AAOGMExx, specify

\[ \text{MVE\_LINK}=\text{host\_name}:\text{port\_number} \]

where

- The host_name is specified as a host name (such as FQDN) or an IP address. When the host_name is specified, you must also specify a port number. This information is used in the mv_mve_host and mv_mve_port slots for BIIZ events.

- There is no default host name. If not specified, the mv_mve_host and mv_mve_port will be null.
6 (required) Specify the BMC Impact Integration server

This step describes the settings for the TGTIP and TGTBIMCF parameters that you must enter to establish the BMC Impact Integration server that facilitates bidirectional communication between MainView AutoOPERATOR and any connecting BMC Event Manager cells.

For an example of what your TGTIP and TGTBIMCF settings might look like after you have completed this step and Step 7, refer to Figure 32 on page 216.

a In BBPARM member AAOGMExx, specify

\[ \text{TGTIP=HOSTNAME\_Value | SSID} \]

where the HOSTNAME\_Value | SSID specification matches the value specified for the HOSTNAME parameter in Step 4.

b In BBPARM member AAOGMExx, specify

\[ \text{TGTBIMCF= BIM\_configuration\_file\_name} \]

where BIM\_configuration\_file\_name is the data set name of the BMC Impact Integration configuration file name.

The file name can also reference a PDS member name using the following format:

\[ \text{TGTBIMCF=DD:<ddname>(IIMCNFnn)} \]

or

\[ \text{TGTBIMCF=<dsname>(IIMCNFnn)} \]

For example:

\[ \text{TGTBIMCF=DD:UBBPARM(IIMCNF01)} \]

or

\[ \text{TGTBIMCF=BMC.SYSA.IIMPARM(IIMCNF01)} \]

This parameter setting must directly follow the setting for TGTIP. Refer to Figure 32 on page 216 for an example.

Refer to Chapter 4, "Configuring BMC Impact Integration for MainView" in the MainView AutoOPERATOR BMC Impact Integration for z/OS User Guide for more information about what the BMC Impact Integration configuration file name is and where it is created.
After this step is complete, MainView AutoOPERATOR will be able to receive requests from BMC Event Manager. This required step allows BMC Event Manager to send actions through the BMC Impact Integration server to MainView AutoOPERATOR that include deleting ALERTs, sending commands or scheduling EXECs to run on MainView AutoOPERATOR.

c In BBPARM member AAOGMExx, specify EVENTS= YES.

This specifies whether the connection sends incoming BRC events to the Rules Processor BAROC messages from ADAPTORs will appear to the MainView AutoOPERATOR Rules processor as BRC (BAROC) events.

You can choose to prevent all incoming BAROC messages from cells, ADAPTORs, or from other BMC products not defined in AAOGMExx from being sent to the MainView AutoOPERATOR Rules Processor. Specify EVENTS= NO on the definition of the server connection; for example: HOSTNAME=ssid of AutoOPERATOR (note this is the default value of HOSTNAME) TGTIP=ssid of AutoOPERATOR, EVENTS= NO.

This parameter setting must directly follow the setting for TGTIP. Refer to Chapter 4, "Configuring BMC Impact Integration for MainView" in the MainView AutoOPERATOR BMC Impact Integration for z/OS User Guide for more information about what the BMC Impact Integration configuration file name is and where it is created.

7 (required) Establish connections to the BMC Event Manager cells

This step describes the settings for the TGTIP and TGTBIMCF parameters that you must enter to establish a connection between MainView AutoOPERATOR and the BMC Event Manager cell.

For each BMC Event Manager cell that you want to communicate with MainView AutoOPERATOR, you must repeat this step.

a In BBPARM member AAOGMExx, specify

TGTIP= BMC_Event_Manager_cell_name

where the BMC_Event_Manager_cell_name specification matches the BMC Event Manager cell name that is specified in the hilevel_qualifier.IIM.DIR data set.

Refer to Chapter 4, "Configuring BMC Impact Integration for MainView" in the MainView AutoOPERATOR BMC Impact Integration for z/OS User Guide for more information about what the BMC Event Manager cell name is and where it is created.

b In BBPARM member AAOGMExx, specify
TGTBIMCF=BIM_configuration_file_name

where BIM_configuration_file_name is the data set name of the BMC Impact Integration configuration file name.

The file name can also reference a PDS member name using the following format:

TGTBIMCF=DD:<ddname>(IIMCNFnn)

or

TGTBIMCF=<dsname>(IIMCNFnn)

For example:

TGTBIMCF=DD:UBBPARM(IIMCNF02)

or

TGTBIMCF=BMC.SYSA.IIMPARM(IIMCNF02)

This parameter setting must directly follow the setting for TGTIP. Refer to Figure 32 on page 216 for an example.

Refer to Chapter 4, "Configuring BMC Impact Integration for MainView" in the MainView AutoOPERATOR BMC Impact Integration for z/OS User Guide for more information about what the BMC Impact Integration configuration file name is and where it is created.

c  (optional) In BBPARM member AAOGMExx, specify

TGTNAME= BIM_connection_name

where BIM_connection_name is an up to 16 character name for a BMC IM connection.

The TGTNAME can be shared by multiple BIM connections and is used by MainView AutoOPERATOR Rules or EXECs to send events or publish ALERTs to a specific target system.

In Rules and EXECs, you may use wildcards when specifying the target name for events and ALERTs. There is no default value.

d  (optional) In BBPARM member AAOGMExx, specify

EVENTS=YES
which specifies whether the connection sends incoming BRC events to the Rules Processor.

BAROC messages from ADAPTORs will appear to the MainView AutoOPERATOR Rules processor as BRC (BAROC) events.

You can choose to prevent all incoming BAROC messages from cells, ADAPTORs, or from other BMC products not defined in AAOGMExx from being sent to the MainView AutoOPERATOR Rules Processor. Specify EVENTS=NO on the definition of the server connection; for example:

\[
\text{HOSTNAME=ssid of AutoOPERATOR (note this is the default value of HOSTNAME)}
\]

\[
\text{TGTIP=ssid of AutoOPERATOR,EVENTS=NO}
\]

e In BBPARM member AAOGMExx, specify

\[
\text{EXEC=Y}
\]

This parameter setting specifies whether to allow the launching of MainView AutoOPERATOR EXECs from a BMC Event Manager cell that will run in the MainView AutoOPERATOR BBI-SS PAS.

f In BBPARM member AAOGMExx, specify

\[
\text{WTO=Y}
\]

This parameter setting specifies whether a BMC Event Manager cell can send write-to-operator message (WTO) through MainView AutoOPERATOR.

8 (required) Reset the BBPARM member AAOGMExx by using the BBI command:

\[
\text{.RESET PARM AAOGMExx RESYNC}
\]

This command starts the communications to the BMC Event Manager cells and the MainView AutoOPERATOR Impact Integration server.

For complete information about each of the parameters in BBPARM member AAOGMExx in this section, refer to “BBPARM member AAOGME00 parameters” on page 296.
**Example of AAOGME\(x\) settings**

The following figure shows an example of what the settings in BBP Parm member AAOGME\(x\) may look like after you have completed steps 2 through 7 in this section.

**Figure 32: Example of AAOGME\(x\) settings**

<table>
<thead>
<tr>
<th>Line</th>
<th>Example of AAOGME(x) parameters</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>GMEACT is set to YES</td>
<td>“Establishing a connection to other BMC products” on page 217</td>
</tr>
<tr>
<td>(2)</td>
<td>HOSTNAME specifies the SSID of the MainView AutoOPERATOR BBI-SS PAS</td>
<td>“Establishing a connection to other BMC products” on page 217</td>
</tr>
<tr>
<td>(3)</td>
<td>MVE_LINK specifies the host name of BBSYSC and a port number of 3941 for where the MainView Explorer server is running</td>
<td>“Establishing a connection to other BMC products” on page 217</td>
</tr>
<tr>
<td>(4)</td>
<td>TGTIP and TGTBIMCF specifies the MainView AutoOPERATOR BBI-SS PAS and the data set name of the BMC Impact Integration configuration file name. The value for TGTIP must match the HOSTNAME Value</td>
<td>“Establishing a connection to other BMC products” on page 217</td>
</tr>
<tr>
<td></td>
<td>SSID specification as specified in “Establishing a connection to other BMC products” on page 217.</td>
<td></td>
</tr>
<tr>
<td>(5)</td>
<td>through (7) three BMC Event Manager cell names, HAGGAI, LUGGAGE, and RINCEWIN, and the data set name of the BMC Impact Integration configuration file name</td>
<td>“Establishing a connection to other BMC products” on page 217 repeated 3 times for 3 BMC Event Manager cells</td>
</tr>
</tbody>
</table>

**Configuring BBP Parm member BBISSP00 to allow rules to send events to BMC Event Manager**

MainView AutoOPERATOR Rules can send events and ALERTS to a BMC Event Manager cell.
To enable this feature, you must edit BBPARM member BBISSP00 and specify

```
PRODUCT=I1Z
```

## Establishing a connection to other BMC products

MainView AutoOPERATOR can connect to other BMC products that use the same communications protocol (cell protocol) as the BMC Event Manager product. The cell protocol transports Basic Recorder of Objects in C (BAROC) messages.

If you do not have the BMC Event Manager product installed, you can still receive and inspect BAROC messages by using the MainView AutoOPERATOR Rules Processor. You can also send BAROC messages with a Rule or through an EXEC by using the IMFEXEC BIM EVENT command.

While these other BMC products have a cell protocol adaptor (ADAPTOR) they cannot process MainView AutoOPERATOR ALERTs so MainView AutoOPERATOR does not send ALERTs to ADAPTORs.

The following section describes a parameter in BBPARM member AAOGME00 that provides support for these ADAPTOR connections.

### Table 13: Parameters in BBPARM member AAOGMExx for ADAPTORs

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>TGTTYPE=[CELL</td>
<td>ADAPTOR]</td>
</tr>
</tbody>
</table>

For example, MainView AutoOPERATOR can connect and send BAROC messages to a BMC Atrium Orchestrator ADAPTOR (BAO1) without having an intermediary BMC Event Manager. The following example shows how this connection is specified in BBPARM member AAOGME00:

```
TGTIP=BAO1,TGTBIMCF=SYS2.SYSC.IIM.CONF,TGTTYPE=ADAPTOR
```
See the *MainView AutoOPERATOR Basic Automation Guide, Volume 1: Using Rules* for more information about BRC events and Rules.
Using the MainView AutoOPERATOR interface to Control-O and CMEM

This chapter describes how to implement an interface between MainView AutoOPERATOR and the Control-O and Control-M Event Manager (CMEM) products.

By following the instructions in this chapter, you will be able to use the **Set IOA Variable** field on the Rule Processor Action Specification(s) panel. The Rule sends a request to set an IOA variable to all subscribing Control-O or CMEM regions on the local LPAR.

From there it can be sent to other INCONTROL products in the sysplex by using INCONTROL resources and facilities.

**Note**

You must be running version 6.2 or later of Control-O or CMEM with the following PTFs applied:

- PA16814
- PA16815
- PA16816
- PA16817
- PA16818

To support sending MainView AutoOPERATOR requests through CMEM, you must also apply the PTF described in the BMC APAR number IO10019. Without this PTF applied, MainView AutoOPERATOR cannot send requests to CMEM.

To allow Control-O or CMEM to process these requests, you must write one or more rules in Control-O to process the MainView AutoOPERATOR request event. You can
review an example of the Control-O rule in BBSAMP member AOSIVTBL and copy the data in this member into the appropriate Control-O or CMEM library. You might also need to modify it and activate it so that Set IOA Variable events from MainView AutoOPERATOR can be processed in Control-O.

Implementing the MainView AutoOPERATOR interface to Control-O and CMEM

The following high level outline shows additional implementation steps. Refer to the Control-O product manuals or your Control-O administrator for additional information.

1. Select the Control-O or CMEM address spaces to which you want to send the Set IOA Variable request.

2. Use the ICE tool that is provided by Control-O or CMEM to register the MainView AutoOPERATOR subsystem as a COMPONENT.

   If the ICE tool is unavailable, manually register the MainView AutoOPERATOR BBI-SS PAS name by editing the following members:

   - for Control-O, the CTOPARM parameter library member for Control-O
   - for CMEM, the CMMPARM parameter library member

   In the COMPONEN section of the parameter library member, add the following statements:

   ```
   COMPONEN . . .
   . . .
   AOSUBSYS=ssss  AutoOPERATOR Subsystem name to support
   ```

   See the INCONTROL for z/OS Installation Guide for more information about AOSUBSYS keyword.

3. Copy BBSAMP member AOSIVTBL to the '&OLPREFO..RULES' CTO or CMEM rule table library.

4. Add the AOSIVTBL table name to the list of rules that are loaded from the DACTMLST (for CMEM) or DARULLST (for CTO) DD when CTO or CMEM is started.

5. Restart the CTO or CMEM task or manually ORDER (or load) the table in the CTO or CMEM address space. To manually order a table, follow the instructions below for the product (CTO or CMEM) that you are using.
a) For CTO, use the Rule Definition panel (the OR command from the IOA online system) to ORDER the table.

Use the O command in the OPT field on the Table List panel to order (activate) the table (and the rules it contains). You will receive a confirmation message that the table has been ordered. After the table has been ordered, you can check the status of the AOSIVRUL rule by using the Rule Status panel with the OS command. It should be marked Active.

b) For CMEM, use the CMEM Rule Definition panel (the C command from the IOA online system) to order the table.

Use the O command in the OPT field on the Table List panel to order (activate) the table and the rules it contains. You should receive a confirmation message that the table has been ordered.
Implementing advanced JES support

The MainView AutoOPERATOR subsystem is always added to the subsystem vector table after the JES subsystem. Because the JES subsystem occurs in the vector table chain before MainView AutoOPERATOR, JES processes all commands and WTOs before MainView AutoOPERATOR.

However, this chapter describes what you need to do if you want to configure MainView AutoOPERATOR to

- modify or suppress JES commands
- modify or suppress WTOs on the JES job log

To accomplish this, you must edit and configure JESFLTR parameter in BBPARM member BBISSP00. When this parameter is used, MainView AutoOPERATOR installs a JES filter that allows MainView AutoOPERATOR to see message or command traffic before JES does.

Note
Editing and configuring the JESFLTR is required for both JES2 and JES3.

If you do not need to use MainView AutoOPERATOR to modify or suppress JES commands or WTOs, then BMC Software recommends that you omit entering this parameter in BBPARM member BBISSP00.

Specifying JESFLTR parameter values

Possible values for the JESFLTR parameter in BBPARM member BBISSP00 are as follows:
Specify the following to allow MainView AutoOPERATOR Rules to modify or suppress commands from JES consoles

**JESFLTR=CMD**

This specification affects all JES commands. JES commands are submitted to the Rules Processor with an event type of CMD.

Specify the following to allow MainView AutoOPERATOR Rules to modify or suppress messages from JES consoles

**JESFLTR=WTO**

JES messages are submitted to the Rules Processor with an event type of MSG before they are seen by JES.

Specify the following to allow MainView AutoOPERATOR Rules to modify or suppress both commands and messages from JES consoles

**JESFLTR=(CMD,WTO)**

### Implementing the JESFLTR parameter

You can implement the JESFLTR parameter by

- editing BBPARM member BBISSP00 and specify the JESFLTR= specification that you want

- using the BBI control command .SET to dynamically specify the option that you want; for example
  
  .SET JESFLTR=WTO
  
  or
  
  .SET JESFLTR=CMD
  
  or
  
  .SET JESFLTR=(CMD,WTO)

### Removing the JESFLTR setting

You can remove the JESFLTR setting whether the MainView AutoOPERATOR BBISS PAS is active or not.

To remove the JESFLTR setting, issue the MVS command UNHOOK prefixed with the BBISS PAS command character prefix defined by the CMDCHAR parameter in BBPARM member BBISSP00. The default command character is a period (.)

For example, issuing the MVS command:
.UNHOOK

removes the filter for this BBI-PAS. Messages are sent to the Log to show the filter has been removed.

Determining the status of the JESFLTR setting

To check the status of the JESFLTR hook at any time, use the following BBI control command:

.DISPLAY JESFLTR

When you issue this command, messages are sent to the BBI Journal that display the current status of the WTO and CMD filters as either ACTIVE or INACTIVE.

After you determine the current status of the JESFLTR setting you can do one of the following actions to change the status:

- Use the BBI control command .SET JESFLTR to activate or inactivate one or both of the filters.
- Use the MVS UNHOOK command to disable the filters.
- Perform a cold restart of the BBI-SS PAS.
  A cold restart of the BBI-SS PAS resets the JESFLTR setting based on the value of the JESFLTR keyword in BBPARM member BBISSPxx.

Additional implementation considerations

Do not restart JES while the filter is installed. If you restart JES while the filter is installed, one of two situations might happen:

- The JES restart might fail.
- MainView AutoOPERATOR might not be able to issue any JES messages or commands.

To prevent this from happening

1. Issue the UNHOOK MVS command
2. Restart (or hotstart) JES
3 Reestablish the filter by either method described in “Implementing the JESFLTR parameter” on page 224.

You should note that this process affects only the current MainView AutoOPERATOR BBI-SS PAS and therefore must be completed for each MainView AutoOPERATOR BBI-SS PAS that has implemented the JESFLTR parameter.
Implementing Continuous State Manager

This chapter provides information to help you implement Continuous State Manager (CSM) and attain the best possible performance.

Performance considerations

The Continuous State Manager application is not available in MainView AutoOPERATOR PASes that use the Rules Management application. For more information about Rules Management, see Chapter 3 in this document.

For the best performance results, add both CSM EXEC libraries, prefix. BBPROC and prefix. BBCLIB, to the Virtual Lookaside Facility (VLF). Refer to the IBM publication *TSO Customization Guide* for more information about VLF.

Define CSM EXECs (CSMUP, CSMDOWN, CSMBUILD, and CSMINIT) as high-priority EXECs in BBPARM member AAOEXP00. See “Continuous operation with the Dynamic Parameter Manager” on page 69, for more information about AAOEXP00 and high-priority EXECs.

Initiating CSM

The following sections describe how to initiate CSM.

Overriding the default group when CSM is initiated

In CSM, a group list is a collection of objects designated to run on a particular CSM partner. When you are defining objects to CSM, you must identify the group to which the object will be a member.
By default, the object belongs to the group named after the subsystem ID (SSID) of the BBI-SS PAS where CSM runs. You can override the group by specifying a different group name on the CSMALTDB parameter in the z/OS start command for the BBI-PAS. Refer to “Subsystem (SS) JCL parameter CSMALTDB” on page 228 for more information.

However, the default group cannot be overridden when CSM is initialized for the first time. To use a group other than the default, CSM must first be initialized with the default group. Subsequently, a group record must be created during a CSM terminal session. Once a group record is created, it can be specified on the CSMALTDB parameter to override the default group during future initializations.

**Subsystem (SS) JCL parameter CSMALTDB**

The SS JCL contains the CSMALTDB parameter. This parameter can be used to specify the name of a CSM group that is used by CSM when it initializes, or you can specify CSMALTDB=MINIMAL, which enables you to access the CSM TS to edit objects without CSM actually controlling the tasks.

---

**Note**

When you modify the CSMALTDB parameter, you must specify `START=COLD,VPOOL=RESET` for the change to take effect unless the new parameter value is the same as the CSMALTDB value from the previous start of MainView AutoOPERATOR. If the CSMALTDB parameter values are the same, and you have not IPL'd just prior starting MainView AutoOPERATOR address space, then you can specify a warm start: `START=WARM,VPOOL=RESET`.

---

An example of specifying the group name on the CSMALTDB parameter follows:

```
CSMALTDB=SYSB
```

where **SYSB** is the name of an SSID that is not local to the BBI-SS PAS that is starting CSM.

**End-of-Memory initiated CSM EXEC (CSMEOM)**

Normally, object termination can be monitored by using standard z/OS and JES messages. However, for some situations, monitoring based on these messages is not sufficient because an object might terminate without producing the expected messages.

Use the CSM EOM (CSM End-of-Memory) EXEC to ensure that critical objects do not terminate unnoticed. The CSMEOM EXEC allows CSM to monitor object termination regardless of how the object is terminated.
CSMEOM is designed to be called from an IMFEOM EXEC with the ASID and status being passed to the EXEC. CSM detects objects that terminate at End-of-Memory by using the CSMEOM EXEC.

See the *MainView AutoOPERATOR Advanced Automation Guide* for more information about EOM EXECs.

### Required security resources

Because CSM processing uses several MainView SYSPROG Service commands, the MainView AutoOPERATOR PAS requires READ authorization to access MainView SYSPROG Services security resources.

CSM uses the following MainView SYSPROG Services:

- PROGRESS
- STATUS
- ENQUEUES

The following example shows these resource names. The asterisk (*) represents the one- to four-character subsystem ID of the BBI-SS PAS where the resource is secured:

<table>
<thead>
<tr>
<th>Resource Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBM.*.RESOLVE.PROGRESS.LIST</td>
</tr>
<tr>
<td>BBM.*.RESOLVE.STATUS.LIST</td>
</tr>
<tr>
<td>BBM.*.RESOLVE.ENQUEUES.LIST</td>
</tr>
</tbody>
</table>

You must also ensure that the MainView AutoOPERATOR PAS has READ authority to the INIT.UPDATE resource if MainView SYSPROG Services security specifies INIT=YES on the TYPE=SYSPROG statement in the BBSEC data set member.

The following example shows this resource name:

<table>
<thead>
<tr>
<th>Resource Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBM.*.RESOLVE.INIT.UPDATE</td>
</tr>
</tbody>
</table>

For more information about MainView SYSPROG Services security resources, see the *MainView Security Reference*.

### Accessing CSM for the first time

This procedure describes accessing CSM for the first time.
1 Obtain and enter a valid key for the MainView AutoOPERATOR for z/OS option. Refer to the Installation System User Guide for information about how to install and activate passwords.

The MainView AutoOPERATOR for z/OS option is required for CSM processing, which includes issuing both z/OS commands and SYSPROG services.

2 In BBPARM member AAOPRM00, specify that the RULESET parameter includes the AAORULBC, AAORULBD, AAORULBE and AAORULBG Rule Set suffixes; for example:

```
RULESET=(BC,BD,BE,BG)
```

These Rule Sets are shipped with the high-level qualifier BBPARM and are required for CSM to function.

Do not use any of the Rule Sets from your previously installed version. Use only the AAORULBC, AAORULBD, AAORULBE, and AAORULBG Rule Sets that came with this release.

In addition, CSM also requires other Rule Sets to function that must be added to this parameter. CSM requires you to create at least one group and requires each group to have a Rule Set associated with it. Once the group Rule Sets are created, add the Rule Set suffix to the RULESET parameter.

3 In BBPARM member AAOPRM00, set the RULESCAN parameter to I for INDIVIDUAL.

The RULESCAN parameter controls the global automation strategy for all MainView AutoOPERATOR Rules. For more information about Rules and automation strategy, refer to the MainView AutoOPERATOR Basic Automation Guide, Volume 1: Using Rules.

4 In BBPARM member AAOEXP00, enter the names of all CSM EXECs and ensure that these EXECs are defined as High Priority. The names of the CSM EXECs are listed in the AAOEXP00 member.

5 Create a Shared Repository data set to contain all defined CSM objects and schedules.

If the IBM library for SAA REXX/370 (program number 5695-014) is not available, customize the BBREXALT JCL found in BBSAMP to run on your system. Submit BBREXALT to install the REXX/370 alternate library. Add the BBLINK to the STEPLIB concatenation of the INITCSM batch job. If you do not run this job, the compiled REXX object in INITCSM will fail.
Create the repository data set by running the batch job contained in BBSAMP member named INITCSM. For more information, refer to "Creating a shared repository data set for CSM" on page 233.

6 In BBPARM member BBIVARxx, specify the name of the Shared Repository.

Use the BBPARM member BBIVARxx to define a preset shared variable for CSM. The SHARED variable that must be present for CSM to function properly is

\[ \text{CSMREPDS} = \text{hilevel.datasetname} \]

where

- \text{CSMREPDS}: is the name of a shared variable that must be present for CSM to function correctly
- \text{hilevel.datasetname}: is the name of the Shared Repository data set created in “Accessing CSM for the first time This procedure describes accessing CSM for the first time.”

**Note**

If you are implementing CSM in an environment where the MainView AutoOPERATOR subsystems have JESFLTR parameters specified in the BBISSPxx member, and you are running more than one CSM system on an LPAR, you need to add a line in the BBIVARxx member: \text{CSMMDSPL = 'Y'}

This action creates a shared variable that the CSM Rule Sets use to determine whether certain communications messages are suppressed or not.

7 Start the BBI-SS PAS and specify \text{START=COLD,VPOOL=RESET}

When the cold start is complete, the BBI Journal should contain the following message:

\[ \text{ACM003I CSM initialization complete} \]

By completing these steps, CSM automatically initializes at BBI-SS PAS startup when message PM0010I is issued.
Migrating from a previous version of MainView AutoOPERATOR

If you are migrating from a previous version of MainView AutoOPERATOR (either MainView AutoOPERATOR 4.1 or 5.1) and you want to migrate your database repository, you must complete the following steps to convert your database:

1. Start the BBI-SS PAS and specify \texttt{START=COLD,VPOOL=RESET}.

2. Follow the instructions in “Migrating a CSM database from a previous release” on page 235.

Creating a group

Use the following procedure to create a group.

1. If you are not migrating from a previous release, log on to a terminal session and access CSM. You must add a group before you can begin adding object records. Refer to MainView AutoOPERATOR Solutions Guide for more information about adding groups and objects.

2. After you have either created a new group or converted all your MainView AutoOPERATOR 4.1 or 5.1 database repositories, enable the newly created groups with the Enable line command on the CSM Object Groups panel.

\textit{Note}

When a converted group becomes enabled, the statistics about the objects are offloaded to the data set specified in the Data set Name field, as shown in Step 4 on page 236.

After the data is offloaded, it is available in comma-separated format. Figure 33 on page 232 shows an example of what the contents in the data set might look like:

\textbf{Figure 33: Example of Offloaded Data}

\begin{verbatim}
BROWSE BAOKM2,KMZ1.OFFLOAD Line 00001813 Col 001 080
Command ===> Scroll ===> PAGE
P,AAOCM22,11:46:47.764226,2000-07-25,SJSD,KMZ1,KMZ1
P,AAOCM58,11:46:48.320489,2000-07-25,SJSD,KMZ1,KMZ1
S,AAOCM27,11:46:51.367131,2000-07-25,SJSD,KMZ1,KMZ1
S,AAOCM55,11:46:53.437561,2000-07-25,SJSD,KMZ1,KMZ1
S,AAOCM59,11:46:53.626765,2000-07-25,SJSD,KMZ1,KMZ1
P,AAOCM17,11:47:43.247170,2000-07-25,SJSD,KMZ1,KMZ1
S,AAOCM01,11:47:45.915276,2000-07-25,SJSD,KMZ1,KMZ1
*******************************************************************************
\end{verbatim}

Data in a comma-separated format can be uploaded to popular spreadsheet applications, such as Microsoft Excel, where it can become input to a variety of reports.
Creating a shared repository data set for CSM

--- Note ---
The information contained in this section is part of Step 5 as described in “Accessing CSM for the first time” on page 229. This section contains a set of procedures that you must complete before using CSM for the first time.

Use the BBSAMP member INITCSM to create and initialize a Shared Repository data set that will contain all defined CSM objects and schedules.

**To customize this job stream**

1. Modify the jobcard to installation standards.

2. Make the following global JCL and SYSIN changes:
   a. Change *HILEVEL* to the high-level qualifier for BMC Software product data sets.
   b. Change *SSID* to the four character subsystem (SSID) name.
   c. Change *VOL* to the volume serial number of the DASD volume on which the linear data set is to be allocated.

3. Submit the newly modified job stream.

After the job stream has run successfully, check for zero return codes for both steps. Additionally, the SYSTSPRT DD should contain output messages. The message **CSM REPOSITORY SUCCESSFULLY INITIALIZED** should appear at the end of this data set.

--- Note ---
The space allocations shown in Figure 34 on page 233 should be enough for over 2,000 objects on most device types. For more details about space allocation, refer to “Calculating repository size for a 3390” on page 234 on page 249.

**Figure 34: BBSAMP member INITCSM Sample JCL**

```plaintext
//INITCSM  JOB ACCOUNT ** MODIFY JOB CARD TO INSTALLATION STANDARDS **
//**-------------------------------------------------------------------*
//** THIS JCL CREATES AND INITIALIZES THE DATA REPOSITORY USED        *
//** BY CSM. THE CSM REPOSITORY IS A KEYED VSAM DATASET WHICH          *
//** CAN BE SHARED BY ALL MEMBERS OF A CSM-PLEX.                      *
//**-------------------------------------------------------------------*
//** INITCSM PROC HILEVEL='*HILEVEL*'.                                 *
//** SSID='*SSID*'.                                                     *
//** SYSOUT=**                                                           *
```
Calculating repository size for a 3390

The repository has a CI size of 14,336. Each physical block of data is 7,168 bytes in length. Each track on a 3390 device can hold 7 physical blocks of data. Each record in the repository has a length of 14,237 bytes. Each cylinder on a 3390 device holds 15 tracks of data.

For example, if you have thirty groups, one master record and eighteen objects you would have 139 to store. The calculation is shown in Figure 35 on page 234:

**Figure 35: Calculating Repository Size**

<table>
<thead>
<tr>
<th>(30 groups x 4 records)</th>
<th>+</th>
<th>(18 objects x 1 record)</th>
<th>+</th>
<th>(one master record)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(1)</td>
<td></td>
<td>(2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(3)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

where

- each group takes four repository records
- each object takes one repository record
- the master takes one repository record

It takes two physical blocks to hold one repository record. Therefore, each track can hold three and one half repository records. If you have three and one half repository records per track, you need 39.14 tracks to store 137 records. To store 39.14 tracks, you need 2.60 cylinders.

39.14 tracks / 15 tracks per cylinder ~= 2.60 cylinders of data
Migrating a CSM database from a previous release

This procedure is required only if you are migrating from MainView AutoOPERATOR version 5.1.xx or earlier. If you are migrating from MainView AutoOPERATOR 6.1.xx or later, you only need to use the REPRO command to copy the CSM repository to a new repository that the newer release of CSM PAS will use.

**Note**

The information contained in this section is part of the set of procedures described in “Accessing CSM for the first time” on page 229, which you must complete before using CSM for the first time.

To migrate a CSM database from a previous release

1. Log on to a terminal session and access the CSM Global Overview panel (as shown in Figure 36 on page 235).

![Figure 36: CSM Global Overview panel](image)

2. On the COMMAND line, enter the CONVERT primary command. The CSM Conversion panel is displayed (see Figure 37 on page 235).

![Figure 37: CSM Conversion panel](image)
3 Follow the instructions on the panel.

   a In the first **Data Set Name** field, enter the name of the CSM data set that you are replacing.

   b In the **Member** field, enter the member name (for example: CSM ssid).

   c In the second **Data Set Name** field, specify the name of a data set where the conversion process logs information that is related to the conversion.

   The data set may be a preallocated RECFM=FB, LRECL=80 data set or the data set may be dynamically allocated with these DCB attributes: RECFM=FB, LRECL=80, BLKSIZE=23680.

   d In the **Disposition** field, specify one of the following choices:

   - MOD causes the Routine to append data to the end of the sequential data set.
   - SHR causes the Routine to write data at the beginning of the data set.
   - The resulting Conversion Log contains information such as the conversion date and time, user ID, the input database name, the Group name, and a list of the converted objects and exceptions.
   - After you enter this information, the Group Detail Control panel is displayed.

4 On the Group Detail Control panel, enter information related to the CSM group that is being converted.

   **Note**

   You must add the Rule Set ID that you specify in the **Rule Set ID** field on this panel to the RULESET parameter in BBPARM member AAOPRM00 (refer to Step 2 on page 235).

   Figure 38 on page 236 shows an example of a completed Group Detail Control panel.

   **Figure 38: Completed Group Detail Control panel**

<table>
<thead>
<tr>
<th>BMC Software</th>
<th>Group Detail Control - Add</th>
<th>AutoOPERATOR TGT KMZ1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Name</td>
<td>SYSB</td>
<td></td>
</tr>
<tr>
<td>Command Prefix</td>
<td>CSMKM</td>
<td></td>
</tr>
<tr>
<td>Rule information:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ruleset Dataset</td>
<td>BAOKMZ.AAO61.KMZ1.UBBPARM</td>
<td></td>
</tr>
<tr>
<td>Ruleset Id</td>
<td>G2</td>
<td>Rule Prefix ACM1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Suffix 0001</td>
</tr>
</tbody>
</table>
Object Statistics:
Dataset Name      ===> BAOKMZ.KMZ1.OFFLOAD
Disposition       ===> MOD

Application information:
Author            ===> BAOKMZ2
Description       ===> GROUP LIST FOR SYSB

Enter END command to process and return or CANCEL to leave

The default name of the CSM group is created from the last four characters of the CSMssid member that is being converted. You can choose your own group name by typing over the default name. If the name that you choose already exists, a pop-up window is displayed, asking for a new Group Name.

5 Enter a new Group Name in the pop-up window (Figure 39 on page 237) or enter CANCEL to terminate the conversion.

**Figure 39: Pop-up window for new Group name**

Enter a new Group Name in the pop-up window (Figure 39 on page 237) or enter CANCEL to terminate the conversion.

6 As the conversion progresses, a pop-up window (Figure 40 on page 237) is displayed and as every five objects are converted, the counter is updated.

**Figure 40: CSM Conversion Status pop-up window**

7 When the conversion is complete, the CSM Conversion Log is displayed (Figure 41 on page 237).

**Figure 41: CSM Conversion Log**

<table>
<thead>
<tr>
<th>Objects</th>
<th>Exception Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAOCSMN1</td>
<td>Accessing CSM for the first time</td>
</tr>
</tbody>
</table>
During conversion, you might encounter conflicts or mismatches with object definitions. For more information about resolving object-definition mismatches, refer to “Resolving object-definition conflicts after converting” on page 238.

8 Press End from the CSM Conversion Log.

The converted CSM objects and newly created Group are displayed on the CSM Global Overview panel (Figure 42 on page 238).

Figure 42: Completed CSM data set conversion

9 Repeat Step 2 on page 235 through Step 8 on page 238 for each version 4.1 or 5.1 CSM data set and member name to be converted.

Resolving object-definition conflicts after converting

The conversion process converts every object into one record in the Shared Repository data set.

Because each object can have only one record within the Shared Repository, but be a member of more than one group, you might encounter object-definition conflicts. If a conflict is encountered, it is identified on the CSM Conversion Log data set (“Migrating a CSM database from a previous release” on page 235).

An object-definition conflict can occur when an object already has a record in the Shared Repository and the object is being converted as a member of a different group with different definitions (see Table 14 on page 239).

Table 14 on page 239 shows all possible definitions for an object. The first column contains the object definitions that can be different for each group. The second
column contains the object definitions that must be identical for all groups containing this object.

Table 14: Object definitions

<table>
<thead>
<tr>
<th>Do not have to match in the repository</th>
<th>Must match within the repository</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parents</td>
<td>• Object Type</td>
</tr>
<tr>
<td>Client</td>
<td>• Restart Only</td>
</tr>
<tr>
<td>Server</td>
<td>• Verify Force Down</td>
</tr>
<tr>
<td>Schedules</td>
<td>• Start Command/EXEC</td>
</tr>
<tr>
<td></td>
<td>• User Post Start EXEC</td>
</tr>
<tr>
<td></td>
<td>• Stop Command/EXEC</td>
</tr>
<tr>
<td></td>
<td>• User Post Stop EXEC</td>
</tr>
<tr>
<td></td>
<td>• Cancel Command/EXEC</td>
</tr>
<tr>
<td></td>
<td>• Recovery Command/EXEC</td>
</tr>
<tr>
<td></td>
<td>• Start Command Time Out</td>
</tr>
<tr>
<td></td>
<td>• Stop Command Time Out</td>
</tr>
<tr>
<td></td>
<td>• Start Command Limit</td>
</tr>
<tr>
<td></td>
<td>• Start Event Text ID and Type</td>
</tr>
<tr>
<td></td>
<td>• Stop Event Text ID and Type</td>
</tr>
<tr>
<td></td>
<td>• Abnormal Termination Events</td>
</tr>
<tr>
<td></td>
<td>• TSO Send User ID</td>
</tr>
<tr>
<td></td>
<td>• Pager Contact</td>
</tr>
<tr>
<td></td>
<td>• Pager Info</td>
</tr>
<tr>
<td></td>
<td>• Alert Queue</td>
</tr>
<tr>
<td></td>
<td>• Rules Generation</td>
</tr>
</tbody>
</table>
You can resolve a conflict by starting CSM and by using the CSM Object Definition panel to edit each object and ensure that the object definitions for each object in every group match exactly (with the exception of the object’s Parents, Clients, Servers and Schedules).

If you do not want each object in every group to have identical definitions, you can also resolve a conflict by using variables in CSM object definitions that can resolve to different values for the different groups on different BBI-SS PASs. The definition can contain MainView AutoOPERATOR SHARED variables. For more information, refer to the section "Using Variables in the Object Definition Fields" in the MainView AutoOPERATOR Solutions Guide.

Another way to resolve a conflicts is to place the new group into a different Shared Repository data set.

**Sample JCL for copying the shared repository**

The following figure shows an example of JCL that you can use to copy the Shared Repository data set into a new data set. You can use this sample JCL when you encounter the need to copy your Shared Repository data set into a larger data set.

Replace the data set names BAONIS.CSM.DATABASE and BAONIS.CSM.ALTDB with your data set names.

**Figure 43: JCL example for copying Shared Repository**

```java
//JOBCARD
//CPY   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//F1       DD DSN=BAONIS.CSM.DATABASE,DISP=SHR <== Current Repository data set
//F2       DD DSN=BAONIS.CSM.ALTDB,DISP=SHR <== New Repository data set
//SYSIN    DD *
REPRO IFILE(F1) OFILE(F2)
```

Using the AATWAIT program for testing Continuous State Manager or the MainView Total Object Manager

MainView AutoOPERATOR distributes the AATWAIT program in `hilevel.BBLINK` that you can use to easily create test Started Tasks (STCs). This is a small program that consumes minimal resources.

Having the capability to easily and quickly create a set of test STCs can be helpful when you want to test proposed changes prior to making the changes to the Continuous State Manager or MainView Total Object Manager applications. By
using the AATWAIT program, you can verify that your changes have the desired effect without affecting the production level objects.

After you invoke AATWAIT, it runs for a specified length of time; the default is forever. You can also control these aspects of the program:

- specify stopping the program after a specified number of minutes or seconds, which you set with the RUN(minutes) parameter (for minutes) or the RUS(seconds) parameter (for seconds)

- specify stopping the program after a specified number of minutes or seconds and issuing a specified abend code. To specify the abend code and the time span, use ABENDU(code) with either RUN(minutes) or RUS(seconds).

- stop the program prematurely by issuing the MVS Stop command for the address space; it will stop either normally or with the user abend code that you specified with the ABENDU(code) parameter.

- specify a number of seconds that simulates a task startup delay, after which a startup completed message (AA2107I) is issued to confirm that the task is ready

- specify a number of seconds that simulates a task shutdown delay (which issues message AA2113I), followed by message AA2108I to confirm that the task has shut down normally.

- issue a MVS Cancel command to cause this program to terminate with a system abend code such as abend S222

These capabilities allow you to make the test STCs simulate real-world conditions. When invoking the AATWAIT program, you can use any of the following optional parameters:

- RUN(minutes) specifies when to stop the program, in number of minutes. Replace minutes with any value from 0 through 999999. RUN(0), the default, tells the program to run without stopping. Valid values are 0 through 999999 minutes.

- RUS(seconds): specifies when to stop the program, in number of seconds. If not specified, the value of the RUN(minutes) is used. Replace seconds with 0 or with any value from 2 through 999999. RUS(0), the default, tells the program to run without stopping. You cannot specify 1 second.

Note

The RUN() and RUS() parameters are mutually exclusive; you can specify only one, or you can omit both.
- **ABENDU**(code) specifies that the program issues the specified abend code when stopped. Valid values for code are NONE or any abend code from 0 through 4095. The default, NONE, means the program ends normally.

- **WSTART**(seconds) specifies the number of seconds between when the task starts and when it issues the startup complete message (AA2107I). Replace seconds with any value from 1 through 999999. Choose a value that simulates the processing time for your task startup, after which the task is ready. If you omit this parameter, the task starts after a 5-second delay (the default) and does not issue message AA2107I. To receive AA2107I, you must specify this parameter.

- **WSTOP**(seconds) specifies the number of seconds between the start of task termination and its completion. Replace seconds with any value from 1 through 999999. Choose a value that simulates the shutdown processing time for a task. If you specify this parameter, the task issues message AA2113I when shutdown starts and AA2108I when it completes. In contrast, if you omit this parameter, the task stops immediately upon receipt of a STOP command, or when the time specified by RUN(minutes) or RUS(seconds) has elapsed; in this case, no messages are issued.

One STC procedure in PROCLIB should suffice no matter how many test objects you need to create. Customize the sample JCL shown in Figure 44 on page 242 with information from your installation environment.

**Figure 44: Sample JCL**

```jcl
//WAIT PROC
//   RUN=, run time (minutes)
//   RUS=, run time (seconds)
//   ABEND=, user ABEND code
//   WSTART=, start delay (seconds)
//   WSTOP=, shutdown delay (seconds)
//*
//* AutoOPERATOR 7.3.00
//* Program for AutoOPERATOR CSM and TOM testing
//*
//* RUN = default is zero; means run forever
//*   = RUN for 0 - 999999 minutes
//*   = RUN and RUS cannot both be specified at the same time
//*
//* RUS = similar to RUN but specified in seconds
//*   = RUS(0) is 'run forever'
//*   = RUS(nnnnnnn) supports a range of 2-999999 seconds
//*   = if not specified, the value for RUN is used
//*   = RUN and RUS cannot both be specified at the same time
//*
//* ABEND = default is none; means terminate normally
//*   = specify a number to stop with user abend nnnn (0 - 4095)
//*
//* WSTART = default is none; means no start up processing delay
//*   = valid values are 1-999999
//*   = when specified, message AA2107I is issued
//*
//* WSTOP = default is none; means no shutdown processing delay
//*   = valid values are 1-999999
```

242  MainView AutoOPERATOR Customization Guide
For example, assuming the PROCLIB member is called AAO, you can start multiple STCs with any desired jobnames as shown in these examples:

```plaintext
S AAO,RUN=100
S AAO.AAOTEST1,ABEND=0
S AAO.JOBNAME=TEST2
S AAO.JOBNAME=TEST3,RUN=3
S AAO.JOBNAME=TEST4,ABEND=5
S AAO.JOBNAME=TEST5,RUN=5,ABEND=6
```

In the example shown in Figure 45 on page 243, the option to terminate the program with user abend code 1 specified. WSTART and WSTOP simulate task startup and shutdown processing delays.

If WSTART is not specified, message AA2101I may be viewed as a typical ‘ready for work’ message. This message is issued 5 seconds after the program starts, to simulate a startup delay.

This STC was manually stopped with an MVS STOP command, and the resulting message is AA2102I. Because of the WSTOP value, the abend is delayed until after the AA2108I message is issued. If a value for WSTOP is not specified, the abend occurs immediately after the STOP command is received.

**Figure 45: Partial sample JES listing**

```
06.41.39 STC01251 ---- TUESDAY, 17 APR YYYY ----
06.41.39 STC01251 IEF695I START AAODL WITH JOBNAME AAODLTST IS ASSIGNED TO USER
06.41.39 STC01251 $HASP373 AAODLTST STARTED
06.41.39 STC01251 IEF403I AAODLTST - STARTED - TIME=06.41.39
06.41.39 STC01251 +AA2104I RUN FOR 3 MINUTES
06.41.39 STC01251 +AA2109I WSTART FOR 10 SECONDS
06.41.39 STC01251 +AA2110I WSTOP FOR 10 SECONDS
06.41.39 STC01251 +AA2101I ISSUE 'P AAODLTST' TO STOP THIS STC (PTF: BPI0000)
06.41.49 STC01251 +AA2107I STARTUP OF AAODLTST HAS COMPLETED; TASK READY
06.41.49 STC01251 +AA2105I STIMERM FOR 00000179 SECONDS
06.41.59 STC01251 +AA2102I STOP COMMAND ACCEPTED FROM CONSOLE BAODDL2
06.41.59 STC01251 +AA2113I SHUTDOWN OF AAODLTST COMMENCING
06.42.09 STC01251 +AA2108I SHUTDOWN OF AAODLTST HAS COMPLETED
06.42.09 STC01251 IEF450I AAODLTST - ABEND=S000 U0001 REASON=00000000
   TIME=06.42.09
111 06.42.09 STC01251 -JOBNAME STEPNAME PROCSTEP RC EXCP CONN TCB SRB
   06.42.09 STC01251 -AAODLTST STEPD01 U0001 5 2 .00 .00
   06.42.09 STC01251 -AAODLTST ENDED. NAME=
   06.42.09 STC01251 IEF404I AAODLTST - ENDED - TIME=06.42.09
   06.42.09 STC01251 $HASP395 AAODLTST ENDED
----- JES2 JOB STATISTICS -----
```
Using the AATWAIT program for testing Continuous State Manager or the MainView Total Object Manager
Specific MainView AutoOPERATOR parameters and JCL

This chapter contains the following information regarding specific MainView AutoOPERATOR parameters and JCL.

Enabling REXX support for EXECs

The REXX EXEC environment is disabled when MainView AutoOPERATOR is initially installed and activated. To enable REXX support, perform the following action and restart your BBI-SS PAS:

Add the following two additional DD statements to the BBI-SS PAS JCL

```
//SYSTSIN DD DUMMY
//SYSTSPRT DD SYSOUT=s
```

where `s` is a SYSOUT class of your choice.

Detecting EXEC loops

Use the following parameters in BBPARM member AAOEXP00 to limit the CPU utilization and time that an EXEC may use. These parameters are effective only on MVS/SP2 systems and later.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIMEXLIM=nnnn</td>
<td>(optional) specifies the CPU time limit for an EXEC</td>
</tr>
<tr>
<td></td>
<td>If this time is exceeded, the EXEC is terminated with a USER 3001 abend code.</td>
</tr>
</tbody>
</table>
**Parameter** | **Description**
---|---
**PEREXLIM=nn** | *(optional)* specifies the CPU usage threshold in an EXEC in percent. If an EXEC CPU usage exceeds this percentage during an interval of 15 seconds, the EXEC is terminated with a USER 3001 abend code.

*Note:* Usable CPU percentage is 100 percent. This amount includes multiprocessor machines.

Refer to “Continuous operation with the Dynamic Parameter Manager” on page 69 for information about using the Dynamic Parameter Manager application to change these parameters. Also, refer to the *MainView AutoOPERATOR Advanced Automation Guide* for more information about performance issues and using EXECs.

## Prioritizing and threading of EXECs

You can modify BBPARM member AAOEXP00 so your system
- allows multithreading of EXECs and allows EXECs to run concurrently
- gives priority to certain EXECs

You can modify the following BBPARM member AAOEXP00 parameters:

<table>
<thead>
<tr>
<th><strong>Keyword</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EXEC=xxxxxxx</strong></td>
<td>enter the names of EXECs that you want to execute a high priority</td>
</tr>
<tr>
<td><strong>MAXNORM=nn</strong></td>
<td>enter the maximum number of normal priority EXECs that you want to execute concurrently. A normal priority EXEC is any EXEC not in the high priority list. Range is 1 to 99. <em>Note:</em> This value is limited more by BBI-SS PAS memory requirements than by the 99 MAX range.</td>
</tr>
<tr>
<td><strong>MAXHIGH=n</strong></td>
<td>enter the maximum number of priority EXECs to execute concurrently. A high priority EXEC is any EXEC listed with the EXEC= keyword in AAOEXP00. Range is 1 to 99.</td>
</tr>
</tbody>
</table>

Refer to “Continuous operation with the Dynamic Parameter Manager” on page 69 for information about using the Dynamic Parameter Manager application to change these parameters. In addition, refer to the *MainView AutoOPERATOR Advanced Automation Guide* for more information about performance issues and using EXECs.
Defining the message counter table

MainView AutoOPERATOR monitors and counts all messages from different sources in the Rule Processor application and displays the results on the Event Activity Statistics panel.

To determine the total number of unique messages to be displayed on the message activity statistics panel, specify in BBPARM member BBISSP00

```
MCTSIZE=xxxx
```

This action defines the size of the event counter table by specifying the number of event to be maintained in storage. The value range is from 1 to 99999; the default is 510. Specify 0 to disable all unique event counting. For each unique event, 128 bytes of ECSA are used to maintain information.

The difference between the total count of events and the unique count of event on the Event Activity Statistics panel is shown in the following example.

If you want to monitor three different types of events such as

- $HASP309
- $HASP310
- $HASP311

the total number of unique events appears on the Event Activity Statistics panel in the **UNIQUE** field as

```
UNIQUE=3
```

Each unique event can have up to 20 different associated activities; for example:

<table>
<thead>
<tr>
<th>$HASP309</th>
<th>INIT</th>
<th>4</th>
<th>ACTIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$HASP309</td>
<td>INIT</td>
<td>2</td>
<td>INACTIVE</td>
</tr>
<tr>
<td>$HASP309</td>
<td>INIT</td>
<td>4</td>
<td>INACTIVE</td>
</tr>
</tbody>
</table>

The total number of the activities associated with the three events appears on the Event Activity Statistics panel as

```
Total messages: 60
```
Displaying MVS JES2 job numbers in address space application

If you have the MVS option for MainView AutoOPERATOR and you want to display JES2 job numbers and job class in the address space application, submit the JCL provided in BBSAMP member JES2MAO. This job reads the user’s SYS1.HASPSRC data set to retrieve relevant JES2 offsets.

If you do not run this job, the warning message AAO229W is issued at BBI-SS PAS startup and the address space application does not show JES2 job numbers or their classes.

Starting BBI-SS PAS before JES

If you want MainView AutoOPERATOR to automate functions that are normally performed before JES is operational, the MainView AutoOPERATOR subsystem must be started before JES.

To start the BBI-SS PAS before JES

1. Allocate a minimum of three dump data sets with RECFM=VBA and LRECL=125,BLKSIZE=1632,DSORG=PS. Make each one large enough to accommodate the largest dump you expect.

2. Add the following allocations to your BBI-SS PAS JCL, pointing to the dump data sets created in the previous step.

   //BBDUMP01 DD DISP=SHR,DSN=dumpdsn1,VOLSER=....,UNIT=....
   //BBDUMP02 DD DISP=SHR,DSN=dumpdsn2,VOLSER=....,UNIT=....
   //BBDUMP03 DD DISP=SHR,DSN=dumpdsn3,VOLSER=....,UNIT=....

BBI-SS PAS opens the first DUMP data set when the BBI-SS PAS starts. This data set closes and the next one opens after an abend occurs. When the last one has been opened and used, it will be reopened and reused over and over until dumps are reset using the BBI control command .RESET DUMP. You can issue this command from a terminal session, the z/OS console, or an EXEC in the standard BBI command format.
These dump data sets are used when a BBI FORMATTED dump is taken that is controlled by the DUMPS= parameter in BBPARM member BBISSP00. If DUMPS=YES or DUMPS=BOTH is set in BBPARM member BBISSP00, a BBI FORMATTED dump is taken. If DUMPS=SDUMP is specified, a SVC dump is taken. These SYSUDUMP data sets are not used although they should always be allocated.

3 Catalog all data sets referenced in the SSJCL PROC in the master catalog or change all allocations to UNIT/VOLSER allocations.

4 Add a TIME= parameter to the EXEC statement in BBSAMP member SSJCL.

Code the TIME= parameter as required for your installation.

TIME=1440 is recommended.

5 Change the following two DD statements in the BBI-SS PAS JCL:

```
//SYSTSSIN DD dummy
//SYSTSPRT DD dummy
```

**Note**
The terminal session (TS) can be started only after JES has initialized.

6 Start the BBI-SS PAS with the command:

```
S SSJCL.SSID,SUB=MSTR
```

**Restrictions**

If the MainView AutoOPERATOR BBI-SS PAS is started before JES, you cannot use any command or program that attempts to allocate to a SYSOUT data set.

If you set JESCNCT=YES in BBPARM member BBISSP00, you can use the IMFEXEC commands JESALLOC and JESSUBM to allocate and submit jobs.

**IMFJNUM option and seven-digit JES job numbers (z/OS version 1.2 and later)**

On z/OS v1.2 and later, JES2 and JES3 can have job numbers greater than 65,534. JES2 and JES3 can operate in either the old "five-digit" mode or the new "seven-digit"
mode. With this support, you can have numbers up to 999,999 with room to expand one digit further.

For more information about JES2, refer to the $ACTIVATE and $T JOBDEF,RANGE=(xx,yy) commands. For more information about JES3, refer to the OPTIONS initialization statements and review the JOBNO= specification. More information is also available in the JES2 and JES3 migration manuals.

The following examples show the old five-digit format (representative, maximum, minimum):

JOB12345
STC65534
TSU00001

The following examples show the new seven-digit format (representative, maximum, minimum):

T0123456
J0999999
S0000001

You must set the AAOPRMxx option IMFJNUM=V to support job numbers greater than 99,999. The default is IMFJNUM=V. The new setting will be effective after the next warm start of the BBI-PAS. New customers should specify the AAOPRMxx option IMFJNUM=V immediately.

When the IMFJNUM option is not specified or is set to 5, the following statements are true:

- The variable IMFJNUM, which is available to both Rules and Rule-Initiated EXECs, is a fixed length five-digit field. The possible values are 1 to 99,999.
- IMFJNUM is set to null for events where the JES job number is greater than 99,999.
- A highlighted warning WTO (AU6020W) message is issued the first time MainView AutoOPERATOR detects a seven-digit mode job number that is between 1 and 99,999. This WTO message warns you that the seven-digit job number is incompatible with MainView AutoOPERATOR.
- A highlighted error WTO (AU6021E) message is issued the first time MainView AutoOPERATOR detects a job number greater than 99,999.
When the IMFJNUM option is set to V, the following statements are true:

- The variable IMFJNUM, which is available to both Rules and Rule-Initiated EXECs, is a variable length field without leading zeros.
- IMFJNUM=V supports all versions of JES.
- IMFJNUM=V supports both five-digit and seven-digit numbers.
IMFJNUM option and seven-digit JES job numbers (z/OS version 1.2 and later)
Accessing advanced SYSPROG services

MainView AutoOPERATOR is distributed with a set of basic SYSPROG commands (formerly called RESOLVE services). If you also own MainView for OS/390, you have the option to allow access to the Advanced SYSPROG services from MainView AutoOPERATOR and to execute the advanced early warning system (AEWS) in the MainView AutoOPERATOR address space.

The full set of Advanced SYSPROG services are available automatically to MainView AutoOPERATOR if you have MainView for OS/390 installed in the same SMP/E zones as MainView AutoOPERATOR or if you have the MainView for OS/390 data sets (both BBLINK and BBLOAD) defined in the linklist concatenation.

If neither of these conditions is met, you must make the following JCL modifications:

- Add the MainView for OS/390 BBLINK data set as the first data set on the STEPLIB DD statement in the SS JCL.
- Add the MainView for OS/390 BBLOAD data set as the first data set on the BBLOAD DD statement in the SS JCL.
- If you have combined the MainView for OS/390 BBLINK and BBLOAD data sets, make a reference to the combined data set on both STEPLIB and BBLOAD.

MainView AutoOPERATOR requires all data sets in the BBLOAD concatenation to be APF authorized. The MainView for OS/390 BBLOAD data set might not be authorized since MainView for OS/390 does not require this authorization. If the data set is not authorized, you must authorize the MainView for OS/390 BBLOAD data set before adding it to the BBLOAD concatenation.

Not all the Advanced RESOLVE commands function under MainView AutoOPERATOR. For example, the CONSOLE service does not function because it uses full screen support and the MainView AutoOPERATOR interface is based on line-mode.

The user security exit (ASTXA1UE) is invoked for each service invocation. Refer to the MainView AutoOPERATOR Options User Guide for more information about the user security exit.
Executing AEWS in the MainView AutoOPERATOR address space

To run the advanced early warning system (AEWS) in the MainView AutoOPERATOR address space, add the LIB DD statement (as described in the MainView SYSPROG Services User Guide and Reference) to the MainView AutoOPERATOR subsystem JCL.
Authorization for special MainView AutoOPERATOR functions

Depending on how a MainView AutoOPERATOR EXEC is scheduled, it is assigned a specific type of origin. The origin is checked against a matching USERID member before default security options are checked. Additional USERID members may be needed for the following origins:

<table>
<thead>
<tr>
<th>Target type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>target ID of the remote system</td>
<td>for an EXEC scheduled through a message from a target</td>
</tr>
<tr>
<td>subsystem ID of the remote BBI-SS PAS</td>
<td>for an EXEC scheduled from a BBI-SS PAS message or one that is time-initiated</td>
</tr>
<tr>
<td>jobname or RACF USERID</td>
<td>for an externally initiated EXEC</td>
</tr>
</tbody>
</table>

Command authorization

Users must be authorized to issue commands or use applications against a BBI-SS PAS target system. Command authorization is defined in user authorization members of the BBPARM data set.

A summary of authorization commands for MainView AutoOPERATOR follows:

For all MainView AutoOPERATOR options

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>authority to schedule EXECs</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MVSCMD</td>
<td>authority to issue MVS commands</td>
</tr>
<tr>
<td>RULEAUTH</td>
<td>Rules processor authorization which can be READ, UPDATE, or none</td>
</tr>
</tbody>
</table>

For the MainView AutoOPERATOR *for IMS* option only

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMSCMD</td>
<td>authority to issue IMS commands</td>
</tr>
<tr>
<td>IMSMSG</td>
<td>authority to send IMS messages to LTERM$s$</td>
</tr>
<tr>
<td>IMSTRAN</td>
<td>authority to issue IMS transactions</td>
</tr>
</tbody>
</table>

For the MainView AutoOPERATOR *for z/OS* option only

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESAUTH</td>
<td>allowed to issue authorized SYSPROG commands</td>
</tr>
</tbody>
</table>

For more information about issuing Advanced SYSPROG commands and the security exit, refer to the [MainView AutoOPERATOR Options User Guide](#).

A description of how to use these commands is in the [MainView Security Reference Guide](#). The comment field of the USERID member of the BBPARM data set also describes the commands and their valid parameters.
Implementing MainView AutoOPERATOR for CICS function security

This chapter discusses function security for MainView for CICS and MainView AutoOPERATOR for CICS, how it is specified by assembling and linking the CMRSECU module using the CMRSECU macro.

The sample security member is distributed in BBSAMP member CMRUSECU. Use the following procedure to update this member:

1. Copy member CMRASM from `prefix.BBSAMP` to `prefix.UBBSAMP`. Make sure the new member name conforms to site naming conventions.

2. Edit the member created in step 1. Read the comments in the member and then modify the JCL accordingly.

   **Tip**
   
   You can dynamically reset the CMRSECU table with the command: `.RESET PARM CMRSECU`
   
   See the *MainView Administration Guide* for a description of the RESET control command and supported parameters.

3. Modify the sample invocations of the CMRSECU macro according to your site’s security requirements.

   The CMRSECU macro can contain up to three types of statements:

   | TYPE=INITIAL | The CMRSECU TYPE=INITIAL statement defines security for each |
   |             | ■ MainView for CICS function or action service |
   |             | ■ MainView AutoOPERATOR for CICS action service |
The CMRSECU TYPE=ENTRY statement further defines security per operator ID. (For example, a specific function may be accessed but only by a specific operator or operators.)

A single CMRSECU TYPE=FINAL statement ends the input. Multiple sets of CMRSECU TYPE=INITIAL and CMRSECU TYPE=ENTRY statements can be specified. Each statement and its parameters are described in the sections that follow.

4 Submit the newly created JCL; check the output to insure that all steps are completed with a return code of 0.

**Note**

S106 ABENDs can occur if the newly link-edited module causes the BBLINK data set to enter secondary extents and the active BBI-SS PAS attempts to load the module. If this situation happens, stop BBI-SS PAS and restart it.

---

### CMRSECU TYPE=INITIAL statement (required)

The CMRSECU TYPE=INITIAL statement defines security for each

- MainView for CICS function or action service
- MainView AutoOPERATOR for CICS action service

To specify which target or targets the security definitions apply to, enter

```
TYPE=INITIAL,TARGET=[(targetname1,targetname2,...,targetnamen. | *)]
```

**Note**

The asterisk (*) indicates that security definitions apply to all nonspecified targets. If an * is not specified and a nonspecified target is referenced online, the first CMRSECU TYPE=INITIAL statement defines function security for that target. Refer to BBSAMP member CMRUSECU for an example.

To set function security, enter YES or NO for each function parameter:

- YES indicates that the function is protected. Only operator IDs with YES specified in the CMRSECU TYPE=ENTRY statement are allowed to access that function (see “CMRSECU TYPE=ENTRY statement by operator ID” on page 263).
- NO indicates that the function is not secured. All operator IDs allowed access to MainView for CICS or MainView AutoOPERATOR for CICS are also allowed access to the function.
If the function is protected and an unauthorized operator tries to access it, an error message is issued to the operator and logged.

Display parameters apply to MainView for CICS services; if you do not have MainView for CICS installed, the parameters in the Display Parameter list do not apply. Action parameters apply to MainView for CICS functions and MainView AutoOPERATOR for CICS IMFEXEC CICS commands. For IMFEXEC CICS syntax requirements, refer to the MainView AutoOPERATOR Advanced Automation Guide.

**Note**

For MainView for CICS, functions that display information and allow actions to be taken, two function parameters are listed: one to secure access to the display; the other to secure actions taken from the display. For example, the TASK function is both a display and an action screen. Specifying TASK=NO allows unlimited access to the TASK statistics and MEMORY displays only. Specifying ALTTASK=YES secures the action services related to tasks.

For the CMRSECU TYPE=INITIAL statement, the action parameters are listed in Table 17 and the display parameters are listed in Table 16 on page 261.

The default for each parameter is NO.

**Table 15: Action parameters for the CMRSECU TYPE=INITIAL and TYPE=ENTRY statements**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>MainView for CICS service description</th>
<th>Applicable Rule command or IMFEXEC CICS command</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTAID=NO</td>
<td>YES</td>
<td>specifies whether to secure the change capability for the Automatic Initiate Descriptor statistics display</td>
</tr>
<tr>
<td>ALTCLAS=NO</td>
<td>YES</td>
<td>specifies whether to secure the change capability for the classes display</td>
</tr>
<tr>
<td>ALTCONN=NO</td>
<td>YES</td>
<td>specifies whether to secure the status change capability in the CONNECT service The expanded displays are included.</td>
</tr>
<tr>
<td>ALTDEST=NO</td>
<td>YES</td>
<td>specifies the status change capability for the DEST service</td>
</tr>
<tr>
<td>ALTER=NO</td>
<td>YES</td>
<td>specifies whether to secure the change capability for the virtual storage address display</td>
</tr>
<tr>
<td>Parameter</td>
<td>MainView for CICS service description</td>
<td>Applicable Rule command or IMEXEC CICS command</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ALTFILE=NO</td>
<td>YES</td>
<td>specifies whether to secure the status change capability in the FILE, DDIR, DS NAMES, and DATATABL services</td>
</tr>
<tr>
<td>ALTICES=NO</td>
<td>YES</td>
<td>specifies whether to secure the change capability in the ICE display</td>
</tr>
<tr>
<td>ALTPROG=NO</td>
<td>YES</td>
<td>specifies whether to secure the status change capability in the PROGRAM and REMOTES services</td>
</tr>
<tr>
<td>ALTTASK=NO</td>
<td>YES</td>
<td>specifies whether to secure the task kill capability in the TASK, ENQUEUE, TERMINAL, CONSOLES, and CONNXPND services</td>
</tr>
<tr>
<td>ALTTERM=NO</td>
<td>YES</td>
<td>specifies whether to secure the status change capability in the TERMINAL and CONSOLES services</td>
</tr>
<tr>
<td>ALTTRAN=NO</td>
<td>YES</td>
<td>specifies whether to secure the status change capability in the TRAN display</td>
</tr>
<tr>
<td>ALTTSUT=NO</td>
<td>YES</td>
<td>specifies whether to secure the status change capability in the TSUT display</td>
</tr>
<tr>
<td>SETTCLA=NO</td>
<td>YES</td>
<td>specifies whether to secure the alteration of the attributes of a transaction class</td>
</tr>
<tr>
<td>SETDMPD=NO</td>
<td>YES</td>
<td>specifies whether to secure the alteration of the attributes of a dump data set</td>
</tr>
</tbody>
</table>
### Table 15: Display parameters for CMRSECU TYPE=INITIAL and TYPE=ENTRY statements

<table>
<thead>
<tr>
<th>Parameter</th>
<th>MainView for CICS Service Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABEND=NO</td>
<td>YES</td>
</tr>
<tr>
<td>AID=NO</td>
<td>YES</td>
</tr>
<tr>
<td>CLASS=NO</td>
<td>YES</td>
</tr>
<tr>
<td>CONNECT=NO</td>
<td>YES</td>
</tr>
<tr>
<td>DB2SYSP=NO</td>
<td>YES</td>
</tr>
<tr>
<td>DB2TASK=NO</td>
<td>YES</td>
</tr>
<tr>
<td>DEST=NO</td>
<td>YES</td>
</tr>
<tr>
<td>DLI=NO</td>
<td>YES</td>
</tr>
<tr>
<td>DUMP=NO</td>
<td>YES</td>
</tr>
<tr>
<td>ENQUEUE=NO</td>
<td>YES</td>
</tr>
<tr>
<td>EXITS=NO</td>
<td>YES</td>
</tr>
<tr>
<td>FILE=NO</td>
<td>YES</td>
</tr>
<tr>
<td>HISTORY=NO</td>
<td>YES</td>
</tr>
<tr>
<td>Parameter</td>
<td>MainView for CICS Service Description</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>ICES=NO</td>
<td>YES</td>
</tr>
<tr>
<td>JOURNAL=NO</td>
<td>YES</td>
</tr>
<tr>
<td>MONITOR=NO</td>
<td>YES</td>
</tr>
<tr>
<td>NUCLEUS=NO</td>
<td>YES</td>
</tr>
<tr>
<td>PLAN=NO</td>
<td>YES</td>
</tr>
<tr>
<td>PPST=NO</td>
<td>YES</td>
</tr>
</tbody>
</table>
| PROBLEM=NO | YES                           | specifies whether to secure the real-time problem analysis display  
If PROBLEM=YES, you must specify OPID=xxxxxxxx in the TYPE=ENTRY statement.  
The OPID= specification must equal the AUTOID or USRID specified in BBPARM member BBIISP00. |
| PROGRAM=NO | YES                           | specifies whether to secure the program status display and the remote program status display |
| PSB=NO | YES                              | specifies whether to secure the DL/1 PSB statistics display |
| REVIEW=NO | YES                           | specifies whether to secure the startup and current values display |
| SHARE=NO | YES                              | specifies whether to secure the VSAM shared resource statistics display |
| SUBPOOL=NO | YES                           | specifies whether to secure the SUBPOOL and REGION displays |
| SUFFIX=NO | YES                              | specifies whether to secure the CICS module and table suffixes display |
| SUMMARY=NO | YES                          | specifies whether to secure the CICS performance display |
| TASK=NO | YES                              | specifies whether to secure the task statistics and MEMORY displays  
The expanded displays are included. |
| TEMP=NO | YES                              | specifies whether to secure the temporary storage usage statistics display  
The expanded displays are included. |
| TERM=NO | YES                              | specifies whether to secure the terminal statistics display in the TERMINAL and CONSOLES services  
The expanded displays are included. |
| TIOT=NO | YES                              | specifies whether to secure the region allocation display |
| TRAN=NO | YES                              | specifies whether to secure the transaction statistics display  
The expanded displays are included. |
| TSUT=NO | YES                              | specifies whether to secure the TSUT statistics display |
| VTAM=NO | YES                              | specifies whether to secure the VTAM buffer statistics display |
CMRSECU TYPE=ENTRY statement by operator ID

The CMRSECU TYPE=ENTRY statement and its parameters are used to qualify access to each

- MainView for CICS function or action service
- MainView AutoOPERATOR for CICS action service

that has been secured through the CMRSECU TYPE=INITIAL statement.

One CMRSECU TYPE=ENTRY statement is required for each operator ID that requires access to the previously secured function or service.

If the function is protected and an unauthorized operator tries to access it, an error message is issued to the operator and logged.

Specify YES or NO for each of the functions or action services that has been secured by CMRSECU TYPE=INITIAL where

- YES indicates that the operator can access the function or service that has been secured
- NO indicates that the operator cannot access the function or service that has been secured

Parameters for the CMRSECU TYPE=ENTRY statement are

**OPID= xxxxxxxx | ***

<table>
<thead>
<tr>
<th>OPID</th>
<th>specifies an operator ID, where xxxxxxxx is</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>■ TSO user ID for access through a terminal session</td>
</tr>
<tr>
<td></td>
<td>■ EXCP user ID for access through an EXCP session</td>
</tr>
<tr>
<td></td>
<td>■ VTAM user ID for access through a VTAM session</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>*</th>
<th>is a generic qualifier for an operator ID; for example</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AB*</td>
</tr>
</tbody>
</table>

The display and action parameters are the same as the CMRSECU TYPE=INITIAL parameters.
For the CMRSECU TYPE=INITIAL statement, action parameters are listed in Table 17 and the display parameters are listed in Table 18 in "CMRSECU TYPE=INITIAL statement (required)" on page 258.

**CMRSECU TYPE=FINAL statement (required)**

The CMRSECU TYPE=FINAL statement has one parameter and ends this option generation; for example:

CMRSECU TYPE=FINAL
Authorizing EXECs scheduled by an operator or the Rule Processor

EXECs can be scheduled from one BBI-SS PAS to another BBI-SS PAS in the following ways:

- using the MVS MODIFY command, a terminal operator on one BBI-SS PAS can schedule an EXEC on another BBI-SS PAS
- a Rule responding to an event can schedule an EXEC that schedules to run an EXEC on a different BBI-SS PAS

In both cases, the BBI-SS PAS where the EXEC is scheduled to run must be authorized by a BBPARM member. The BBPARM member name is the subsystem ID (SSID) of the BBI-SS PAS that sent the EXEC.

For example, if a terminal operator on BBI-SS PAS SS01 issues this command:

F SS01,%execname

and EXEC named %execname starts an EXEC on a BBI-SS PAS named SS02, the SS02 BBI-SS PAS must have a BBPARM member SS01.

Also, if the EXEC %execname issues an EXEC from SS02 to be scheduled on SS01, SS01 must have a BBPARM member SS02. This is because MainView AutoOPERATOR sends the ORIGIN of each step in an EXEC to the receiving BBI-SS PAS to verify authorization.

If the required BBPARM authorization member does not exist, MainView AutoOPERATOR issues error message AM3119W and does not schedule the EXEC.
Issuing commands from terminal sessions

MainView AutoOPERATOR allows IBM z/OS, IMS, and DBCTL commands to be issued from a terminal session (TS) command line.

Issuing z/OS commands

z/OS commands may be issued from the TS either with or without response. When issued with response, the TS enters a wait and can perform no other functions while waiting for the response. If the response is received within three seconds, it is recorded in the TS user’s local log and an automatic switch to this display occurs.

If the response is not received within three seconds, the short message

Response not available

is displayed on the current screen.

When commands are issued without response, the short message

Command issued

is displayed on the current screen.

If the MainView AutoOPERATOR for z/OS (MAO) key is active in the subsystem, MainView AutoOPERATOR attempts to issue the command with response, which requires the availability of a console. If a console is not immediately available, the command is issued without response.

Using consoles to issue z/OS commands
MainView AutoOPERATOR consoles are either subsystem consoles or Extended MCS (X-MCS) consoles. Refer to “MVS console considerations” on page 121 to determine which type of consoles are used in your environment and how they are allocated by the subsystem.

Commands issued without response are issued from a special no-response X-MCS console named SSID0000. Commands issued with response are issued using one of the Extended MCS consoles allocated to the subsystem. If a console is not immediately available, the command is issued without response from the special no-response X-MCS console named SSID0000.

The z/OS Modify command and commands that begin with a subsystem command character are issued with response from consoles that have migration IDs (MIGIDs) associated with them. All other commands can be issued from any console. If a console is not immediately available, the command is issued without response and cannot be issued from a console that has a migration ID.

Issuing BBI commands

A terminal session user may record BBI commands and responses in the local log by specifying BBIRESP=LOCAL in the BBITSP00 member of the BBPROF data set.

When this option is used, the TS enters a wait and can perform no other functions while waiting for the response. If the response is received within three seconds, an automatic switch to the local log occurs.

If the response is not received within three seconds, the short message

Response not available

is displayed on the current screen.

BBI command responses are recorded in the subsystem log regardless of the user’s BBIRESP= specification.

Issuing IMS commands

A terminal session user may record IMS commands and responses in the local log by specifying IMSRESP=LOCAL in the BBITSP00 member of the BBPROF data set.

When this option is used, the TS enters a wait and can perform no other functions while waiting for the response. If the response is received within three seconds, an automatic switch to the local log occurs.
If the response is not received within three seconds, the short message

Response not available

is displayed on the current screen.

IMS command responses are not recorded in the subsystem log unless a Rule records it, regardless of the user’s IMSRESP= specification.

**Issuing DBCTL commands**

DBCTL commands are issued from a terminal session the same way as IMS commands. For example to stop database ABC, issue the following command:

/STO DATABASE ABC
Changing the use of CICS recoverable temporary storage

BMC Software occasionally starts a CICS transaction by using interval control services. This type of transaction must complete processing because of existing recoverable temporary storage definitions in the CICS Temporary Storage Table (TST).

If a conflict exists between the BMC Software temporary storage prefix and an existing temporary storage definition, the prefix that is used by BMC Software (CMRI) must be changed by applying the following ZAP:

<table>
<thead>
<tr>
<th>NAME</th>
<th>CMRINT2</th>
<th>CMRINT2</th>
</tr>
</thead>
<tbody>
<tr>
<td>VER</td>
<td>00BC</td>
<td>C3D4D9C9</td>
</tr>
<tr>
<td>REP</td>
<td>00BC</td>
<td>xxxxxxxx (any nonrecoverable prefix)</td>
</tr>
</tbody>
</table>

For information about how to determine whether a conflict exists between the BMC Software temporary storage prefix and an existing temporary storage definition, refer to

- “Implementation considerations” on page 160 when implementing MainView AutoOPERATOR for CICS only

- "Implementation Considerations" in the chapter "Standard Implementation Procedures" in the MainView for CICS Customization Guide.
CICS transaction and program names

This appendix describes the transaction and program names generated by MainView for CICS and MainView AutoOPERATOR for CICS.

Transaction entries for MainView for CICS and MainView AutoOPERATOR for CICS

The following table describes the transaction IDs used by MainView for CICS and MainView AutoOPERATOR for CICS.

Table 17 on page 275 lists ZAPs that may be applied to change these IDs.

Table 17: Transaction IDs

<table>
<thead>
<tr>
<th>Tran ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCE</td>
<td>is started by the CICS External Interface on behalf of a request from the BBI-SS. This transaction is not created dynamically.</td>
</tr>
<tr>
<td>BCRT</td>
<td>processes GET and SET requests that require CICS SPI commands to collect data and process modifications (for CICS 4.1 and later) For CICS Transaction Server 2.x and later, it also handles, on an interval basis, the queues that control the allocation and reuse of the MainView for CICSTIE (Task Interface Element) space used by the data extractor. For user security purposes and other monitoring exits, BCRT should be handled like transaction JNL2.</td>
</tr>
<tr>
<td>FCD2</td>
<td>performs an action request that uses CICS services for completion It is scheduled once for each request to ensure proper serialization.</td>
</tr>
<tr>
<td>FCM1</td>
<td>routes messages to terminals through BMS for MainView AutoOPERATOR for CICS message broadcast</td>
</tr>
<tr>
<td>Tran ID</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>FIC2</td>
<td>captures and records MainView for CICS statistical data. It runs once each minute.</td>
</tr>
<tr>
<td>FST2</td>
<td>controls the communications, the data collection and other agent functions within the CICS. The following functions are available if the appropriate product is installed:</td>
</tr>
<tr>
<td></td>
<td>- QON: starts BBI-SS communications; MainView AutoOPERATOR; MainView for CICS data collection extractor</td>
</tr>
<tr>
<td></td>
<td>- QOFF: stops the MainView for CICS data collection extractor</td>
</tr>
<tr>
<td></td>
<td>- SYSM ON: starts MainView AutoOPERATOR for CICS</td>
</tr>
<tr>
<td></td>
<td>- SYSM OFF: stops MainView AutoOPERATOR for CICS</td>
</tr>
<tr>
<td></td>
<td>- LON: starts the MainView for CICS BCRT control task (should be used only under the direction of MainView for CICS support)</td>
</tr>
<tr>
<td></td>
<td>- LOFF: stops the MainView for CICS BCRT control task (should be used only under the direction of MainView for CICS support)</td>
</tr>
<tr>
<td></td>
<td>The following functions are available when automatic maintenance is active:</td>
</tr>
<tr>
<td></td>
<td>- INIT: build and initialize the MainView for CICS agents</td>
</tr>
<tr>
<td></td>
<td>- TERM: terminate and remove MainView for CICS from the CICS</td>
</tr>
<tr>
<td></td>
<td>- KON: activate the MainView for CICS task kill function</td>
</tr>
<tr>
<td></td>
<td>- KOFF: deactivate the MainView for CICS task kill function</td>
</tr>
<tr>
<td></td>
<td>- KRLD: reload the MainView for CICS task kill function</td>
</tr>
<tr>
<td></td>
<td>The FST2 transaction is not created dynamically. <strong>Note:</strong> On commands start BBI-SS communications if not already started.</td>
</tr>
<tr>
<td>JNL2</td>
<td>monitors the status of the BBI-SS and starts transaction FCD2 when CICS action requests are processed. JNL2 is always shown as active on the TASK display, but most of that time is spent waiting (user ECB). When JNL2 starts, it remains active because it must be present in the system at all times once communication with the BBI-SS is established. JNL2 is assigned a very high internal dispatching priority.</td>
</tr>
</tbody>
</table>
Tran ID | Description
--- | ---
SMN2 | allows manual starting of communications and data collection components. It also allows manual stopping of data collection (communications, however, remain active once started). It schedules FST2 (FST2 QON, FST2 QOFF) to perform a service and reports on the status of the components. Its use is required if the PLT is not used to start data collection and communications automatically.

If your site has existing transaction IDs that are the same as those used by MainView for CICS or MainView AutoOPERATOR for CICS, or your site has standards that preclude the use of these IDs, you need to change them. To change these IDs, apply the following ZAPs:

**Table 18: ZAPs for transaction ID changes**

<table>
<thead>
<tr>
<th>Tran ID</th>
<th>ZAP Required if Transaction ID Is Changed</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCRT</td>
<td>NAME CMRINT2 CMRINT2 VER 0080 C2C3D9E3 REP 0080 xxxxxxx</td>
</tr>
<tr>
<td>BMCE</td>
<td>NAME CMRINT2 CMRINT2 VER 00AC C2D4C3C5 REP 00AC xxxxxxx</td>
</tr>
<tr>
<td>FCD2</td>
<td>NAME CMRINT2 CMRINT2 VER 00C0 C6C3C4F2 REP 00C0 xxxxxxx New Transaction ID</td>
</tr>
<tr>
<td>FCM1</td>
<td>NAME CMRINT2 CMRINT2 VER 00A8 C6C3D4F1 REP 00A8 xxxxxxx New Transaction ID</td>
</tr>
<tr>
<td>FIC2</td>
<td>NAME CMRINT2 CMRINT2 VER 00B8 C6C9C3F2 REP 00B8 xxxxxxx New Transaction ID</td>
</tr>
<tr>
<td>FST2</td>
<td>NAME CMRINT2 CMRINT2 VER 00B0 C6E2E3F2 REP 00B0 xxxxxxx New Transaction ID</td>
</tr>
<tr>
<td>JNL2</td>
<td>NAME CMRINT2 CMRINT2 VER 00B4 D1D5D3F2 REP 00B4 xxxxxxx New Transaction ID</td>
</tr>
<tr>
<td>SMN2</td>
<td>(No ZAP required)</td>
</tr>
</tbody>
</table>
Note
While the transaction IDs may change in CICS, the original transaction IDs still appear in online Help.

Program entries for MainView for CICS and MainView AutoOPERATOR for CICS

This section provides a descriptive list of program entries generated for MainView for CICS and MainView AutoOPERATOR for CICS. The lists are grouped by release-independent and release-dependent programs.

Release-independent programs

The programs listed in the following table can operate under any CICS release.

Table 19: Release-independent programs

<table>
<thead>
<tr>
<th>Program</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMRINT2</td>
<td>CICS products internal table</td>
</tr>
<tr>
<td>CMRLOGM</td>
<td>CICS products message logging program</td>
</tr>
<tr>
<td>CSLOAD</td>
<td>CICS products common services program</td>
</tr>
<tr>
<td>CMRTMON</td>
<td>CICS products interactive startup program</td>
</tr>
<tr>
<td>CMRTMAP</td>
<td>CICS products interactive startup BMS mapset</td>
</tr>
<tr>
<td>CBQCMD</td>
<td>MainView AutoOPERATOR for CICS broadcast program</td>
</tr>
<tr>
<td>CMRXTFG</td>
<td>MainView for CICS verification program</td>
</tr>
<tr>
<td>OLTSCAO</td>
<td>start and stop MainView AutoOPERATOR for CICS</td>
</tr>
<tr>
<td>OLTVCAO</td>
<td>CSD program that initially called for CAO startup</td>
</tr>
<tr>
<td>OLTCAORL</td>
<td>checks for valid CICS release</td>
</tr>
<tr>
<td>OLTFSSET</td>
<td>CICS products PLTPI and initialization program</td>
</tr>
<tr>
<td></td>
<td>This program is not dynamically created.</td>
</tr>
<tr>
<td>OLTCNTL</td>
<td>CICS products release and product verification program</td>
</tr>
<tr>
<td>OLTVER</td>
<td>determines the OLTP release level</td>
</tr>
<tr>
<td>OLTCCRE</td>
<td>common resource definition create program</td>
</tr>
<tr>
<td>BALPUTLV</td>
<td>PUT maintenance level verification</td>
</tr>
</tbody>
</table>
Program entries for MainView for CICS and MainView AutoOPERATOR for CICS

<table>
<thead>
<tr>
<th>Program</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMRVCMR</td>
<td>verify presence of MainView for CICS</td>
</tr>
<tr>
<td></td>
<td>This program is not created dynamically.</td>
</tr>
<tr>
<td>CMR9RLSE</td>
<td>release identification table</td>
</tr>
<tr>
<td>BBK1CCRE</td>
<td>MainView for CICS resource definition create program</td>
</tr>
<tr>
<td>BBK1CRTR</td>
<td>EXEC CICS and TIE space control program</td>
</tr>
<tr>
<td>BBK1TE3</td>
<td>CTCPXRV function processor</td>
</tr>
</tbody>
</table>

**Release-dependent programs**

These programs operate under a specific CICS release. The release dependency is indicated by the following suffixes that are used with the program name.

Table 20 on page 277 is a descriptive list of the programs.

<table>
<thead>
<tr>
<th>Suffix</th>
<th>CICS Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>CICS Transaction Server 3.1</td>
</tr>
<tr>
<td>8</td>
<td>CICS Transaction Server 3.2</td>
</tr>
<tr>
<td>4</td>
<td>CICS Transaction Server 4.1</td>
</tr>
<tr>
<td>7</td>
<td>CICS Transaction Server 4.2</td>
</tr>
<tr>
<td>6</td>
<td>CICS Transaction Server 5.1</td>
</tr>
<tr>
<td>3</td>
<td>CICS Transaction Server 5.2</td>
</tr>
<tr>
<td>9</td>
<td>CICS Transaction Server 5.3</td>
</tr>
</tbody>
</table>

Table 20 on page 277 is a descriptive list of the CICS release-dependent programs.

**Table 20: Release-dependent programs**

<table>
<thead>
<tr>
<th>Program</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMRFSETx</td>
<td>CICS Products initialization programs</td>
</tr>
<tr>
<td>OLTFTSETx</td>
<td>CICS products general agent command processor and initialization program</td>
</tr>
<tr>
<td>CMRFCMDx</td>
<td>CICS products action service program</td>
</tr>
<tr>
<td>CMRTDPPx</td>
<td>MainView AutoOPERATOR for CICS transient data exit program</td>
</tr>
<tr>
<td>Program</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CMRCMPXx</td>
<td>MainView for CICS monitoring task-related user exit program</td>
</tr>
<tr>
<td>CMRKCPXx</td>
<td>data collection global exit program</td>
</tr>
<tr>
<td>CMRTRUEx</td>
<td>task related user exit program stub</td>
</tr>
<tr>
<td>CMRROLLx</td>
<td>statistics output program</td>
</tr>
<tr>
<td>CMRSLOGx</td>
<td>statistics gathering program</td>
</tr>
<tr>
<td>CMRXEIOx</td>
<td>task kill function processor</td>
</tr>
<tr>
<td>BBK1TE9x</td>
<td>TSQUECF function processor</td>
</tr>
<tr>
<td>BBK1RE5x</td>
<td>CJVMPL function processor</td>
</tr>
</tbody>
</table>
Manual activation of BBI-SS to CICS communication

If BBI-SS to CICS communication has already been implemented, but is not started automatically through the CICS PLTPI table, it can be started manually either from an z/OS console that has been defined to CICS, or from a CICS terminal.

For instructions on implementing BBI-SS to CICS communication, refer to

- “Implementing BBI-SS PAS to CICS Communication” in the MainView Customization Reference when implementing MainView AutoOPERATOR for CICS only
- “Standard Implementation Procedures” in the MainView for CICS Customization Guide (when implementing MainView for CICS or MainView AutoOPERATOR for CICS or both)

Manual activation from a CICS terminal

To start BBI-SS to CICS communication from a CICS terminal:

1. Invoke the SMN2 transaction.
2. In the ESTABLISH COMMUNICATIONS field, type YES.
3. Press Enter.
Manual activation from a z/OS console that is defined to CICS

To start BBI-SS to CICS communication from an z/OS console which has been defined to CICS:

Use the MODIFY command to invoke the FST2 transaction in the CICS region. For example:

```
F CICSPROD, FST2 INIT
```

invokes the FST2 transaction in the CICS region with the z/OS job name of CICSPROD.
This section describes BBPARM data set members that you can use as samples to set up and customize MainView AutoOPERATOR.

Table 21 on page 281 describes all of the BBPARM members that are specific to MainView AutoOPERATOR. Documentation for other BBPARM members (such as BBISSP00) or that are shared among several products can be found in the MainView Customization Reference.

### Table 21: BBPARM data set members: MainView AutoOPERATOR only

<table>
<thead>
<tr>
<th>BBPARM member name</th>
<th>Description</th>
</tr>
</thead>
</table>
| AAOALS00           | defines limits for extended storage for MainView AutoOPERATOR ALERTs and ALERT parameters on your system  
This member supports system variables substitution and the following control card:  
```
PROCESS [SYM=YES|NO,] [LIST=YES|NO]
```
Keywords can be specified in any order but when both keywords are used, they must be separated by a comma. Refer to “System variables substitution in parameter library members” on page 284 for more information. |
| AAOARP00           | used by the MainView AutoOPERATOR Automation Reporter application to activate and control collection intervals for offline data collection  
This member supports system variables substitution and the following control card:  
```
PROCESS [SYM=YES|NO,] [LIST=YES|NO]
```
Keywords can be specified in any order but when both keywords are used, they must be separated by a comma. Refer to “System variables substitution in parameter library members” on page 284 for more information. |
<table>
<thead>
<tr>
<th>BBPARM member name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **AAOEXP00**       | defines thread values and high-priority EXEC names and limits the CPU time that EXECs can use. This member supports system variables substitution and the following control card: \[
\text{PROCESS [SYM=YES|NO,] [LIST=YES|NO]}
\] Keywords can be specified in any order but when both keywords are used, they must be separated by a comma. Refer to “System variables substitution in parameter library members” on page 284 for more information. |
| **AAOGME00**       | used by MainView AutoOPERATOR to create General Messages Exchange (GME) connections to allow applications to communicate. This member supports system variables substitution and the following control card: \[
\text{PROCESS [SYM=YES|NO,] [LIST=YES|NO]}
\] Keywords can be specified in any order but when both keywords are used, they must be separated by a comma. Refer to “System variables substitution in parameter library members” on page 284 for more information. |
| **AAOMQL00**       | used to specify which MQSeries queues will be made eligible for automation. This member supports system variables substitution and the following control card: \[
\text{PROCESS [SYM=YES|NO,] [LIST=YES|NO]}
\] Keywords can be specified in any order but when both keywords are used, they must be separated by a comma. Refer to “System variables substitution in parameter library members” on page 284 for more information. |
| **AAONCD00**       | used for the MainView AutoOPERATOR Access NV option. This member supports system variables substitution and the following control card: \[
\text{PROCESS [SYM=YES|NO,] [LIST=YES|NO]}
\] Keywords can be specified in any order but when both keywords are used, they must be separated by a comma. Refer to “System variables substitution in parameter library members” on page 284 for more information. |
| **AAOPLX00**       | defines MainView AutoOPERATOR General RTCS Server AOPlex specifications. This member supports system variables substitution and the following control card: \[
\text{PROCESS [SYM=YES|NO,] [LIST=YES|NO]}
\] Keywords can be specified in any order but when both keywords are used, they must be separated by a comma. Refer to “System variables substitution in parameter library members” on page 284 for more information. |
<table>
<thead>
<tr>
<th>BBPARM member name</th>
<th>Description</th>
</tr>
</thead>
</table>
| AAOPRM00           | defines Operator Request parameters for the Operator Request application. This member supports system variables substitution and the following control card:  
PROCESS [SYM=YES|NO,] [LIST=YES|NO]  
Keywords can be specified in any order but when both keywords are used, they must be separated by a comma. Refer to “System variables substitution in parameter library members” on page 284 for more information. |
| AAORUL00           | contains Rule Set definitions. This member does not support system variables substitution, but does support the LIST option on the PROCESS control card. Refer to “System variables substitution in parameter library members” on page 284 for more information. |
| AAOTRN00           | contains the identification of IMS transactions that can be submitted from a MainView AutoOPERATOR EXEC or from a terminal session. This member does not support system variables substitution, but does support the LIST option on the PROCESS control card. Refer to “System variables substitution in parameter library members” on page 284 for more information. |
| AAOTSP00           | used for the TapeSHARE for MainView AutoOPERATOR option. This member supports system variables substitution and the following control card:  
PROCESS [SYM=YES|NO,] [LIST=YES|NO]  
Keywords can be specified in any order but when both keywords are used, they must be separated by a comma. Refer to “System variables substitution in parameter library members” on page 284 for more information. |
| BBIVAR00           | defines any number of preset SHARED variables and their values to be set in the SHARED pool at the COLD start of the BBI-SS PAS COLD when VPOOL=RESET is specified. This member supports system variables substitution and the following control card:  
PROCESS [SYM=YES|NO,] [LIST=YES|NO]  
Keywords can be specified in any order but when both keywords are used, they must be separated by a comma. Refer to “System variables substitution in parameter library members” on page 284 for more information. |
Some MainView AutoOPERATOR BBPARM members support system variables substitution. (Refer to Table 21 on page 281 for a brief description of each BBPARM member.) With symbolic substitution, you can reduce the number of parameter libraries that you maintain. When the member is read, the value of the symbol is substituted for the symbol. The symbol must begin with an ampersand, followed by one to eight characters, and can optionally end with a period. You can define your own symbols by placing them in the IBM PARMLIB member IEASYMxx.

Symbolic names can be concatenated as a string, and names can be added at the beginning, middle or end of the string. See the following example:

```
DSN=VAM3.&SYSPLEX..&SYSNAME..OFFLOAD
```

If &SYSPLEX=BMC1 and &SYSNAME=SJSD, the preceding string converts to

```
DSN=VAM3.BMC1.SJSD.OFFLOAD
```

**Note**

A period at the end of a symbolic name is assumed to be a delimiter and is omitted in the conversion. If you want to generate the string BMC1SJSD, you would code &SYSPLEX&SYSNAME.

For more information about symbolic substitution, refer to "System Symbols and Symbolic Substitution" in the IBM Initialization and Tuning Reference.

You can control substitution on a member-by-member basis by adding a control card (a PROCESS statement) at the beginning of each member. The PROCESS statement is described in Table 22 on page 285.
### Table 22: PROCESS statement

<table>
<thead>
<tr>
<th>Control Card</th>
<th>Description</th>
</tr>
</thead>
</table>
| PROCESS [SYM=YES| NO,][LIST=YES| NO]                                                 | indicates whether system variables substitution is allowed When present, the PROCESS statement must be the first statement in the member and must begin in column 1. Possible keywords are as follows  
- SYM=YES| NO  
  - YES is the default and indicates that symbols are replaced by their value if the calling program has enabled symbol substitution.  
  - NO indicates that symbolic substitution is suppressed for this member even if the calling program has enabled substitution.  
  Only the first character (Y or N) needs be specified. Substitution is not performed on comments or any data beyond column 71.  
- LIST=YES| NO  
  This keyword indicates whether the contents of the member being processed (all statements following the PROCESS statement) are written to the job log, one statement at a time as the member is read.  
  - YES causes the entire contents of the member (except the PROCESS statement) to be written to the job log. If substitution is performed, two WTO messages are displayed for each statement where substitution is performed. The first message contains the statement with the symbol, and the second message contains the same statement with the substituted value of the symbol. The second statement is also prefixed with < > to indicate that the statement is modified.  
  - NO is the default and indicates that the contents of this member are not written to the job log or displayed as WTO messages.  |

You can use the Dynamic Parameter Manager (DPM) to display and alter specific parameters in members AAOEXPx, AAOALSxx, AAOARPx, and AAOATSxx. For more information about DPM, see “Continuous operation with the Dynamic Parameter Manager” on page 69.

---

**BBPARM member AAOALS00 parameters**

BBPARM member AAOALS00 is used by all MainView AutoOPERATOR products to define limits for extended storage for ALERTs on your system.
Note
This member supports system variables substitution. For more information, see “System variables substitution in parameter library members” on page 284.

For information about using the ALERTs application, refer to the *MainView AutoOPERATOR Basic Automation Guide, Volume 2*, for information about using the MainView AutoOPERATOR Dynamic Parameter Manager application to update the parameters in AAOALS00, refer to “Modifying ALERT thresholds in AAOALSxx” on page 74.

Table 23 on page 286 lists the parameters for AAOALS00.

### Table 23: BBPARM member AAOALS00

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALERTLOG= [YES</td>
<td>NO]</td>
</tr>
<tr>
<td>ALERTXCF= [YES</td>
<td>NO]</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **AUDIT**=[NONE | ALL | ADD | ESCAL | DELETE] | specifies options for auditing the life cycle of an ALERT  
The default is NONE. Valid values are as follows  
  - **ALL**: All stages of an ALERT’s life cycle will be audited.  
  - **NONE**: No auditing will be performed.  
  - **ADD**: Generate an audit message when an ALERT is created with the ADD keyword.  
  - **ESCAL**: Generate an audit message when an ALERT’s priority escalates.  
  - **DELETE**: Generate an audit message when an ALERT is deleted with the DELETE keyword.  
  Audit messages are generated into the BBI Journal. |
| **MAXSTOR**=[0 | nnnnn] | maximum amount of BBI-SS PAS private virtual storage to be allocated to ALERTs  
Storage is allocated in kilobytes. The default is 0, which means ALERTs are allocated without regard to the amount of virtual storage they occupy. BMC Software strongly recommends that you set this number to a value greater than 0.  
For example, a value of 2000 K means that you can store about 7800 ALERTs. Use the MainView AutoOPERATOR ALERT Statistics panel to estimate the average size of an ALERT on your system.  
Possible values range from 0 to 99999. A value of 0 means that no checking is done. |
| **WARNLVL1**=[60 | nn] | first warning threshold level for queued ALERTs when the MAXSTOR parameter is set to a value greater than 0 K  
The default is 60%, which means that if MAXSTOR=1000, the warning message (AU6100W) is issued when the amount of virtual storage that is allocated to queued ALERTs reaches 600 K.  
Possible values range from 0 to 99. |
| **WARNLVL2**=75 | nn | second warning threshold level for queued ALERTs when the MAXSTOR parameter is set to a value greater than 0 K  
The default is 75%, which means that if MAXSTOR=1000, the warning message (AU6100W) is issued when the amount of virtual storage allocated to queued ALERTs reaches 750 K.  
Possible values range from 0 to 99.  
Important ALERTs are not generated when ALERT storage usage reaches 100%. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| ALERTNV=[YES | NO] | default setting for the RETAIN keyword for MainView AutoOPERATOR ALERTs  
ALERTs that are retained are written and saved to disk and therefore, exist across BBI-SS PAS restarts and system z/OS IPLs.  
When you are creating ALERTs (using the IMFEXEC ALERT statement, the Rules Processor ALERT panels, the AOEXEC ALERT statement, and so on) and you do not explicitly specify a setting for the RETAIN keyword, the ALERTNV parameter allows you to specify whether the default setting should be RETAIN(NO) or RETAIN(YES).  
For example, for ALERTs that do not specify the RETAIN keyword:  
- when the ALERTNV parameter is not specified or is specified as NO, RETAIN(NO) is used as the default.  
- when ALERTNV=YES is specified, RETAIN(YES) is used as the default.  
Regardless of which setting you specify for the ALERTNV parameter, you can override the setting when you create a specific ALERT, where the RETAIN=YES or RETAIN=NO keyword is explicitly set. When you override this setting, only that ALERT is retained across BBI-SS PAS restarts and z/OS IPLs.  
**Note:** When an ALERT is targeted to a different BBI-SS PAS, by default, the ALERT retains the ALERTNV setting of the target BBI-SS PAS. Changing the ALERTNV setting takes effect only after a BBI-SS PAS warm or cold start.  
The default is NO. |
| PUBLISH=[ADD | REPLACE | NO] | default for the PUBLISH parameter when creating/deleting MainView AutoOPERATOR ALERTs  
Specifies whether an ALERT is published and how it is published to connected PATROL Enterprise Manager 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 as follows:  
- ADD: Send ALERTs but do not override previously published ALERTs; does not apply to ADAPTOR connections  
- REPLACE: Send ALERTs but delete previously published ALERTs that had the same Key and Queue before new ALERT is published; does not apply to ADAPTOR connections  
- NO: Do not send ALERTs  
The default is ADD. |
BBPARM member AAOARP00 parameters

BBPARM member AAOARP00 is used by the MainView AutoOPERATOR Automation Reporter application to activate and control the collection intervals for the offline data collection portion of the Automation Reporter.

**Note**
This member supports system variables substitution. For more information, see “System variables substitution in parameter library members” on page 284.

For documentation about how to use the Automation Reporter, refer to the *MainView AutoOPERATOR Basic Automation Guide, Volume 2*; for documentation about using the MainView AutoOPERATOR application Dynamic Parameter Manager to update the AAOARP00 member, refer to “Accessing the Dynamic Parameter Manager” on page 70.

Table 24 on page 290 lists the parameters for AAOARP00.
Table 24: BBPARM member AAOARP00

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| AROPTS=ALL | activates all, one, or more Automation Reporter data collectors. The default if AROPTS is not specified means the following items are collected:  
- ACT: Actions  
- ALT: Alerts  
- AUS: Automation Stats  
- CPU: CPU usage by Program  
- EVT: Events  
- EXE: EXECs  
- EXC: EXEC Commands  
- HTH: Health Metrics  
- RES: Resource Usage  
- RST: Rule Sets  
- RUL: Rules  
- TOM: TOMEXEC functions  
- TSL: TapeShare - Local  
- TSP: TapeShare - Plex  
- ELU: Include EXE data collection  
- RLU: Include RUL data collection  
ALL is mutually exclusive with all of the other options. If you do not specify ELU, no EXE data collection is performed unless you specify the EXE value. If you do not specify ELU, no RUL data collection is performed, unless you specify the RUL value. |
| DSN=[data set name] | name of a user-allocated data set where collected data is offloaded. This data set must be allocated or no data collection occurs. Data Set Requirements: This user-allocated data set should be a fixed block sequential data set and the maximum record length should be 580 bytes. |
| DISP=[SHR | MOD] | disposition (shared or modify) of the user-allocated data set. The default is SHR. Specifying SHR causes the Automation Reporter to overwrite existing data; specifying MOD causes the collected data to be appended to the end of the data set. |
| INTVL=[0 | 10 -- 1440] | time interval (in minutes) at which data is collected. The default is 10 minutes; specifying 0 means no data collection is performed. The minimum interval is 10 minutes and the maximum is 1440 (once a day). |
### IRHIST=[NO | YES]

Specifies whether the windows mode Automation Reporter will collect interval data that allows you to perform short-term and long-term analysis of automation processes and performance.

This parameter also enables the windows-mode Automation Reporter views that show real-time data collected by interval and allows for local or SSI summarization of the automation information.

For more information about the windows-mode version of the Automation Reporter, refer to the *MainView AutoOPERATOR Basic Automation Guide, Volume 2*.

**Note:** BMC recommends that you migrate your automation data collection processes to use the MainView AutoOPERATOR version 7.1 windows-mode Automation Reporter application instead of using the Automation Reporter, available with MainView AutoOPERATOR version 6.5.xx and earlier. In a future release, BMC Software plans to discontinue supporting the MainView AutoOPERATOR version 6.5.xx (and earlier) version of the Automation Reporter application.

The default is NO; valid values are YES or NO.

### OFFINTVL=[60 -- 1440]

Data offload interval in minutes.

The default is 1440 minutes (once a day).

You can specify any value from 60 minutes (once an hour) to 1440 minutes.

**Note:** The greater the frequency of offloading collected data, the faster the user-allocated data set is filled.

### QUEUE=[MainView AutoOPERATOR ALERT queue name]

One- to eight-character MainView AutoOPERATOR ALERT queue name to which any ALERTs generated by the Automation Reporter are routed.

The queue specified here will be created, if it does not already exist, when the first Automation Reporter ALERT is created.

---

## BBPARM member AAOEXP00 parameters

BBPARM member AAOEXP00 is used by all MainView AutoOPERATOR products to define thread values, priority EXEC names, EXEC loop parameters, data set allocation, and abend count parameters.

**Note**

This member supports system variables substitution. For more information, see “System variables substitution in parameter library members” on page 284.

For information about MainView AutoOPERATOR EXECs, refer to the *MainView AutoOPERATOR Advanced Automation Guide*; for documentation about using the
MainView AutoOPERATOR application Dynamic Parameter Manager to update the parameters in AAOEXP00, refer to the MainView AutoOPERATOR Customization Guide.

Table 25 on page 292 lists the parameters for AAOEXP00.

Table 25: BBPARM member AAOEXP00

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APIAUTH = [RACF</td>
<td>ORIGIN]</td>
</tr>
<tr>
<td></td>
<td>Valid values are</td>
</tr>
<tr>
<td></td>
<td>ORIGIN specifies that the EXEC origin is used as user ID for MV API calls. For ALERT-initiated EXECs, the origin is the origin of the ALERT. For Rule-initiated EXECs, the origin is the SSID of the BBI-SS PAS started task.</td>
</tr>
<tr>
<td></td>
<td>RACF specifies that the user ID associated with the BBI-SS PAS started task is used for MV API calls.</td>
</tr>
<tr>
<td></td>
<td>Note: This parameter applies only when you are using the IMFEXEC MV API (IMFEXEC MV commands) and does not apply to the MainView Extended API (IMFEXEC MVE commands).</td>
</tr>
<tr>
<td>AUDITMSG = [YES</td>
<td>NO]</td>
</tr>
<tr>
<td>AUDITRES = [YES</td>
<td>NO]</td>
</tr>
<tr>
<td>AUDITEXE = [YES</td>
<td>NO]</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CICSECID=[ORIGIN</td>
<td>specifies whether the user ID or the current origin of the EXEC is used for security checking of a IMFEXEC CICS command that is used in a remote EXEC when it is scheduled by a local Rule-initiated EXEC. Default value is ORIGIN, which is used if you do not specify a value for this keyword; valid values are ORIGIN or RACF. Default value is ORIGIN, which is used if you do not specify a value for this keyword; valid values are as follows: ORIGIN specifies that the EXEC origin is the name of the originating BBI-SS PAS where the EXEC was scheduled and the name of that BBI-SS PAS is used during security checking. RACF specifies that the security ID of the BBI-SS PAS is used during security checking. In both cases, the user ID is used for security checking of CICS commands that are used in EXECs scheduled in a remote BBI-SS PAS that are called by a local EXEC.</td>
</tr>
<tr>
<td>RACF ]</td>
<td></td>
</tr>
<tr>
<td>ELAPSLIM=[0</td>
<td>nnnn ] is the elapsed time limit for all EXECs If an EXEC is still running after the number of minutes specified, it will be canceled. Cancellation messages are sent to the BBI Journal that say an EXEC has exceeded the ELAPSLIM limit. You can set elapsed time limits for an individual EXEC with the IMFEXEC CNTL ELAPSLIM() keyword. Valid values range from 0 to 9999 and the default is 0.</td>
</tr>
<tr>
<td>EMLOG=[YES</td>
<td>NO] specifies whether messages from EXECs will be written to the EMLOG Valid values are YES or NO. The default is NO. When you specify EMLOG=YES, the windows-mode EMLOG view is available for viewing, sorting and filtering EXEC messages. In the EMLOG view, you can select to view (filter) all the messages from a specific EXEC. This filtering ability is not supported in the MVLOG view. Use the MVLOGSS parameter in BBPARM member BBISSP00 to specify the SSID of the MainView Logger where EMLOG records are sent.</td>
</tr>
<tr>
<td>EXHMC=[YES</td>
<td>NO] specifies whether the IMFEXEC HMC command can be used in EXECs Valid values are YES or NO. The default is NO. Refer to “Describing the fields” on page 92 for more information about the implication of using IMFEXEC HMC.</td>
</tr>
<tr>
<td>EXABCNT=[5</td>
<td>nn] is the maximum number of abends that a single EXEC can sustain before it is disabled Valid values range from 1 to 99. The default is 5. For example, if you specify EXABCNT=10 and an EXEC abends 10 times, the EXEC is disabled after the tenth abend.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| EXEC= [name]  | specifies the names of EXECs that are to be executed as high priority EXECs  
|               | A high priority EXEC is any EXEC that is listed with the EXEC= keyword. This parameter can be repeated.                                                                                                    |
|               |                                                                                                                                                                                                            |
| MAXTPUT= [0 | nnnnn]   | is the maximum number of TSO TPUTs allowed for each execution of an EXEC  
|               | Valid values range from 0 to 32767  
|               | The default is 0, which means *no checking is done*.  
|               | TPUTs for EXECs scheduled with the WAIT=YES parameter are counted separately from the scheduling EXEC.                                                                                                     |
|               |                                                                                                                                                                                                            |
| MAXHIGH= [5 | nn]    | is the maximum high priority EXECs to execute concurrently  
|               | A high priority EXEC is any EXEC that is listed with the EXEC= keyword.                                                                                                                                     |
|               |                                                                                                                                                                                                            |
| MAXHIGHQ= [0 | nnn]   | is the maximum number of high priority EXECs that may be queued for execution  
|               | Valid values range from 0 to 999. The default is 0, which means *no checking is done*.                                                                                                                     |
|               |                                                                                                                                                                                                            |
| MAXHOTQ= [10 | nnn]  | is the maximum number of First or Hot priority EXECs that can be queued  
|               | When the maximum is reached, no more EXECs with First or Hot priority specified are scheduled until the queue level drops below the maximum level set.  
|               | Valid values range from 0 to 999. The default is 10, which means *no checking is done* and all Hot or First priority EXECs are queued.                                                                       |
|               |                                                                                                                                                                                                            |
| MAXNORM= [1 | nn]   | is the maximum normal priority EXECs to execute concurrently  
|               | A normal priority EXEC is any EXEC not in the high priority list.                                                                                                                                          |
|               |                                                                                                                                                                                                            |
| MAXNORMQ= [0 | nnn]  | is the maximum number of normal priority EXECs that may be queued for execution  
|               | Valid values range from 0 to 999. The default is 0, which means *no checking is done*.                                                                                                                     |
|               |                                                                                                                                                                                                            |
| MSGID = [Y | N | I]  | updates the TSO user profile MainView AutoOPERATOR uses to turn on (and off) the TSO MSGID option  
|               | The following lists the valid values:  
|               | ■ Y: sets a PROFILE MSGID in the TSO user profile  
|               | ■ N: sets a PROFILE NOMSGID in the TSO user profile  
|               | ■ I: specifies that the EXEC Manager will bypass any setting of the user profile  
<p>|               | The default setting is Y.                                                                                                                          |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSPINUM=[0</td>
<td>nnn]</td>
</tr>
<tr>
<td>OSPIPRFX=[OSPI</td>
<td>name]</td>
</tr>
<tr>
<td>PEREXLIM=[0</td>
<td>nn ]</td>
</tr>
<tr>
<td>PREFIX=[SSID</td>
<td>name]</td>
</tr>
<tr>
<td>RESAUTH=[ORIGIN</td>
<td>RACF]</td>
</tr>
<tr>
<td>SELLIM=[0</td>
<td>nn]</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SUBXAUTH=[DEFER</td>
<td>(default) the decision for using JOBNAME or user ID for the security check is deferred to the batch job (IMFSUBEX) execution time.</td>
</tr>
<tr>
<td>JOBNAME</td>
<td>the JOBNAME is used for the IMFSUBEX security checks. This parameter overrides the origin specified on the batch job.</td>
</tr>
<tr>
<td>RACF</td>
<td>the user ID is used for the IMFSUBEX security checks. This parameter overrides the origin specified on the batch job.</td>
</tr>
<tr>
<td>USER</td>
<td>same as RACF</td>
</tr>
<tr>
<td>TIMEXLIM=[00</td>
<td>specifies an optional CPU time limit for an EXEC. If the time limit is exceeded, the EXEC is terminated automatically. With the default value of 0, no time limit is assigned.</td>
</tr>
<tr>
<td>nnnn]</td>
<td>specifies the time-out value (in seconds) for attached TSO commands. A value of 0 disables timing.</td>
</tr>
<tr>
<td>TSOTIME=[1200</td>
<td>specifies the time-out value (in seconds) for attached TSO commands. A value of 0 disables timing.</td>
</tr>
<tr>
<td>nnnn]</td>
<td>specifies the time-out value (in seconds) for attached TSO commands. A value of 0 disables timing.</td>
</tr>
<tr>
<td>UCPARMS=[YES</td>
<td>specifies whether EXEC parameters are translated to uppercase.</td>
</tr>
<tr>
<td>NO]</td>
<td>specifies whether EXEC parameters are translated to uppercase.</td>
</tr>
<tr>
<td>UNITNAME=[SYSALLDA</td>
<td>specifies an esoteric unit name to be used when you allocate a data set in an EXEC and the unit parameter is omitted.</td>
</tr>
<tr>
<td>name]</td>
<td>specifies an esoteric unit name to be used when you allocate a data set in an EXEC and the unit parameter is omitted.</td>
</tr>
<tr>
<td>WARNLVL1=[60</td>
<td>specifies a preliminary percentage of EXECs in a queue that, when met, causes a warning message (EM6100W) to be issued. Valid values range from 0% to 99%. The default is 60%. Use this threshold to set the first percentage that triggers sending a warning message.</td>
</tr>
<tr>
<td>nn]</td>
<td>specifies a preliminary percentage of EXECs in a queue that, when met, causes a warning message (EM6100W) to be issued. Valid values range from 0% to 99%. The default is 60%. Use this threshold to set the first percentage that triggers sending a warning message.</td>
</tr>
<tr>
<td>WARNLVL2=[75</td>
<td>specifies a secondary percentage of EXECs in a queue that, when met, causes a warning message (EM6100W) to be issued. Valid values range from 0% to 99%. The default is 75%. Use this threshold to set a second percentage that triggers sending a warning message.</td>
</tr>
<tr>
<td>nn]</td>
<td>specifies a secondary percentage of EXECs in a queue that, when met, causes a warning message (EM6100W) to be issued. Valid values range from 0% to 99%. The default is 75%. Use this threshold to set a second percentage that triggers sending a warning message.</td>
</tr>
</tbody>
</table>

**BBPARM member AAOGME00 parameters**

BBPARM member AAOGME00 is used by the MainView AutoOPERATOR General Messages Exchange (GME) to

- identify which GME nodes receive messages from and send messages to MainView AutoOPERATOR
- define the local GME node
- control which instrumentation events are traced
Note
This member supports system variables substitution. For more information, see “System variables substitution in parameter library members” on page 284.

Table 26 on page 297 lists the parameters for AAOGME00.

### Table 26: BBPARM member AAOGME00

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| CONAUDIT= [YES | NO]                                                                                                                                          | optional parameter that activates or deactivates WTO for messages that indicate cell connection status  
|                 | Specifying YES causes the following messages to be issued as a single multiline WTO in addition to Journal messages:                                                                                 |
|                 | ■ GI0712I                                                                                                                                                                                                 |
|                 | ■ GI0702I                                                                                                                                                                                                 |
|                 | ■ GI0684I                                                                                                                                                                                                 |
|                 | The default is NO.                                                                                                                                |                                                                                                                                                           |
| DEBUGMSG= [YES | NO]                                                                                                                                          | optional parameter that specifies whether to display GME debug messages in the BBI Journal and in WTO format  
|                 | Valid values are YES and NO. The default value is NO.                                                                                           |                                                                                                                                                           |
| EVENTS= [YES | NO]                                                                                                                                          | optional for the Server BIM connection that specifies whether the connection sends incoming events to the Rules Processor  
|                 | Valid values are YES and NO. The default is YES.                                                                                               |                                                                                                                                                           |
|                 | If you specify NO, there will be no BAROC events sent sent to the Rules Processor to drive BRC initiated Rules. This parameter is valid only on the server definition that is the TGTIP statement for the HOSTNAME definition. |
| EXEC= [YES | NO ]                                                                                                                                          | optional parameter that specifies whether to allow the launching of EXECs from a BMC Impact Manager cell to run in the MainView AutoOPERATOR BBI-SS PAS.  
|                 | The default is YES.                                                                                                                             |                                                                                                                                                           |
| GMEACT= [YES | NO ]                                                                                                                                          | when used in conjunction with the RESET PARM AAOGMEEnn, RESYNC | RECYCLE command, activates or deactivates the GME facility  
<p>|                 | Valid values are YES and NO. The default value is NO.                                                                                          |                                                                                                                                                           |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOSTNAME=</td>
<td>optional parameter that specifies the name of the local GME node and is used to identify this node to other nodes</td>
</tr>
<tr>
<td></td>
<td>The default value is the SSID of the BBI-SS PAS. The value can consist of numeric and alphabetic characters, underscores (_), and periods (.), but the name must begin with an alphabetic character. The value cannot contain blanks and the maximum length is 48 characters.</td>
</tr>
<tr>
<td>MVE_LINK=HOST:PORT</td>
<td>optional parameter that specifies the host name where the MainView Explorer server is running and the port it is listening on</td>
</tr>
<tr>
<td></td>
<td>The default is the IP address where MainView AutoOPERATOR is running and port 3940.</td>
</tr>
<tr>
<td></td>
<td>You can specify the HOST address as a DNS name or an IP address. If HOST is specified, you must also specify the PORT. Valid values can be 3- to 48-characters in length.</td>
</tr>
<tr>
<td></td>
<td>This information is used in the mv _ mve_host and mv_mve_port slots of the BRC events.</td>
</tr>
<tr>
<td>TGTBIMCF=</td>
<td>when defining BMC Impact Manager connections, a required parameter that specifies the data set name of the BMC Impact Manager configuration file</td>
</tr>
<tr>
<td>[data set name]</td>
<td>The maximum length is 44 characters; do not use quotation marks.</td>
</tr>
<tr>
<td></td>
<td>For more information about how to specify this parameter when configuring MainView AutoOPERATOR to BMC Impact Manager, refer to the section “Establishing a connection to a BMC Event Manager” on page 210 and see Steps 6 and 7.</td>
</tr>
<tr>
<td>TGTHB=</td>
<td>specifies the heartbeat interval of the remote GME in minutes and if this amount of time passes without any messages being received, the local node closes the connection</td>
</tr>
<tr>
<td>[0</td>
<td>nnn ]</td>
</tr>
<tr>
<td></td>
<td>This parameter is not applicable for connections to BMC Impact Manager.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TGTIP=[network address]</td>
<td>required parameter for BMC Impact Manager connections where you specify the BMC Impact Manager cell name that is also specified in the BIM server directory data set. The BIM server directory data set is specified by the ServerDirectoryName= parameter in the BMC Impact Manager configuration file. For more information refer to Chapter 4, “Configuring BMC Impact Integration for MainView” in the MainView AutoOPERATOR BMC Impact Integration for z/OS User Guide. The cell name specified both here and in the BMC Impact Manager server directory data set must be specified in uppercase even though the actual cell name may contain mixed case characters. MainView AutoOPERATOR must be defined as a target when using BMC Impact Manager connections. This means that a TGTIP=xxxx,TGTBIMCF=yyyy specification is required for BMC Impact Manager connections; xxxx must be the same value that the HOSTNAME= parameter is using. For more information about this parameter when configuring MainView AutoOPERATOR to BMC Impact Manager, refer to the section “Establishing a connection to a BMC Event Manager” on page 210 and see Steps 6 and 7.</td>
</tr>
<tr>
<td>TGTMXMSL=[32768</td>
<td>nnnnn]</td>
</tr>
<tr>
<td>TGTNAME=[name]</td>
<td>is a 1 to 16 character name assigned to a BMC Impact Manager (BMC IM) connection. The TGTNAME can be shared by multiple BMC IM connections and is used by MainView AutoOPERATOR Rules or EXECs to send events or publish ALERTs to a specific target system. In Rules and EXECs, you may use wildcards when specifying the target name for events and ALERTs. There is no default value.</td>
</tr>
<tr>
<td>TGTRTC=[9999</td>
<td>nnnn]</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| TGTRTI=[3 | interval in minutes between connection retry attempts  
Valid values are 0-999. A value of 0 disables the reconnection attempt.  
The default value is 3.  
Certain platforms have a TCP/IP parameter that delays the closing of  
TCP/IP connections. In Solaris, this parameter is named  
tcp_close_wait_interval. GME/MVS cannot reconnect while the  
connection is in this close_wait state, because connectAAO sees it as a  
duplicate connection. For information regarding this parameter, refer  
to your platform-specific documentation. |
| TGTTYPE=[CELL | optional for each BEM connection that specifies whether the  
ADAPTOR connection is to a BEM cell or to another BMC adapter  
The following values are valid:  
■ CELL: is a BEM cell  
■ ADAPTOR: is an adaptor that does not have rules for managing  
synchronization with alerts  
The default is CELL. The parameter is valid only for TGTIP definitions  
for the client cell definitions. |
| TRACEAPP=ERROR | minimum severity of application trace records to create  
Valid values are NONE, ERROR, WARN, INFO, or DEBUG and are  
specified separately for both the local and remote GME nodes. The  
default value is ERROR. |
| TRACEBUF=[50 | amount of storage, in thousands of bytes, to allocate for the in-storage  
wraparound trace buffer  
The maximum value is 99999. The default value is 50. |
| TRACEGME=[NONE | minimum severity of GME trace records to create  
Valid values are NONE, ERROR, WARN, INFO, or DEBUG and are  
specified separately for both the local and remote GME nodes. The  
default value is ERROR. |
| TRACELK=[YES | specifies whether to perform a DNS lookup when logging IP  
addresses in the Trace Table  
Valid values are YES and NO. Specifying TRACELK=YES may  
lengthen the response times because a DNS lookup involves a  
network communication. The default value is NO. |
| TRACESEC=[NONE | minimum severity of security trace records to create  
Valid values are NONE, ERROR, WARN, INFO, or DEBUG and are  
specified separately for both the local and remote GME nodes. The  
default value is WARN. |
| TRACEZN=[LOCAL | time zone to use for time stamps in the Trace Table  
Valid values are GMT and LOCAL. The default value is LOCAL. |
Parameter | Description
--- | ---
WTO=[Y | N] | specifies whether BMC Impact Manager users can issue a write-to-operator message (WTO) through MainView AutoOPERATOR

**BBPARM member AAOMQL00 parameters**

Use the BBPARM member AAOMQL00 parameters TYPE, QMGR, and QUEUE to specify IBM MQ queues that are eligible for automation.

A fourth optional parameter, OPEN, indicates how the queue should be processed. These parameters must all be typed on one line; the syntax is

```
TYPE(INCL|EXCL) QMGR(queuemanagername) QUEUE(queuename) OPEN(S,I )
```

**Note**

This member supports system variables substitution. For more information, see “System variables substitution in parameter library members” on page 284.

---

Table 27: BBPARM member AAOMQL00

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>**TYPE</td>
<td>T [INCL</td>
</tr>
<tr>
<td>**QMGR</td>
<td>M (queue manager name)**</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>QUEUE</td>
<td>specifies the 48-character name of the queue that MainView AutoOPERATOR makes eligible for automation. The wildcard characters asterisk (<em>) and plus (+) are supported. This parameter is required and has no default value. You must specify a queue name, or a partial queue name with wildcards (a plus (+) represents one character and an asterisk (</em>) represents one or more characters). QUEUE can be abbreviated to U; for example you can specify QUEUE(SYSTEM.ADMIN.QMGR.EVENT) or U(SYSTEM.ADMIN.QMGR.EVENT)</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>OPEN</td>
<td>specifies how a queue is opened and what happens to the messages in the queue after it is opened; possible values are as follows: option 1:</td>
</tr>
<tr>
<td>[EXCLUSIVE</td>
<td>E]: specifies that the queue should be opened with the MQOO_INPUT_EXCLUSIVE open option Other applications are unable to open the queue while MainView AutoOPERATOR has it open. It also means that if another application has the queue opened already, MainView AutoOPERATOR cannot open it. The abbreviation is E.</td>
</tr>
<tr>
<td>[SHARED</td>
<td>S]: specifies that the queue should be opened with the MQOO_INPUT_SHARED option This subparameter is the default for option 1. The abbreviation is S. option 2:</td>
</tr>
<tr>
<td>[PROCESSOLD</td>
<td>P]: specifies that all messages found on the queue when it is opened should be routed through the Rule Processor to allow automation to take place This option is useful for processing messages that were put on the queue while the queue was not opened by MainView AutoOPERATOR. The abbreviation is P.</td>
</tr>
<tr>
<td>[IGNOREOLD</td>
<td>I]: specifies that MainView AutoOPERATOR does not process the existing messages on the queue and no automation takes place for those messages The abbreviation is I. This subparameter is the default for option 2.</td>
</tr>
</tbody>
</table>

Abbreviations can be used; for example: OPEN(EXCLUSIVE,PROCESSOLD) or O(E,P) indicates that the queue opened with the MQOO_INPUT_EXCLUSIVE option, and that all existing messages found on the queue at open time will be routed through the Rule Processor for automation OPEN(SHARED,IGNOREOLD) OR O(S,I) indicates that the queue is opened with the MQOO_INPUT_SHARED option, and that all existing messages found on the queue at open time will be ignored.
BBPARM member AAONCD00 is the Access NV BBPARM member. Table 28 on page 304 lists the parameter that is supported by AAONCD00.

**Note**
This member supports system variables substitution. For more information, see “System variables substitution in parameter library members” on page 284.

<table>
<thead>
<tr>
<th>Table 28: BBPARM member AAONCD00</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
</tr>
</tbody>
</table>
| DEBUG=[YES | NO] | the only supported parameter  
For normal operations, leave this parameter to the default setting of NO. |

**BBPARM member AAOPLX00 parameters**

BBPARM member AAOPLX00 contains parameters, some of which you must modify, when you use the Rules Management application.

**Note**
This member supports system variables substitution. For more information, see “System variables substitution in parameter library members” on page 284.

<table>
<thead>
<tr>
<th>Table 29: BBPARM member AAOPLX00</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter name</strong></td>
</tr>
</tbody>
</table>
| AO_NAME             | *(required)* provides the name to identify this PAS within the AOPlex  
This name is used when identifying the PAS in the Valid PAS List and in AGS messages and reports.  
This value must be 8 characters long. The name can contain only alphanumeric characters. No special characters are allowed.  
**Example:** AO_NAME=AGSPROD1 |
| AOPLEX_NAME         | *(required)* provides the name of the AOPlex to which this PAS belongs  
This name must be the same for all PASes that share the same registry.  
This value must be 8 characters long. This name can contain only alphanumeric characters. No special characters are allowed.  
**Example:** AOPLEX_NAME=AOPLEX01 |
<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Description</th>
</tr>
</thead>
</table>
| AOPLEX_XCF_GROUP    | *(required)* provides the name of the XCF group that the AOPlex will use to communicate  
This name must be the same for all PASes that share the same registry dataset.  
This value must be 8 characters long. This name can contain only alphanumeric characters. No special characters are allowed.  
This value cannot be the same as AOPLEX_NAME and cannot be the same XCF group name specified in the XCFGROUP= parameter in BBPARM member BBISSPxx.  
**Example:** AOPLEX_XCF_GROUP=AOPLXCF1 |
| REGISTRY_DSN        | *(required)* is the data set name of the linear VSAM data set allocated for use by this AOPlex.  
This value can be 1 to 44 characters long. |
| REGISTRY_WARNHWM    | *(optional)* is the Registry high-water-mark percentage warning message threshold  
If specified with a non-zero value, the AGSRY0014W warning message is issued when the percentage of the Registry space that is formatted for use exceeds the value and either of the following situations is true:  
  ■ The PAS is being initialized  
  ■ The registry 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 AGSRY0014W was issued.  
Specifying a value of 0 means that no monitoring will be done and no messages are issued.  
This value can be 0 to 99. If you do not specify a value, the default value is 0. |
<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Description</th>
</tr>
</thead>
</table>
| USAGE_MODE_RULES            | *(required)* informs the PAS how to use the registry  
Valid values are:  
**OFF**: the MainView AutoOPERATOR PAS will not use a registry for Rules in any way. Rules Management is not functioning for the PAS. Rules are read from the BBIPARM DD concatenation.  
**ADMIN**: the MainView AutoOPERATOR PAS opens the registry and allows you to perform administrative tasks, which the registry records. However, MainView AutoOPERATOR is not using the Rules from the registry for any automation in the PAS. Rules are read from the BBIPARM DD concatenation.  
**FULL**: the MainView AutoOPERATOR PAS uses the Rules from the registry, allows you to perform administrative tasks on the registry, and also reads from the registry to supply all the Rules and Rule Sets for automation in the PAS. No Rules are used from the BBIPARM DD concatenation and the full-screen Rules editor is disabled.  
**Note**: When switching to FULL mode, you must perform a COLD start of the PAS to access all of the Rules from the registry.  
The default is OFF. |
| AGSLOG                       | *(optional)* specifies whether the MainView AutoOPERATOR Generalized Services (AGS) messages are sent to a logger  
Valid values are:  
**YES**: (recommended) messages are sent to a logger. If the logger becomes unavailable for any reason, and AGSJRNL=NO and FORCEWTO=NO are also specified, the value on the AGSLOG_FAILOVER parameter determines what happens to the message.  
**NO**: messages are not be sent to a logger |
| AGSLOG_FAILOVER             | *(optional)* specifies what happens to AGS messages when AGSLOG=YES, FORCEWTO=NO, and AGSJRNL=NO are specified and the logger is not available  
Valid values are:  
**JRNL**: send the message to the BBI-SS journal. JRNL is the default.  
**WTO**: send the message as a WTO to the PAS JES joblog.  
**NONE**: do not write any messages to any destination |
| FORCEWTO                    | *(optional)* specifies whether AGS product messages are sent to the PAS JES job log  
Valid values are:  
**YES**: messages are sent to the PAS JES job log.  
**NO**: messages are not sent to the PAS JES job log. NO is the default. |
<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Description</th>
</tr>
</thead>
</table>
| AGSJRNJL                | *(optional)* specifies whether AGS product messages are sent to the BBI-SS journal.  
  Valid values are:  
  **YES**: messages are sent to the BBI-SS journal.  
  **NO**: messages are not sent to the BBI-SS journal. NO is the default. |
| FORCE_CLASSIC_RSNAMES   | *(optional)* specifies whether Rule Sets created in the registry by the user address space (UAS) follow the classic Rule Set naming that requires the name to start with RUL or AAORUL.  
  Valid values are:  
  **YES**: the names of Rule Sets created in the registry by the UAS are restricted to a Rule Set naming standard and the Rule Set name must begin with RUL or AAORUL. This ensures these Rule Sets can be exported and used in a BBPARM data set without a registry. YES is the default.  
  **NO**: the names of Rule Sets created in the registry by the UAS are not restricted to start to begin with RUL or AAORUL. Therefore, these Rule Sets cannot be used as BBPARM Rule Sets even if you export them and place them in a valid BBPARM data set unless you manually rename them to use RUL and AAORUL. |
| AUDIT_STARTUP_RULESETS  | *(optional)* causes message AGSIA4050I to be generated to the PAS syslog during a PAS COLD start to list the rulesets that the PAS will use and is similar to the PROCESS LIST= parameter in BBPARM member AOPRMxx.  
  Valid values are:  
  **YES**: produce the AGSIA4050I message(s) during PAS COLD start.  
  **NO**: do not produce the AGSIA4050I message(s) during PAS COLD start. |

**BBPARM member AAOPRM00 parameters**

BBPARM member AAOPRM00 is used by all MainView AutoOPERATOR products as a general purpose member for defining Operator Request parameters and specifying which IBM MQ queues MainView AutoOPERATOR will monitor.

*Note*  
This member supports system variables substitution. For more information, see “System variables substitution in parameter library members” on page 284.

Table 29 on page 304 lists the parameters for AAOPRM00.
### Table 30: BBPARM member AAOPRM00

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| AOLOGACT=\[NO | YES\] | specifies whether the actions taken by a Rule will be sent to the Automation Logger application  
To send all actions taken by a Rule to the Automation Logger application, specify YES.  
Default value is NO; valid values are YES or NO. |
| AOLOGEV=\[NONE | ALL | \(event\_type\_1, event\_type\_2, \ldots\)] | specifies one or more types of events that will be sent to the Automation Logger application  
Default value is NONE; valid values are ALL (all events are logged), ALRM, ALRT, BRC, CICS, CMD, DB2, EOM, EOS, EXT, IMS, JRNL, MQS, NVC, NVM, TIME, TLM, VAR. |
| AOOPTION=xxx       | (optional) allows you to limit the licensed MainView AutoOPERATOR product components that are started during BBI-SS PAS initialization.  
When this parameter is not specified, all MainView AutoOPERATOR components with valid BMC passwords are activated at the time of BBI-SS PAS start up.  
The format of the statement to activate one component is  
\(AOOPTION=xxx\)  
To activate more than one component:  
\(AOOPTION=(xxx,yyy,\ldots,zzz)\)  
For example:  
\(AOOPTION=(IAO,CAO)\)  
For best results, perform a cold start of the BBI-SS PAS after you make changes to the AOOPTION parameter.  
The MainView AutoOPERATOR component abbreviations that are valid for this parameter are:  
- ANV - MainView AutoOPERATOR Access NV  
- TSH - MainView AutoOPERATOR TapeShare  
- CAO - MainView AutoOPERATOR for CICS  
- IAO - MainView AutoOPERATOR for IMS  
- QAO - MainView AutoOPERATOR for MQ  
- MAO - MainView AutoOPERATOR for z/OS  
**Note:** You must specify the PRODUCT=AAO parameter in BBPARM member BBISSPxx to be able to activate any MainView AutoOPERATOR component. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| AOREPLAY = ![NO | YES] | specifies whether to send Event Replay records to the MainView Logger where you can review (replay) them at a later time  
   Default value is NO; valid values are YES or NO.  
   This keyword is valid only if AOLOGEV is not set to NONE. Event Replay records will be saved only for the event types specified on the AOLOGEV statement.  
   The MVLOGSS parameter in BBPARM member BBISSP00 specifies the SSID of the MainView Logger for the EMLOG records. |
| BIIZJRNL=[Y | N] | specifies that messages that describe BMC Impact Integration for z/OS product (Biiz) events are automatically sent to the BBI Journal  
   The default is N (no).  
   Specifying YES means that multiple diagnostic messages are written to the BBI Journal for events and MainView AutoOPERATOR ALERTs that are sent and published to a BMC Impact Manager (BMC IM) cell.  
   The messages show diagnostic information about the event such as target, CLASS, severity, and other event information.  
   It is also possible to write messages to the BBI Journal for individual events or ALERTs by specifying one of the following values:  
   - Y (for YES) in the Journal Detail field of the Rule definition  
   - JOURNAL(Y) in the following commands:  
     - IMFEXEC BIM EVENT  
     - AOEXEC ALERT  
     - IMFEXEC ALERT  
   **Note:** BMC warns that by setting this parameter to YES that the volume of messages sent to the BBI Journal might be large. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| DBCTLCMD=[IMS | MVS] | when specifying IMS (default value), requires the MainView AutoOPERATOR for IMS product to be active in the PAS, and the PAS is connected to IMS DBCTL. With this setting, DBCTL commands (issued from the Command line or from EXECs or Rules) are processed as IMS commands and you do not need to specify a target. Generic DBCTL commands are also supported. You must specify DBCTLCMD=MVS when you want to issue commands to IMS DBCTL and the MainView AutoOPERATOR for IMS product is not active within the PAS. With this setting, DBCTL commands must be issued from EXECs using the IMFEXEC CMD TYPE(IMS) statement and in addition:  

- the DBCTL keyword is required to identify the target  
- commands are issued as MVS commands using the DBCTL command character  

When issuing DBCTL commands from Rules, you must specify the commands as MVS commands. You can also specify DBCTLCMD=MVS for compatibility with automation written earlier than MainView AutoOPERATOR version 7.2. This setting ensures that responses to IMFEXEC CMDsatatements are returned to the EXEC in the same format as earlier than version 7.2. You should try to migrate your EXECs to use the current settings as time permits.  

**Note:** Specifying DBCTLCMD=MVS will disable some product functionality as described in the following list:  

- Commands issued from the Journal command line require a target and their response is returned to the Local Log.  
- Responses to commands issued with IMFEXEC CMD (with response) within an EXEC are returned in a different format. For more information, refer to the MainView AutoOPERATOR Advanced Automation Guide.  
- DBCTL commands issued with IMFEXEC CMD requires the use of the DBCTL parameter to specify the target.  
- IMS generic commands issued against DBCTL will fail.  
- Any Rules written with an action to issue DBCTL commands must issue those commands as MVS-type commands with the DBCTL command recognition character.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBCTLRG=[NO</td>
<td>YES]</td>
</tr>
<tr>
<td>DZDSN= [V5AM data set name]</td>
<td>specifies the name of the VSAM data set generated by the BMC product BMC Discovery for z/OS The VSAM data set name contains the Configuration Alias names that are discovered on the z/OS image by the BMC Discovery for z/OS product. The BMC Impact Integration for z/OS product uses these names to populate SHARED variables such as the following where sssssssss resolves to the name of the server address space:&lt;br&gt;&lt;br&gt;■ QCIA.QSERVER.sssssssss resolves to the configuration item (CI) alias value.&lt;br&gt;■ QCIN.QSERVER.sssssssss resolves to the CI name which is the same as CI alias.&lt;br&gt;■ QCIO.QSERVER.sssssssss resolves to the CI object class. The BMC Impact Integration for z/OS product attempts to read the DZDSN value every hour.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| HONORMPF=\[YES | NO] | specifies if the Rules Processor application honors MPF suppression  
Default is N; valid values are Y(es) or N(0).  
The HONORMPF parameter setting determines whether messages are  
handled by the Rule Processor when MPF suppression is active. |
| IMFDATE         | specifies the format of the IMFDATE variable  
The following list shows valid formats and an example of each:  
- YYYY/MM/DD; 2050/10/23  
- YYYYMMDD; 20501023  
- DD/MM/YYYY; 23/10/2050  
- DDMMYYYY; 23102050  
- MMDDYYYY; 10232050  
- MM/DD/YYYY; 10/23/2050  
- MMMDDYYYY; OCT232050  
- DDMMMYYYY; 23OCT2050 |
| IMFJNUM=\[V | 5] | specifies whether the variable IMFJNUM will be 5 digits in length with leading zeros or a variable length number without leading zeros  
You must specify IMFJNUM=V to support JES job numbers greater than 99,999.  
Default value is V; possible values are 5 or V.  
Refer to “IMFJNUM option and seven-digit JES job numbers (z/OS version 1.2 and later)” on page 249 for more information. |
| IMFORGN=LTERM | xxxxxxxxx | specifies that the LOCAL variable IMFORGN is set to the IMS LTERM name for IMS command and response  
Default is the VTAM node name.  
**Note:** The value of IMFORGN is sometimes used by security applications to check whether actions that are specified within an EXEC are allowed. Most actions within an EXEC are not individually checked for explicit authorization. Note that there are some exceptions such as SYSPROG Services commands, which are checked for proper authority before the SYSPROG Services commands can be issued.  
BMC recommends that you explicitly define the users who have authorization to schedule EXECs. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| IMSAUTH=\[NONE | USER] \] | *(optional)* specifies whether the User ID should be passed to IMS along with all IMS commands issued from MainView AutoOPERATOR.  
Valid values are as follows:  
  - NONE: (default) the user ID is not passed with IMS commands  
  - USER: the user ID will be passed to IMS  
  
  Standard IMS command security rules will be applied. For commands that are issued from the User Address Space (UAS), the user ID is the issuer of the command. For commands that are issued from automation such as Rules and EXECs, the user ID value is the user ID associated with the MainView AutoOPERATOR PAS.  
  
  This parameter applies to:  
    - Type-2 (IMSplex) commands  
    - Type -1 for IMS DBCTL with DBCTLCMD=MVS  
    - commands issued from the Journal  
    - line commands from IMS OPERATOR displays  
    - commands issued by Rules and EXECs  
  
  This parameter does not apply to IMS commands:  
    - Type-1 for IMS DBCTL with DBCTLCMD=IMS  
    - Type-1 for IMS DB/DC region  
  
  **Note:** This parameter is not related to MainView windows-mode or full-screen mode security. |

Appendix E  BBPARM data set members for MainView AutoOPERATOR 313
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMSCMDCN=[NO</td>
<td>YES]</td>
</tr>
<tr>
<td></td>
<td>- NO: (default) no confirmation panel is displayed</td>
</tr>
<tr>
<td></td>
<td>- YES: specifies that a confirmation panel is displayed for all IMS action commands that are manually issued from MainView AutoOPERATOR views or from the Journal.</td>
</tr>
<tr>
<td></td>
<td>A confirmation panel is displayed for both Type-1 and Type-2 commands.</td>
</tr>
<tr>
<td></td>
<td>A confirmation panel is not displayed for non-action commands such as /DISPLAY and QUERY or for commands issued by automation (such as Rules or EXECs).</td>
</tr>
<tr>
<td></td>
<td>You must ensure that the User Address Space (UAS) is locally connected to the MainView AutoOPERATOR PAS where this parameter is specified.</td>
</tr>
<tr>
<td>IMSPLAUD=[NO</td>
<td>YES]</td>
</tr>
<tr>
<td></td>
<td>- NO: (the default) audit messages are not issued.</td>
</tr>
<tr>
<td></td>
<td>- YES: audit messages, containing the user ID, are issued to the Journal for all IMSplex commands. For commands that originate from the user address space (UAS), the user ID is the issuer of the command. For commands that MainView AutoOPERATOR automation (such as Rules and EXECs) issues, the user ID is the MainView AutoOPERATOR PAS user ID.</td>
</tr>
<tr>
<td>IMSOMEX=[NO</td>
<td>YES]</td>
</tr>
<tr>
<td></td>
<td>- NO: The AutoOPERATOR IMS OM Input user exit does not broadcast IMS Type-2 commands to this PAS. IMP event type Rules do not fire.</td>
</tr>
<tr>
<td></td>
<td>- YES: (the default) when the MainView AutoOPERATOR for IMS product is active, the MainView AutoOPERATOR IMS OM Input user exit broadcasts IMS Type-2 commands to this PAS and IMP event type Rules will capture and automate these commands.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IMSPLTIM = [30</td>
<td>specifies a user-defined timeout limit for waiting for a response from IMSplex commands that are issued from the Journal.</td>
</tr>
<tr>
<td>5 - 9999]]</td>
<td>The default timeout limit is 30 seconds. Possible values range from 5 to 9999 seconds. The IMSplex command timeout limit represents the maximum time to</td>
</tr>
<tr>
<td></td>
<td>wait for a response from IMS. If a response is received from all IMS regions of the IMSplex before the timeout is reached, the response is displayed and</td>
</tr>
<tr>
<td></td>
<td>waiting is terminated. If a response is not received from all IMS regions of the IMSplex before the timeout is reached, a partial response is displayed and the IMS Operations Manager terminates execution of the IMSplex command.</td>
</tr>
<tr>
<td>IMSROUTE = [ALL</td>
<td>specifies the default routing for IMSplex commands issued from the MainView AutoOPERATOR Journal, Rules, or EXECs. Possible values are ALL and LOCAL. Default is ALL.</td>
</tr>
<tr>
<td>LOCAL]]</td>
<td>ALL: IMSplex commands are sent to all IMS regions of the IMSplex. LOCAL: IMSplex commands, issued without a ROUTE() keyword are sent only to the IMS region connected to the MainView AutoOPERATOR PAS.</td>
</tr>
<tr>
<td></td>
<td>This parameter does not apply when:</td>
</tr>
<tr>
<td></td>
<td>■ the ROUTE() parameter is specified with an IMSplex command</td>
</tr>
<tr>
<td></td>
<td>■ The IMSPLEX() parameter on an IMSplex command from EXEC specifies IMSplex that does not include the IMS region connected to the MainView AutoOPERATOR PAS.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IMSMSTIM=[5</td>
<td>(optional) specifies the maximum time to wait between receiving segments of IMS multisegment messages. Arrival of the final segment terminates the wait and the message is processed by the Rules processor. If the time is exceeded without receiving a final segment from IMS, the wait is terminated and any future segments are ignored. The delay may occur due to system heavy workloads, a lower dispatching priority of the IMS issuing the message, or other IMS processing. The default timeout limit is 5 seconds. Possible values range from 1 to 999 seconds. Specify this parameter when:</td>
</tr>
<tr>
<td>1 - 999]</td>
<td>- Rules automation is not processing the complete IMS multisegment message(s). For example, if only the first message segment is logged on the Journal.</td>
</tr>
<tr>
<td></td>
<td>- EXECs do not receive the complete IMS command response and IMFCC is set to 8.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Increasing the default value might create a delay of Rules-based automation for IMS multisegment messages.</td>
</tr>
<tr>
<td>IMSSHTIM=[2</td>
<td>(optional) specifies the internal time to wait for termination of the MainView AutoOPERATOR tasks in the IMS control region address space during IMS shutdown. The default timeout limit is 2 seconds. Possible values range from 1 to 999 seconds. Specify this parameter only if your IMS control region takes unusually long to terminate or frequently experiences abend sA03 at shutdown. <strong>Note:</strong> Increasing the default value reduces the possibility of sA03 abends, but might delay the time it takes for IMS to shut down.</td>
</tr>
<tr>
<td>1 - 999]</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| MAXDLYQ=\[100 | nnn\]  | *(optional)* specifies the maximum concurrent number of delayed requests from Rules. A delayed request can be specified for a command or message. The default is 100; valid values are 0 to 999. Specifying zero means no maximum is set. The maximum value is 999 or half the value of the MAXPQE parameter specified in the BBPARM member BBISSPxx member, whichever is lower. When the number of concurrent DELAYED requests reaches 60 percent of the MAXDLYQ threshold, the following message is issued:  
|                      | DY6160W  xxxxxx Delayed Requests queued                                                                                                                                                                   |
|                      | When the number of concurrent DELAYED requests reaches 75 percent of the MAXDLYQ threshold, the following message is issued:  
|                      | DY6175W  xxxxxx Delayed Requests queued                                                                                                                                                                   |
|                      | If the MAXDLYQ value is reached, no additional requests can be queued until the number drops below the specified threshold and the following message is issued:  
<p>|                      | DY6199E  Maximum Delay Request queued                                                                                                                                                                      |
|                      | The MAXDLYQ threshold calculations are re-adjusted when the AAOPRMxx member is reset. The threshold is calculated every time a DELAYED request is queued. Delayed actions are not retained over a warm or cold restart of the BBI-SS PAS. Therefore if outstanding delayed actions exist when the BBI-SS PAS is restarted, those delayed actions are lost and are not re-queued when the BBI-SS PAS restarts. You can use the BBI command .DISPLAY DLY before you restart the BBI-SS PAS to review any delayed actions that have not yet been completed. Determine whether you should delay restarting the BBI-SS PAS or, restart the BBI-SS PAS but note that you need to reissue the delayed actions when the BBI-SS PAS restarts. |
| MPFFILTR=(AUTO=YES | NO | token,SUP=YES | NO)  | specifies whether MainView AutoOPERATOR checks WTORs for the MPF settings for the AUTO and SUP keywords. <strong>Note:</strong> This parameter cannot be used when HONORMPF is set to HONORMPF=YES. For information about how this parameter is set and how it affects automating WTOs, refer to “Implementing support for MPF” on page 135. |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| MQALTUSR=[RACF | *user ID]* | specifies an AlternateUserid to be specified when issuing MQSC or PCF MQ commands from MainView AutoOPERATOR  
The user ID is used when MQ commands are issued from an EXEC or from a Rule.  
There is no default; valid values are as follows:  
**RACF** specifies that the security ID of the BBI-SS PAS is used as the AlternateUserid for commands.  
**user ID** specifies that a valid user ID is used as the AlternateUserid.  
This global parameter can be overridden with the ALTUSER() | AU() parameter of the IMFEXEC CMD TYPE(MQS) command or as an action from a MQ initiated Rule. When you use this parameter the MQ PUT1 options will include  
MQOO_ALTERNATE_USER_AUTHORITY,  
MQPMO_DEFAULT_CONTEXT and  
MQPMO_ALTERNATE_USER_AUTHORITY and the AlternateUserid will be placed in the object descriptor.  
Refer to the *MainView AutoOPERATOR for MQ Installation and User Guide* for more information about the IMFEXEC CMD TYPE(MQS) and issuing MQ commands from Rules. |
| MQAUPRMS=[Y | N] | specifies whether individual MQSC or PCF MQ commands that are issued from EXECs or Rules can use the ALTUSER parameter to override the global parameter MQALTUSR (when specified) and specify an AlternateUserid  
When no value is specified for the MQALTUSR parameter, this parameter allows or denies the use of AlternateUserid for MQ command.  
Default value is NO, which means that the ALTUSER value is ignored. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| MQEV=\[YES | NO | or any combination of SLIPRFMCVBX] | specifies that MainView AutoOPERATOR for MQ should automatically enable instrumentation events for a queue manager during connection if it is not already enabled. You can specify that all the events are enabled, that none of the events are enabled, or that selected event types are enabled. When you specify multiple event types, you must code them consecutively, without using blanks or commas (does not apply if you are specifying Y or N). Valid values are as follows:  
  - N: do not enable any events  
  - Y: enable all events  
  - S: enable start/stop (strstpev) events  
  - L: enable local (localev) events  
  - I: enable inhibit (inhibtev) events  
  - P: enable performance (perfmev) events  
  - R: enable remote (remoteev) events  
  - F: enable configuration (configev) events  
  - M: enable command (cmdev) events  
  - C: enable channel (chlev) events  
  - V: enable ssl (sslev) events  
  - B: enable bridge (bridgeev) events  
  - X: enable non-display command events (CMDEV(NODISPLAY))  
The default value is NO. |
| MQEVLPRC= xxxxxxxx | specifies the name of the IBM MQ Event Listener PROC  
Do not use this parameter if the IBM MQ Event Listener is not required. Refer to the MainView AutoOPERATOR for MQ Installation and User Guide for more information about the IBM MQ Event Listener and coexistence of MainView AutoOPERATOR and other BMC Software products that require the IBM MQ event queues. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| MQGINHIB=[ xxxxxxxx | specifies the action MainView AutoOPERATOR should take when MainView AutoOPERATOR attempts to listen to a queue which is defined as GET(DISABLED). Possible settings are  
JRNL: Issue a message to the BBI Journal stating that MainView AutoOPERATOR cannot listen to the queue.  
WTO: Issue a write-to-operator (WTO) message stating that MainView AutoOPERATOR cannot listen to the queue.  
IGNORE: Take no action.  
ALTER: Alter the queue to GET(ENABLED).  
The default value is JRNL. |
| MQNSHARE=[ xxxxxxxx | specifies the action MainView AutoOPERATOR should take when MainView AutoOPERATOR attempts to listen to a queue which is defined as NOSHARE. Possible settings are  
JRNL: issue a message to the BBI Journal stating that MainView AutoOPERATOR cannot listen to the queue.  
WTO: issue a write-to-operator (WTO) message stating that MainView AutoOPERATOR cannot listen to the queue.  
IGNORE: take no action.  
ALTER: alter the queue to share.  
The default value is JRNL. |
| MSGJOBNM=[ACTUAL|ORIGIN] | specifies how MainView AutoOPERATOR should derive batch, Started Task, and TSO job names for MSG events  
The job name or the IMFOJOB variable can be used in the Rule Processor Selection Criteria panel while defining Rules. The following values are valid:  
ACTUAL: Use the job name as it is derived from the address space that creates the event. There are situations where this value might not match the address space that initiated the event (WTO). For example some JES3 messages will have a value of "CONSOLE".  
ORIGIN: Use the job name as it is derived from the MVS control block (WQE). Specifying ORIGIN would result in a more accurate name of the issuer of the WTO.  
The default value is ORIGIN. |
### Parameter Description

#### MVIMSDB=\[MSG | CMD\]

*(optional)* specifies whether IMS DBCTL commands issued from MainView for IMS windows mode views will result in a Journal message or actual command.

The following values are valid:

- **MSG**: IMS DBCTL commands from MainView for IMS views sends the BBFPAC0XI message to the Journal. You must enable the RULIPX00 Rule set to detect this message and issue the IMS DBCTL command as an MVS command.
- **CMD**: MainView AutoOPERATOR issues the actual IMS DBCTL commands when they are entered from MainView for IMS views. All applicable AAOPRMxx parameters (such as DBCTLCMD) will apply.

**Note**: Changing the value of the MVIMSDB parameter requires that you restart the IMS DBCTL region.

The default value is **MSG**.

#### QALIAS=\[Y | N\]

specifies whether MainView AutoOPERATOR for MQ should open and listen for messages on QALIAS type queues

Note that if you specify an alias queue name in BBPARM AAOMQL00, in addition to specifying a base queue, any Rules that you have written for the same message for the base queue might also fire for the alias queue.

For example, if the selection criteria for a Rule for messages going to the base queue does not specifically contain the queue name then the Rule might fire twice; once for the base queue and once for the alias queue.

#### RTCSCMD=\[MVS_startCommand\_for\_RTCS\]

specifies the MVS command start command for BMC Runtime Component System (RTCS)

This parameter requires that you have the MainView for z/OS product component installed in the BBI-SS PAS.

For example:

```
RTCSCMD=S OSZINIT.RTCS
```

Do not use quotes and the length of the command is limited to column 72. This line does not support comments or continuations.

When MainView AutoOPERATOR for z/OS PAS is started and the RTCSCMD parameter is specified, MainView AutoOPERATOR tests for the presence of RTCS. If RTCS is absent, MainView AutoOPERATOR issues the MVS command to start RTCS specified with this parameter.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| RULEPROT      | activates serialization on a Rule when it is edited, preventing multiple users from updating the same Rule, in the same BBI-SS PAS at the same time. When the parameter is not specified, the default is NO. The following values are valid:  
  - **NO**: no serialization is performed when a Rule is edited  
  - **YES**: serialization is performed within the scope of each PAS to ensure only one user is updating a Rule in that PAS as the same time |
| RULESAVE      | specifies what happens when a Rule Set is saved from the Rule Processor Detail Control panel. Valid options are as follows:  
  - **FIRST** (default): the Rule Set is automatically saved in the first data set of the BBIPARM DD concatenation, regardless of where it was read from  
  - **ORIGINAL**: the Rule Set is automatically saved in the data set that it was originally read from in the BBIPARM DD concatenation  
  - **PROMPT**: specify the data set (FIRST or ORIGINAL) to which to save the Rule Set |
| RULESCAN      | specifies the automation strategy of F (FIRST), A (ALL), or I (INDIVIDUAL) to determine how Rule Sets and Rules are searched to match events  
  - **F | FIRST**: indicates only the first Rule with matching selection criteria is fired.  
  - **A | ALL**: indicates Rules with matching selection criteria are fired.  
  - **I | INDIVIDUAL**: indicates that each Rule Set can have its own (individual) automation strategy. Each Rule Set can be set with First, or All.  
  The automation strategy specified here is used when the BBI-SS PAS is either warm or cold started. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RULESET= xx or</td>
<td>specifies one or more RULESETs to become active upon initialization of the BBI-SS PAS</td>
</tr>
<tr>
<td>RULESET=( xx,yy,...,zz)</td>
<td>Format of the statement is RULESET= xx for one set, or RULESET= (xx,yy,...,zz) for multiple sets. The two-character code is appended to</td>
</tr>
<tr>
<td></td>
<td>AAORUL.</td>
</tr>
<tr>
<td></td>
<td>You also can use the 5-character suffix (where the prefix is RUL); for example, RULESET=RUL xxxxx or RULESET=(RULxxxxx, RULyyyyy,</td>
</tr>
<tr>
<td></td>
<td>RULzzzzz ).</td>
</tr>
<tr>
<td></td>
<td>You can specify a long list of Rule Set names by continuing to the next line. Separate each Rule Set name with a comma.</td>
</tr>
<tr>
<td>SPLXVARS=[Y</td>
<td>N]</td>
</tr>
<tr>
<td></td>
<td>Default values is N; valid values are Y(yes) or N(no).</td>
</tr>
<tr>
<td>SPLXPREF=[1- to 16-</td>
<td>specifies the prefix that denotes that the SHARED variable name is a sysplex variable</td>
</tr>
<tr>
<td>character_userSpecifiedPrefix]</td>
<td>Default value is AOSPLX_; valid values are 1 to 16 alphanumeric characters long.</td>
</tr>
<tr>
<td></td>
<td>You can specify only one prefix for the MainView AutoOPERATOR PAS and, all of the MainView AutoOPERATOR PASs that use the same</td>
</tr>
<tr>
<td></td>
<td>SYSPLEX variables must also use the same prefix.</td>
</tr>
<tr>
<td></td>
<td>If you use the default setting, an example of a sysplex variable name would be AOSPLX_DB2STATUS.</td>
</tr>
<tr>
<td>SPLXSTR=[structure name]</td>
<td>specifies the name of the XCF Structure that stores the sysplex variables</td>
</tr>
<tr>
<td></td>
<td>The XCF Structure resides in a Coupling Facility.</td>
</tr>
<tr>
<td></td>
<td>Valid values can be 1 to 8 alphanumeric characters long.</td>
</tr>
<tr>
<td></td>
<td>SPLXSTR uses the default value of the XCFGROUP option in BBISSPxx.</td>
</tr>
<tr>
<td></td>
<td>BMC recommends that you avoid coding user-specified values for the XCFGROUP and SPLXSTR parameters and use the default values. The</td>
</tr>
<tr>
<td></td>
<td>default value for XCFGROUP is BMCAB and the default XCF Structure name is MVAO_BMCAB.</td>
</tr>
<tr>
<td>SUPREPLY=[NO</td>
<td>YES]</td>
</tr>
<tr>
<td></td>
<td>Specifying YES changes the text of the command issued by a terminal session (TS) user or EXEC. For example, if an EXEC issues the</td>
</tr>
<tr>
<td></td>
<td>command \texttt{R 33,XYZ} the BBI-SS PAS Journal shows the command as \texttt{R 33,*SUPPRESSED*}</td>
</tr>
<tr>
<td></td>
<td>Therefore, specify YES when you want to conceal text in the BBI-SS PAS Journal (such as the use of a password). The local Journal shows</td>
</tr>
<tr>
<td></td>
<td>the original message.</td>
</tr>
<tr>
<td></td>
<td>The default is NO.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| UCCICTRN=[Y | N]     | specifies whether CICSTRAN transaction parameters will be translated to uppercase  
This specification applies to CICSTRAN when executed from anywhere except the TS Command Execution Application (which supports mixed case specification of parameters).  
The following values are valid:  
N: specifies that CICSTRAN parameters are not translated to uppercase  
Y: specifies that CICSTRAN parameters are translated to uppercase  
The default value is Y.                                                                 |
| UCVARS=[NO | YES]   | specifies whether the Rules Processor should convert variable names to uppercase before resolving  
The following values are valid:  
NO: do not convert variable names to uppercase before resolving the variable value  
YES: convert the variable name to uppercase before resolving  
Specify UCVARS=YES if you want the Rules Processor to convert variable names to uppercase.  
Two passes are made when resolving compound variables, which might result in a mixed case name. If the variable name is mixed case, you must specify UCVARS=YES for that variable to be resolved.  
**Note:** When creating Rules that allow you to send events from MainView AutoOPERATOR to a BMC Impact Manager cell, be aware that the slot fields on the Event Action (BA) or Alert Action (AE) panels are case sensitive. Variable names specified on this panel are converted to uppercase before they are resolved when UCVARS=YES is specified. |
| WTOEARLY           | specifies that early WTOs, captured by the QAOCNZ15 utility, are available as HWTO events to the Rule Processor  
This parameter is valid only when the BMC MainView Automation Solution is active in the MainView AutoOPERATOR BBI-SS PAS. |
| WTOINTRM           | specifies that WTOs captured during brief MainView AutoOPERATOR outages are available as HWTO events to the Rule Processor  
This option is ignored if the MainView AutoOPERATOR BBI-SS PAS abends or does not terminate normally.  
This parameter is valid only when the BMC MainView Automation Solution is active in the MainView AutoOPERATOR BBI-SS PAS. |
BBPARM members AAORUL00 parameters

BBPARM members AAORUL00 contain sets of distributed MainView AutoOPERATOR Rule Sets for use with Rules. These Rule Set names are therefore reserved for use to distribute Rules.

The distributed members include

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAORULBA</td>
<td>contains a set of Rules that can be used to automate the startup and shutdown of IMS, DB2, and CICS</td>
</tr>
<tr>
<td>AAORULBC</td>
<td>contains a starter set of Rules that is used for and implemented by the MainView AutoOPERATOR application Continuous State Manager (CSM)</td>
</tr>
<tr>
<td>AAORULBx</td>
<td>contains sample Rules for the MainView AutoOPERATOR product where x is a predefined variable</td>
</tr>
<tr>
<td>AAORULCS</td>
<td>contains a starter set of Rules that can be used for the implementation of the MainView AutoOPERATOR application CSM</td>
</tr>
<tr>
<td>AAORULC1</td>
<td>contains sample Rules that intercept CICS and MainView for CICS messages</td>
</tr>
<tr>
<td>AAORULD1</td>
<td>contains sample Rules for DB2 solutions</td>
</tr>
<tr>
<td>AAORULD2</td>
<td>contains sample Rules for DB2 solutions</td>
</tr>
<tr>
<td>AAORULD3</td>
<td>contains sample Rules for DB2 solutions</td>
</tr>
<tr>
<td>AAORULD5</td>
<td>contains sample Rules for DB2 solutions</td>
</tr>
<tr>
<td>AAORULD6</td>
<td>contains sample Rules for DB2 solutions</td>
</tr>
<tr>
<td>AAORULD8</td>
<td>contains sample Rules for DB2 solutions</td>
</tr>
<tr>
<td>AAORULI1</td>
<td>contains sample Rules that generate ALERTs for IMS messages</td>
</tr>
<tr>
<td>AAORULJV</td>
<td>contains sample Rules to intercept messages and commands</td>
</tr>
<tr>
<td>AAORULM1</td>
<td>contains sample Rules for MainView Sample Solutions</td>
</tr>
<tr>
<td>AAORUL00</td>
<td>contains sample Rules for MainView AutoOPERATOR Rule Processor application</td>
</tr>
</tbody>
</table>

For more information about these Rule Set members, refer to the section about Rules in the *MainView AutoOPERATOR Basic Automation Guide, Volume 1: Using Rules.*
BBPARM member AAOTRN00 parameters

The following table lists the IMS transaction parameters for BBPARM member AAOTRN00.

Table 31: BBPARM member AAOTRN00

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RLTERM= [MASTER</td>
<td>LTERM name]</td>
</tr>
<tr>
<td>TRA[N=[transaction name] [USERID=[LTERM</td>
<td>AAOPAS]]</td>
</tr>
</tbody>
</table>

BBPARM member AAOTSP00 parameters

BBPARM member AAOTSP00 is used by the TapeSHARE for MainView AutoOPERATOR component.
Note
This member supports system variables substitution. For more information, see “System variables substitution in parameter library members” on page 284.

For information about using the TapeSHARE for MainView AutoOPERATOR component, refer to the MainView AutoOPERATOR Options User Guide; for documentation about how to use the MainView AutoOPERATOR application Dynamic Parameter Manager to update the parameters in AAOTSP00, refer to the MainView AutoOPERATOR Customization Guide.

Table 32 on page 327 lists the parameters for the MainView AutoOPERATOR TapeSHARE component.

Table 32: BBPARM member AAOTSP00

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTION=[DEFAULT</td>
<td>CANCEL</td>
</tr>
<tr>
<td>■ DEFAULT: allows the installation’s default action to occur</td>
<td></td>
</tr>
<tr>
<td>■ CANCEL: cancels the job</td>
<td></td>
</tr>
<tr>
<td>■ WTOR: issues a WTOR to the operator If the site does not have a defined default action, this action is the default.</td>
<td></td>
</tr>
<tr>
<td>■ NOHOLD: enables the job to wait without holding resources</td>
<td></td>
</tr>
<tr>
<td>■ HOLD: enables the job to wait while holding resources</td>
<td></td>
</tr>
<tr>
<td>Example: ACTION=WTOR</td>
<td></td>
</tr>
<tr>
<td>FREE=[DEALLOC</td>
<td>deallocation procedure name]</td>
</tr>
<tr>
<td>Example: FREE=RELEASE</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| **NOGIVE= device address**<br>Default value is none. | list of device addresses (or a range of addresses) that should not be GIVEN to other partners within the TapeSHAREplex when they request devices to TAKE (in other words, a list of device addresses that are dedicated to this image)  
*Note:* This parameter does not prevent you from manually GIVING a NOGIVE device.  
If a device address is not associated with this parameter, TapeSHARE assumes no devices are dedicated to this image.  
**Example:**  
NOGIVE=0123  
NOGIVE=0120-012F  
You also can use this parameter in conjunction with the NOTAKE parameter. By specifying a device address on both these parameters, the device will be a NOGIVE-NOTAKE device and it will be excluded completely from TapeSHARE control.  
**Example:**  
NOGIVE=0123  
NOTAKE=0123  
If the device addresses 0123 is specified on both the NOGIVE and NOTAKE parameters, the 0123 is completely excluded from TapeSHARE control. TapeSHARE does not attempt to GIVE or TAKE this device. |
| **NOTAKE= device address**<br>Default value is none. | list of device addresses (or a range of addresses) that this image should not TAKE when a request for resources is made. TapeSHARE on this image will not accept (TAKE) the devices specified. If a value is not associated with this parameter, TapeSHARE assumes that it can accept any tape devices when a request for resources is made.  
**Example:**  
NOGIVE=0123  
NOGIVE=0120-012F  
You also can use this parameter in conjunction with the NOGIVE parameter. By specifying a device address on both these parameters, the device will be a NOGIVE-NOTAKE device and it will be excluded completely from TapeSHARE control.  
**Example:**  
NOGIVE=0123  
NOTAKE=0123  
If the device address 0123 is specified on both the NOGIVE AND NOTAKE parameters, the 0123 device is completely excluded from TapeSHARE control. TapeSHARE does not attempt to GIVE or TAKE this device. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARTNER= subsystem name</td>
<td>BBI-SS PAS name (that can be associated with either one or a range of tape devices) that is a TapeSHARE PLEX partner. You also can include with the BBI-SS PAS name the device address (or a range of device addresses) that TapeSHARE cannot give to the named partner. (See examples below.) If no values are associated with this parameter, TapeSHARE assumes that any BBI node defined in BBINOD00 member is a partner within a TapeSHARE PLEX. <strong>Example:</strong> PARTNER=SYSA SYSA is a partner to this image. <strong>Example:</strong> PARTNER=SYSC, 0123 SYSC is a partner to this image and the device whose address is 0123 cannot be given to partner SYSC. <strong>Example:</strong> PARTNER=SYSB, 0120-0127 SYSB is a partner to this image and the devices whose addresses fall between 0120 and 0127 cannot be given to partner SYSB.</td>
</tr>
<tr>
<td>PREF=[YES</td>
<td>NO</td>
</tr>
<tr>
<td>Parameter</td>
<td>Definition</td>
</tr>
<tr>
<td>----------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| PREFDEV= *device address*  | specifies a tape device address (or a range of tape device addresses) followed by either an I (for input-only) or O for (output-only) that are identified to TapeSHARE on this image that are allocated to perform input-only or output-only functions.  
  Default value is none.  
  This parameter should be used in conjunction with the PREF parameter.  
  With the PREF activated, when a device is needed for an output-only operation, TapeSHARE attempts to choose from those devices identified by this parameter output-only.  
  If a value is not associated with this parameter, TapeSHARE does not invoke device preferencing.  
  **Example:**  
  ```
  PREFDEV=01A0-01A3,I  
  PREFDEV=0130,O  
  PREFDEV=0150-015F,O  
  ``` |
| RETRYCNT=(0 | 2 through 10) | number of times TapeSHARE attempts to satisfy a request for devices after an initial attempt has failed.  
  For example, suppose this system’s request for devices is not satisfied and 3 retries is specified. TapeSHARE attempts three more times to satisfy this request for devices.  
  **Example:** RETRYCNT=3  
  This parameter is used in conjunction with the retryint parameter which specifies how long TapeSHARE waits between retry attempts.  
  Possible values are 0 - 10. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>RETRYINT=(0</td>
<td>30 through 300)</td>
</tr>
<tr>
<td></td>
<td>For example, suppose</td>
</tr>
<tr>
<td></td>
<td>■ RETRYCNT= parameter is set to 3 retries</td>
</tr>
<tr>
<td></td>
<td>■ RETRYINT= parameter is set to 45 seconds</td>
</tr>
<tr>
<td></td>
<td>If this system’s request for devices is not satisfied, TapeSHARE waits 45 seconds after the initial attempt fails and retries to satisfy the request (retry attempt 1). If this attempt fails, TapeSHARE waits another 45 seconds and retries (attempt 2). If this attempt also fails, TapeSHARE waits another 45 seconds and retries (attempt 3). Should all three attempts fail, an Allocation Failed Event occurs and the specified action is taken. Example: RETRYCNT=3 Possible values are 0 - 300.</td>
</tr>
<tr>
<td>TIMEOUT=(0</td>
<td>120 through 999)</td>
</tr>
<tr>
<td></td>
<td>When this time is reached and a successful give has not been completed, an Allocation Failed Event occurs unless you have specified a number of retries on the retrycnt parameter. Example: TIMEOUT=20 Possible values are 0 - 999.</td>
</tr>
<tr>
<td>TRACE=[YES</td>
<td>NO]</td>
</tr>
<tr>
<td></td>
<td>Example: TRACE=NO</td>
</tr>
<tr>
<td></td>
<td>If YES is specified, the BBITST DD statement must appear in the BBI-SS PAS startup JCL for information to be written to it (see BBSAMP member SSJCL).</td>
</tr>
</tbody>
</table>

**BBPARM member BBIVAR00 parameters**

This BBPARM member defines any number of preset SHARED variables and their values to be set in the SHARED pool at the COLD start of the BBI-SS PAS COLD when VPOOL=RESET is specified.

**Note**

This member supports system variables substitution. For more information, see “System variables substitution in parameter library members” on page 284.
Table 33: BBPARM member BBIVAR00

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MYVAR999=ACTIVE</td>
<td>contains any number of variables that are to be set at BBI-SS PAS startup when VPOOL=RESET is specified with the startup parameters (VPOOL=RESET requires a COLD start)</td>
</tr>
<tr>
<td>MYVAR999 = active</td>
<td>These SHARED variables are set before the MainView AutoOPERATOR Rules processor is initialized. Therefore, Rules for the event type VAR will not match when these SHARED variables are initialized. However, these variables are available to be used in all other Rules.</td>
</tr>
</tbody>
</table>

**Note**

Sysplex variables are not supported in BBPARM member BBIVAR00. Do not specify sysplex variables in BBPARM member BBIVAR00 or unpredictable results might occur.

You can create more than one BBIVARxx member (where xx is the suffix for member) and the following rules apply to determine which BBIVARxx member is processed.

- You can request multiple BBIVARxx members with the BBPARM member BBICFG.
  - The name of the member is constructed from the literal BBIVAR and a suffix. The default suffix is 00. An alternative suffix member can be specified through configuration of the BBI-SS PAS with a BBCFG DD statement in the PAS JCL.
  - If a BBCFG DD statement is present and it contains a valid BBIVAR=xx statement, the BBIVAR= xx statement is processed.
  - If no valid BBIVAR= xx statement is found in the BBCFG configuration member, a suffix of 00 is assumed.
  - An example of specifying a specific BBIVARxx member in the BBCFG member is
    ```
    BBIVAR=(99)
    ```
    where 99 is the suffix for BBPARM member BBIVAR99.
    - To specify multiple BBIVARxx members
      ```
      BBIVAR=(99,23,12,15)
      ```
      where 99, 23, 12 and 15 are suffixes for BBPARM members BBIVAR99, BBIVAR12, BBIVAR12 and BBIVAR15.
— When the BBCFG configuration member has multiple BBIVAR= statements specified, the last one is the one that the system starts with.

For example, if the BBCFG member contains

BBIVAR=(99,12)
BBIVAR=(99,12,13)

the specification BBIVAR=(99,12,13) is the one the system will start up with.

■ Error or warning messages are issued for the following possible situations:

— BBCFG contains a valid BBIVAR= xx statement but the member was not found.

— BBCFG contains syntax errors. Syntax errors in the configuration member might cause the default BBIVAR00 to be processed.

— The BBIVAR xx member contains syntax errors.

The following syntax rules apply to processing within the BBIVARxx member:

Any number of entries can be specified.

■ Each line is restricted to 72 characters and entries cannot be continued over multiple lines.

■ Variables names can have a maximum length of 32 characters.

■ An asterisk in column one indicates a comment line and causes the line to be ignored.

■ An assignment statement can be in either of the following formats:

  VARIABLE = 'VALUE'
  VARIABLE = value

  The first format must be used if the variable value contains blanks; otherwise, blank spaces are ignored. For example:

  MYVAR = A B C - MYVAR is set to A
  MYVAR = 'A B C' - MYVAR is set to A B C

■ Variable names are always translated to uppercase letters.

■ Variable names must conform to TSO/E REXX syntax.

■ Variables are created in the variable pool exactly as specified. REXX stem variables do not experience any form of substitution. For example, the following specification creates a variable called A and a variable called I.A, not I.0:

A = 0
I.A = 5
Variables beginning with the letter Q cannot be assigned. Q-variables are BBI-SS PAS internal and have read-only access.

If the variable itself is to contain a single quotation mark ('), enclose the string in double quotation marks ("'). If the variable is to contain double quotation marks, enclose the string in single quotation marks (').

If any of the preceding conditions are not met, an error or warning message is issued, accompanied by an additional message that identifies the invalid line. Invalid entries are skipped, and processing continues.

Examples of valid entries:

TEST1 = 25

Test2 = 15

end_of_day = TRUE

COMMENT = 'SHIFT CYCLE'

TEST1.A = 5

Examples of invalid entries:

'TEST1' = 25

TEST1* = 255

COMMENT = 'SHIFT CYCLE (End quotation mark is missing)

QSMID = SYSB (Q-variables are reserved and read-only)
Sample data set members

The BBSAMP data set contains sample members that you can edit for your site’s use.

Sample members for MainView AutoOPERATOR only

The following table describes BBSAMP members used only for MainView AutoOPERATOR.

Table 34: BBSAMP data set members for MainView AutoOPERATOR

<table>
<thead>
<tr>
<th>BBSAMP member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$EXECJOB</td>
<td>indicates the EXEC testing batch job</td>
</tr>
<tr>
<td>$EXECTST</td>
<td>indicates the EXEC testing CLIST</td>
</tr>
<tr>
<td>$TESTDAT</td>
<td>indicates the EXEC testing sample test case</td>
</tr>
<tr>
<td>AOALRTDF</td>
<td>defines resources to RACF for MainView AutoOPERATOR ALERT queue name resources that control a TS-user’s authority to delete ALERTs and ALERT queues through the ALERT Detail and Stats panels</td>
</tr>
<tr>
<td>AOALRTPE</td>
<td>permits users to MainView AutoOPERATOR ALERT queue name resources that control a TS-user’s authority to delete ALERTs and ALERT queues through the ALERT Detail and Stats panels</td>
</tr>
<tr>
<td>AOAPPLDF</td>
<td>defines all MainView AutoOPERATOR application resources to RACF</td>
</tr>
<tr>
<td>AOAPPLPE</td>
<td>permits users to MainView AutoOPERATOR application resources</td>
</tr>
<tr>
<td>AOCMDDDF</td>
<td>defines all MainView AutoOPERATOR command-level resources to RACF</td>
</tr>
<tr>
<td>AOCMDPE</td>
<td>permits users to MainView AutoOPERATOR command-level resources</td>
</tr>
<tr>
<td>BBSAMP member name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>AOEXECDF</td>
<td>defines all MainView AutoOPERATOR EXEC name resources to RACF</td>
</tr>
<tr>
<td>AOEXECPE</td>
<td>permits users to MainView AutoOPERATOR EXEC name resources</td>
</tr>
<tr>
<td>AOANYDF</td>
<td>defines all MainView AutoOPERATOR AOAnywhere functions to RACF</td>
</tr>
<tr>
<td>AOANYPE</td>
<td>permits users to MainView AutoOPERATOR AOAnywhere functions</td>
</tr>
<tr>
<td>AOPARMDF</td>
<td>defines all MainView AutoOPERATOR DPM member name resources that control a TS-user’s authority to update, read, add, or activate a DPM member through the Dynamic Parameter Manager application</td>
</tr>
<tr>
<td>AOAPPLPE</td>
<td>permits users to all MainView AutoOPERATOR DPM member name resources that control a TS-user’s authority to update, read, add, or activate a DPM member through the Dynamic Parameter Manager application</td>
</tr>
<tr>
<td>AOSIVTBL</td>
<td>contains sample Control-O rules that you can copy into the appropriate Control-O / CMEM library and activate so that Set IOA Variable events from MainView AutoOPERATOR can be processed in Control-O</td>
</tr>
<tr>
<td>CAODTABU</td>
<td>contains macro to generate the CAODTAB table that allows specific Transient Data queues to be included or excluded</td>
</tr>
<tr>
<td>CAODTAB</td>
<td>contains examples of how to specify the CAODTAB macro in DTABJCL depending on what you want to accomplish</td>
</tr>
<tr>
<td>CAOTTAB</td>
<td>contains a macro used to generate the list of terminal types supported by the MainView AutoOPERATOR for CICS Broadcast application</td>
</tr>
<tr>
<td>CICSTART</td>
<td>contains a sample EXEC used to automate the start up of the CICS environment</td>
</tr>
<tr>
<td>CICSTERM</td>
<td>contains a sample EXEC used to automate the shut down of the CICS environment</td>
</tr>
<tr>
<td>DB2START</td>
<td>contains a sample EXEC used to automate the start up of the DB2 environment</td>
</tr>
<tr>
<td>DB2TERM</td>
<td>contains a sample EXEC used to automate the shut down of the DB2 environment</td>
</tr>
<tr>
<td>DIVDEF</td>
<td>creates the linear data set used by MainView AutoOPERATOR to store a variety of data, including nonvolatile ALERTs and data recorded by the Automation Reporter. The subsystem must be down prior to submitting this job stream.</td>
</tr>
<tr>
<td>DIVUTIL</td>
<td>copies the BBIDIV data set to a newly allocated data set or produces a report of space utilization in the BBIDIV</td>
</tr>
<tr>
<td><strong>BBSAMP member name</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>IMSSTART</td>
<td>contains a sample EXEC used to automate the start up of the IMS environment</td>
</tr>
<tr>
<td>IMSTERM</td>
<td>contains a sample EXEC used to automate the shut down of the IMS environment</td>
</tr>
<tr>
<td>DTABJCL</td>
<td>contains sample JCL to assemble and link CAODTAB</td>
</tr>
<tr>
<td>JES2MAO</td>
<td>contains sample JCL to assemble with the user version of SYS1.HASPSRC</td>
</tr>
<tr>
<td>NAICMDS</td>
<td>defines NAIEXEC and NAISTUB to NetView</td>
</tr>
<tr>
<td>NAIINIT</td>
<td>initializes an Access NV controlled OST</td>
</tr>
<tr>
<td>NAIOP00</td>
<td>defines the BBI-SS PAS that communicates with NetView and OSTs used by that BBI-SS PAS</td>
</tr>
<tr>
<td>NAIOPROF</td>
<td>contains a sample OST initialization profile</td>
</tr>
<tr>
<td>NAISVAR</td>
<td>sets global variables for Access NV tasks</td>
</tr>
<tr>
<td>NAITASK</td>
<td>defines NATASK as a NetView optional subtask (OPT)</td>
</tr>
<tr>
<td>QAOCTMCO</td>
<td>calls the CTMAPI in conversational mode to request the RBA of a job</td>
</tr>
<tr>
<td>QAOCTMNC</td>
<td>calls the CTMAPI in non-conversational mode to request the RBA of a job</td>
</tr>
<tr>
<td>RVARTST</td>
<td>contains an EXEC that shows settings of variables</td>
</tr>
<tr>
<td>TTABJCL</td>
<td>contains sample JCL to assemble and link CAOTTAB</td>
</tr>
</tbody>
</table>

## Sample members for MainView AutoOPERATOR for CICS and MainView for CICS

The following table describes BBSAMP members used by both MainView AutoOPERATOR for CICS and MainView for CICS.
Table 35: BBSAMP data set members for MainView AutoOPERATOR *for CICS* and MainView *for CICS*

<table>
<thead>
<tr>
<th>BBSAMP member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMRASMD</td>
<td>sample JCL to assemble and link-edit any of the following modules:</td>
</tr>
<tr>
<td></td>
<td>■ CMRPRBT: MainView <em>for CICS</em></td>
</tr>
<tr>
<td></td>
<td>■ CMRRAPM: MainView <em>for CICS</em></td>
</tr>
<tr>
<td></td>
<td>■ CMRSECU: MainView <em>for CICS</em> and MainView AutoOPERATOR <em>for CICS</em></td>
</tr>
<tr>
<td></td>
<td>■ CMRSOPT: MainView <em>for CICS</em></td>
</tr>
<tr>
<td>Note:</td>
<td>Descriptions for each of these modules can be found in this appendix.</td>
</tr>
<tr>
<td>CMRCSDAL</td>
<td>JCL used to create the CSD resource groups used by MainView <em>for CICS</em> or MainView AutoOPERATOR</td>
</tr>
<tr>
<td>CMRCSDEL</td>
<td>JCL used to delete resources from the BOOLEPPT and BOOLEPCT resource groups</td>
</tr>
<tr>
<td>CMRCSDLA</td>
<td>JCL used to modify the group list that contains the resource groups installed at CICS initialization MainView AutoOPERATOR only</td>
</tr>
<tr>
<td>CMRCSDLB</td>
<td>JCL used to modify the group list that contains the resource groups installed at CICS initialization MainView AutoOPERATOR and MainView <em>for CICS</em></td>
</tr>
<tr>
<td>CMRCSDLM</td>
<td>JCL used to modify the group list that contains the resource groups installed at CICS initialization MainView <em>for CICS</em> only</td>
</tr>
<tr>
<td>CMRPLT</td>
<td>macro used to generate valid PLT entries for CICS and CICS Transaction Server regions</td>
</tr>
<tr>
<td>CMRSECU</td>
<td>security for CICS option of MainView AutoOPERATOR and MainView <em>for CICS</em></td>
</tr>
<tr>
<td>CMRUSECU</td>
<td>sample for defining function security</td>
</tr>
<tr>
<td>CSDOLTDF</td>
<td>contains the DEFINE statements required for the program and transaction resources used by the MainView <em>for CICS</em> and MainView AutoOPERATOR <em>for CICS</em> (all releases of CICS)</td>
</tr>
<tr>
<td>CSDOLT41</td>
<td>contains the DELETE statements used to remove program and transaction resource definitions from the CSD for the CICS 410 release</td>
</tr>
<tr>
<td>CSDOLT51</td>
<td>contains the DELETE statements used to remove program and transaction resource definitions from the CSD for the CICS 510 release</td>
</tr>
<tr>
<td><strong>BBSAMP member name</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>CSDOLT52</td>
<td>contains the DELETE statements used to remove program and transaction resource definitions from the CSD for the CICS 520 release</td>
</tr>
<tr>
<td>CSDOLT53</td>
<td>contains the DELETE statements used to remove program and transaction resource definitions from the CSD for the CICS 530 release</td>
</tr>
<tr>
<td>CSDOLT61</td>
<td>contains the DELETE statements used to remove program and transaction resource definitions from the CSD for the CICS 610 release</td>
</tr>
<tr>
<td>CSDOLT62</td>
<td>contains the DELETE statements used to remove program and transaction resource definitions from the CSD for the CICS 620 release</td>
</tr>
<tr>
<td>CSDAAO41</td>
<td>contains the DELETE statements used to remove program and transaction resource definitions from the CSD for the CICS 410 release</td>
</tr>
<tr>
<td>CSDAAO52</td>
<td>contains the DELETE statements used to remove program and transaction resource definitions from the CSD for the CICS 520 release</td>
</tr>
<tr>
<td>CSDAAO53</td>
<td>contains the DELETE statements used to remove program and transaction resource definitions from the CSD for the CICS 530 release</td>
</tr>
<tr>
<td>CSDAAO61</td>
<td>contains the DELETE statements used to remove program and transaction resource definitions from the CSD for the CICS 610 release</td>
</tr>
<tr>
<td>CSDAAO62</td>
<td>contains the DELETE statements used to remove program and transaction resource definitions from the CSD for the CICS 620 release</td>
</tr>
<tr>
<td>CSDCMR41</td>
<td>contains the DELETE statements used to remove program and transaction resource definitions from the CSD for the CICS 410 release</td>
</tr>
<tr>
<td>CSDCMR51</td>
<td>contains the DELETE statements used to remove program and transaction resource definitions from the CSD for the CICS 510 release</td>
</tr>
<tr>
<td>CSDCMR52</td>
<td>contains the DELETE statements used to remove program and transaction resource definitions from the CSD for the CICS 520 release</td>
</tr>
<tr>
<td>CSDCMR53</td>
<td>contains the DELETE statements used to remove program and transaction resource definitions from the CSD for the CICS 530 release</td>
</tr>
</tbody>
</table>
### Sample members for MainView AutoOPERATOR for IMS, MainView for IMS, and MainView for DBCTL

The following table describes BBSAMP members used only for MainView AutoOPERATOR for IMS, MainView for IMS, and MainView for DBCTL.

<table>
<thead>
<tr>
<th>BBSAMP Member Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICOPY7</td>
<td>sample installation JCL that is used to copy BMC Software modules to IMS.RESLIB data set available to the IMS 4.1 control region</td>
</tr>
<tr>
<td>ICOPY8</td>
<td>sample installation JCL that is used to copy BMC Software modules to IMS.RESLIB data set available to the IMS 5.1 control region</td>
</tr>
<tr>
<td>ICOPY9</td>
<td>sample installation JCL that is used to copy BMC Software modules to IMS.RESLIB data set available to the IMS 6.1 control region</td>
</tr>
<tr>
<td>IMRUVTF</td>
<td>interface from MainView for IMS / AO to the BMC Software Delta IMS virtual terminal If this module is present, MainView for IMS / AO can access all Delta IMS lines, nodes, and LTERMs (CLB, CTB, and CNT).</td>
</tr>
<tr>
<td>LINKNUC</td>
<td>sample JCL that is used to link BMC Software AOI exit IBAOUE3 into the IMS nucleus (IMS version 4.1 and earlier)</td>
</tr>
</tbody>
</table>
Expanding the PROFILE pool data set

This appendix discusses the XTLOAD utility that is located in BBSAMP member XTJCL. You use this utility to expand the PROFILE pool data set.

The PROFILE variables are stored in the PROFILE pool data set. When this data set becomes full, you must allocate and initialize a new, larger data set.

You can use the XTLOAD utility to copy all the PROFILE variables from the old data set into the new data set. You can identify a full PROFILE pool data set if requests to the data set from IMFEXEC VPUT or IMFEXEC VPUTL fail with return codes of 20 or 8 respectively.

**Note**

Do not use IEBGENER to copy the smaller data set into the new, larger data set. You must use the XTLOAD utility for MainView AutoOPERATOR to recognize the additional space.

Using the XTLOAD utility

To expand the PROFILE pool data set, use the XTLOAD utility. You must bring down MainView AutoOPERATOR before you can use this utility.

XTLOAD copies the content of the existing BBIVARS data set into a larger data set. After successful execution of XTLOAD, the new data set can be specified on the BBIVARS DD statement. Alternatively, the new data set can be renamed to match the original data set name on the BBIVARS DD statement.

Use the following sample JCL to execute XTLOAD.

**Figure 46: Sample JCL to execute XTLOAD**

```
//STEP1 EXEC PGM=XTLOAD
//STEP2 LIB DD
DSN=hilevel.?????..BLINK,DISP=SHR
//BBIVARS DD DSN=old.bbivars.dataset,DISP=OLD
```
To modify the sample JCL

1. Provide a valid job card.

2. On the STEPLIB DD statement, specify the BBLINK load library containing XTLOAD.

3. On the BBIVARS DD statement, specify the name of the existing PROFILE pool data set.

4. On the NEWVARS DD statement, specify the name of the new PROFILE pool data set. The NEWVARS data set must be larger than the BBIVARS data set; otherwise, XTLOAD will terminate and write an appropriate error message to the SYSPRINT DD.

The XTLOAD return codes are

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>NEWVARS, BBIVARS, or SYSPRINT DD statement is missing</td>
</tr>
<tr>
<td>8</td>
<td>NEWVARS DD data set is not larger than the BBIVARS DD data set</td>
</tr>
<tr>
<td></td>
<td>If the BBIVARS data set is full, the IMFEXEC VPUTL command receives a return code of 8</td>
</tr>
<tr>
<td>8</td>
<td>BBIVARS data set is full</td>
</tr>
</tbody>
</table>
OSPI terminal definitions for CICS and IMS

The following sections provide information about defining OSPI virtual terminals to CICS and IMS and includes the following sections:

IMS terminal definitions

IMS requires all terminals to be defined in the IMS SYSGEN. If an attempt is made to log on with an OSPI ACB that has not been defined to IMS, the session is not accepted. The following example shows the IMS terminal definitions required for OSPI ACBs:

```
TERMINAL MODEL=2,
FEAT=(.NOCD,.NOPEN),
FPBUF=256,OPTIONS=(TRANRESP),
NAME=OSPI0000         ** OSPI ACB name **
NAME
OSPI0000         ** Any LTERM name **
** (subject to IMS security) **
```

CICS terminal definitions

The terminal definitions required by CICS depend on which features of CICS are used.

If the CICS autoinstall feature is used, local terminal definitions are not required. However, the appropriate terminal models must be defined prior to logon. See "Autoinstall terminals" for information about defining OSPI terminals to CICS.

If the CICS autoinstall feature is not used, each terminal must be defined prior to logon. See “Explicit terminal definitions” on page 344 for information about defining OSPI terminals to CICS.
Autoinstall terminals

If the CICS autoinstall feature is used, terminals need not be defined but the appropriate models must be defined prior to logon. If a model terminal definition that matches the logmode data associated with the OSPI ACB is not available, CICS issues an error message to the transient data queue.

Explicit terminal definitions

If the CICS autoinstall feature is not used, each terminal must be defined before you logon using RDO. If an attempt is made to logon with an OSPI ACB that has not been defined to CICS, the session is not accepted and error messages are written to the transient data queue.

Figure 47 on page 344 provides an example of how to use Resource Definition Online to define an OSPI virtual terminal.

RDO macro

The following figure is an example of using a Resource Definition Online macro.

Figure 47: Example of using Resource Definition Online
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINTadapter</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>PROgsymbols</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Validation</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Formfeed</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>HOrizform</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>VErticalform</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>TEXTKybd</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>TEXTPrint</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Query</td>
<td>No</td>
<td>Yes</td>
<td>Cold</td>
</tr>
<tr>
<td>Outline</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>SOsi</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>BAcktrans</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Cgcsgld</td>
<td>00000, 00000</td>
<td>0-65535</td>
<td></td>
</tr>
<tr>
<td>SESSION PROPERTIES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AScii</td>
<td>No</td>
<td>No</td>
<td>7</td>
</tr>
<tr>
<td>SENTsize</td>
<td>00000</td>
<td>0-30720</td>
<td></td>
</tr>
<tr>
<td>RECEivesize</td>
<td>00000</td>
<td>0-30720</td>
<td></td>
</tr>
<tr>
<td>BRacket</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>LOGMode</td>
<td>D6327B02</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 48: Example of using Resource Definition Online**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIAGNOSTIC DISPLAY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ERRLastline</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>ERRIntensify</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>ERRColor</td>
<td>NO</td>
<td>Blue</td>
<td>Red</td>
</tr>
<tr>
<td>ERRHilight</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>OPERATIONAL PROPERTIES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AUTOConnect</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>ATi</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>TTi</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>CReatesess</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>RELreq</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Discreq</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Nepclass</td>
<td>000</td>
<td>0-255</td>
<td></td>
</tr>
<tr>
<td>Signoff</td>
<td>Yes</td>
<td>No</td>
<td>Logoff</td>
</tr>
<tr>
<td>MESSAGE RECEIVING PROPERTIES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R0utedmsgs</td>
<td>All</td>
<td>All</td>
<td>None</td>
</tr>
<tr>
<td>LOGOnmsg</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>APPLICATION FEATURES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Buildchain</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>USerarealen</td>
<td>02000, 02000</td>
<td>0-32767</td>
<td></td>
</tr>
<tr>
<td>U tras</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>RECOVERY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVOption</td>
<td>Sysdefault</td>
<td>Sysdefault</td>
<td>Clearconv</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Uncondrel</td>
<td>None</td>
</tr>
<tr>
<td>RECOVNotify</td>
<td>None</td>
<td>None</td>
<td>Message</td>
</tr>
</tbody>
</table>
Implementing Print LOGJRNL records

To print or write journal message information that is collected in the MainView Logger when you specify MVLOG=JRNAL in BBPARM member BBSSPxx, use the BBSAMP member PLOGJRNL to set up a started task. You can schedule the started task on a regular basis such as printing all the journal messages for a specific time period to a flat file that can be retained and searched as needed.

Printing LOGJRNL records

Use the following procedure to set up a started task procedure for MainView Logger.

To create the started task procedure

1. Copy BBSAMP member PLOGJRNL to SYS1.PROCLIB to a JES PROCLIB data set and make the following changes to the JCL in the PROC statement:
   - change PLOGJRNL to a name you want to use for the procedure
   - change the PREFIX parameter to the high-level qualifier that you used to load the product to DASD.

2. Refer to Table 37 on page 348 which describes the additional parameters you can use with the START PLOGJRNL command.
### Table 37: Valid parameters for the START command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Specified as one of the following options</th>
<th>Description</th>
</tr>
</thead>
</table>
| SSID=     | no       | ■ a substitution value for the PROC statement  
■ directly in the PARM= value of the JCL step card as the first positional parameter; do not include the characters "SSID=" | specifies the SSID of the MV Logger SSID (if you do not want the default) to connect to. If no value is specified, the SSID of the default MainView Logger in the LPAR is used. |
| STRT=     | yes      | ■ a substitution value for the PROC statement  
■ directly in the PARM= value of the JCL step card as the second positional parameter; do not include the characters "STRT=" | specifies the time and date where the search begins. This parameter is required and you must enclose the value in single quotation marks if it is passed to the PROC statement from a START command. You do not need to use single quotation marks if you specify the value with the PARM statement in the JCL. Valid formats are as follows:  
■ hh:mm:ss/yyyyymmdd  
■ YESTERDAY  
When you are not using the value YESTERDAY, the value you specify must be 17 characters long; for example:  
12:39:21/20111201  
16:00:00/20110305  
05:15:36/20111205  
When you specify YESTERDAY for the STRT value, the system computes yesterday's date and creates a STRT value string in the form: "00:00:00/yesterday's date". You can use this when you want to run the job daily and dump all of "yesterday's" data using a DURA value of 1D. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Specified as one of the following options</th>
<th>Description</th>
</tr>
</thead>
</table>
| RECS=     | no       | ■ a substitution value for the PROC statement  
|           |          | ■ directly in the PARM= value of the JCL step card as the third positional parameter (do not include the characters "RECS=") | specifies the number of records to be printed  
|           |          |                                           | Specify a numeric value between 1 to 7-numbers long between 1-9999999; for example:  
|           |          | ■ 100                                   |             |
|           |          | ■ 10                                    |             |
|           |          | ■ 50000                                 |             |
| ORIG=     | no       | ■ a substitution value for the PROC statement  
|           |          | ■ directly in the PARM= value of the JCL step card as the fourth positional parameter (do not include the characters "ORIG=") | specifies an origin value that is used to filter the records  
|           |          |                                           | Specify a 1-8 alphanumeric value. An asterisk (*) is supported as a wild card character; for example:  
<p>|           |          | ■ AAODL71                              |             |
|           |          | ■ AAODL* (any origin whose name begins with the characters AAODL) |             |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Specified as one of the following options</th>
<th>Description</th>
</tr>
</thead>
</table>
| DURA=     | no       | ■ a substitution value for the PROC statement  
■ directly in the PARM= value of the JCL step card as the fifth positional parameter (do not include the characters "DURA=") | specifies a period of time for the records to be printed  
Specify 1 to 3-digit number followed by a alphabetic value that defines the unit of time; valid alphabetic values are as follows:  
■ M: minutes  
■ H: hours  
■ D: days  
For example:  
■ to print out 10 minutes of data beginning at the time specified by the STRT parameter, specify 10M  
■ to print out 1 day’s worth of data beginning at the time specified by the STRT parameter, specify 1D  
■ to output 8 hours of data beginning at the time specified by the STRT parameter, specify 8H  
You must specify either the RECS= or the DURA= parameter.  
When you specify both RECS and DURA parameters, whichever criteria completes first will end the output. For example, if you specify RECS=100 and DURA=1D, either 100 records or 1 day of records will be output, whichever comes first. |
BBPARM member BBISSPxx parameters

MainView AutoOPERATOR version 7.3 includes BBPARM member BBISSPxx parameter DMABTIM that allows you to specify a time limit (in minutes) within which abends will disable the BBI-SS PAS SSI routine.

Deciding what to specify for this parameter is important because when the BBI-SS PAS SSI routine is disabled, all Rules-based automation processing becomes disabled. For example, you might want to specify that 50 abends occurring within 1 minute (DMABTIM=50) is more severe than 50 occurring abends within 10 hours (DMABTIM=6000).

If the BBI-SS PAS SSI routine is disabled because more than 50 abends have occurred with the specified time limit, you must perform a COLD START of MainView AutoOPERATOR to re-establish the SSI routines, and turn Rules-based automation back on.

If you do not specify a time limit with the DMABTIM parameter, the BBI-SS PAS SSI component is disabled when the 51st abend occurs since the last COLD START of the PAS.

In addition, when you specify the DMABTIM parameter, the QAODUMP EXEC (located in the BBPROC) is scheduled to run in MainView AutoOPERATOR and obtain a console dump of the BBI-SS PAS. This console dump can provide critical diagnostic information to BMC customer support. You can specify your own EXEC with the DMABEXEC parameter.

Refer to Table 38 on page 352 for more information about the DMABTIM and DMABEXEC parameters.
**Table 38: BBISSPx parameters**

<table>
<thead>
<tr>
<th>BBISSPx parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| DMABTIM= [0| 1-999999] (minutes) | specify the number of minutes during which 50 abends disables the BBI-SS SSI component  
**WARNING:** Specifying 0 minutes indicates that the BBI-SS PAS SSI routine should not be disabled because of abends occurring and continue processing regardless of how many abends occur within this component. Therefore, specifying 0 could lead to system degradation or a system outage and you should only specify 0 after careful consideration. |
| DMABEXEC= [execName] | specify an EXEC name that MainView AutoOPERATOR schedules when it detects the 51st abend within the period of time specified by the DMABTIM parameter  
You must use the DMABEXEC parameter in conjunction with the DMABTIM parameter. If you specify a value for DMABEXEC and not for DMABTIM, the value is ignored.  
If you omit this parameter and you specify a value for DMABTIM, MainView AutoOPERATOR schedules the EXEC QAODUMP to obtain a console dump. |
Index

$EXECJOB 335
$EXECTST 335
$TESTDAT 335

A

AAOALS00
BBPARM member 281
modifying ALERT parameters, Dynamic Parameter Manager 74, 75
parameters 285
  ALERTNV 285
  AUDIT 285
  PUBLISH 285
  WARNLVL1 285
  WARNLVL2 285

AAOARP00
BBPARM member 281
modifying Automation Reporter parameters, Dynamic Parameter Manager 79, 80
parameters 289
  DISP 289
  DSN 289
  INTVL 289
  OFFINTVL 289
  QUEUE 289

AAOEXP00

description 245, 246, 291
modifying EXEC parameters
  EXEC= 246
  MAXHIGH= 246
  MAXNORM= 246
modifying EXEC parameters, Dynamic Parameter Manager 82, 91
parameters
  APIAUTH 291
  AUDITEXE 291
  AUDITMSG 291
  AUDITRES 291
  CICSECID 291
  ELAPSLIM 89, 291
  EMLOG 291
  EXABCNT 291
  EXEC 291
  MAXHIGH 291
  MAXHIGHQ 291
  MAXHOTQ 291
  MAXNORM 291
  MAXNORMQ 291
  MAXTPUT 291
  OSPINUM 291
  OSPIPRFX 291
  PEREXLIM 291
  PREFIX 291
  RESAUTH 291
  SELLIM 291
  SUBXAUTH 291
  TIMEXLIM 291
  TSOTIME 291
  UCPARMS 291
  UNITNAME 291
  WARNLVL1 291
  WARNLVL2 291

AAOGME00
BBPARM member 281
parameters
  CONAUDIT 296
  DEBUGMSG 296
  EXEC 296
  GMEACT 296
  HOSTNAME 296
  MVE_LINK 296
  TGTBIMCF 296
  TGTHB 296
  TGTIP 296
  TGTMXMSL 296
  TGTTNAME 296
  TGTRTC 296
  TGTRTI 296
  TRACEAPP 296
  TRACEBUF 296
  TRACELK 296
  TRACESEC 296
  TRACEZN 296
  WTO 296

AAOMQL00 281
  BBPARM parameter
    OPEN 301
    QMGR 301
    QUEUE 301
    TYPE 301

AAONCD00
  BBPARM member 281
  BBPARM parameter
    DEBUG 301, 304
parameters 304

AAOPRM00 307
description 307

AAOPRM00, BBPARM member 281
AAORUL00
  BBPARM member 281
list of Rule Set members 325

AAOTRN00
  BBPARM member 281
  BBPARM parameter
    RLTERM 326
    TRAN 326
parameters 326
  TapeSHARE 326
AAOTSMP00
BBPARM member 281
BBPARM parameter
   ACTION 326
   FREE 326
   NOGIVE 326
   NOTAKE 326
   PARTNER 326
   PREF 326
   PREFDEV 326
   RETRYCNT 326
   RETRYINT 326
   TIMEOUT 326
   TRACE 326
modifying TapeSHARE parameters, Dynamic Parameter Manager 109
ACTION, modifying AAOTSP00 parameters 110
Actv field, Dynamic Parameter Manager 73
address space application, modifying JES2 job numbers 248
ALERT QUEUE
   AAOARP00 parameter, modifying 80
ALERTNV
   AAOALS00 parameter, modifying 75
ALERTs
   setting storage thresholds, Dynamic Parameter Manager 75
   thresholds, Dynamic Parameter Manager 74, 75
alternate database, CSM application, selecting 227
AO exit
   execution 189
   return code 189
   routines 187
AO exit routines
   DFSAOE00 192
   DFSAOUE0 192
AOALRTDF 335
AOALRTPE 335
AOAPPLEDF 335
AOAPPLEPE 335
AOCMDDF 335
AOEEXIT, control statements 187
AOEXECDF 335
AOIEXIT, parameter 187
AOLOGACT
   AAOPRM00 parameter, modifying 101
AOLOGEV
   AAOPRM00 parameter, modifying 101
AOOPTION
   AAOPRM00 parameter, modifying 96
AOPARMDF 335
AOREPLAY
   AAOPRM00 parameter, modifying 101
APIAUTH
   AAOEXP00 parameter, modifying 89
ARM See Automatic Restart Manager 131
ARMPOLCY parameter, valid values 133
AUDIT parameter
   AAOALS00 member, valid values 75
AUDITEXE, field 89
AUDITMSG
   AAOEXP00 parameter, modifying 89
AUDITRES
   AAOEXP00 parameter, modifying 92
authorization 255
Automatic Restart Manager (ARM)
default policy 132
   implementing support 132, 133
   parameter values 133
Automation Logger parameters
   Dynamic Parameter Manager, modifying 101
      AOLOGACT 101
      AOLOGEV 101
Automation Reporter parameters, Dynamic Parameter Manager 79, 80
AutoOPERATOR
   BMC products 31
   BMC Software products 49
AutoOPERATOR Access NV
   checklist 197
   creating
      DSIPARM member DSICMD 200
      DSIPARM member DSIDMN 201
      DSIPARM member DSIOPF 202
      DSIPARM member NAIOPT00 199
      operator profiles 203
      security 203
   implementing 197, 207
   modifying
      NAISTUB 199
      NetView JCL 199
      NETVIEW parameter 199

B

BBI-SS PAS to CICS communication

Index 355
available functions 171
modifying CICS tables 161
overview 159
processing program table (PPT) 161
program control table (PCT) 161
program list table (PLT) 161
restarting CICS target regions 170
Rule Processor 171
specifying Resources to CICS 161
startup JCL 170
BBI-SS PAS, starting before JES 248
BBI-SS to CICS communication
available functions 171
BROADCAST application 171
BBIVAR00, BBPARM member 331
BBKEYS, 7.4, 50
BBKEYS, 8.1, 32
BBLINK
modifying IMS control region JCL 186
BBLINK data set
dynamic install and uninstall 192
BBPARM
AAOALS00 281
AAOARPO0 281
AAOEXP00 281
AAOGME00 281
AAOMQL00 281
AAONCD00 281
AAOPRM00 281
AAORUL00 281
AAOTRN00 281
AAOTSP00 281
members 281, 331
BBPARM member 281
BBSAMP, members
JES2MAO 248
NAICMDS 200
NAINIT 202
NAINITX 202
NAIOPT00 199
NAIPRF 202
NAIPROFX 202
NAISVAR 202
SSJCL 248
BMC Impact Manager
implementing TCP/IP interface 209
interface
implementing 209
overview 209
procedure for implementing connection 210
TCP/IP connection 210
BMC products, AutoOPERATOR 31
BMC Software products, AutoOPERATOR 49

C

CA-ACF2 security 160
CAOTTAB
BBSAMP member 335
macro 177, 178
Changed field, Dynamic Parameter Manager 73
changing transaction IDs, required ZAPs 273
CICSECID
AAOEXP00 parameter, modifying 89
CICSTART 335
CICTERM 335
Cmd field, Dynamic Parameter Manager 73
CMRASM 257, 337
CMRCSDES 337
CMRDETL (detail performance file) 273
CMRPLT 337
CMRSECU
action parameters 258
BBSAMP member 337
implementing 257
setting 258
TYPE=ENTRY 257
TYPE=FINAL 257
TYPE=INITIAL 257
See also security
CMRSTAT (statistical file) 273
CMRUSECU, BBSAMP member 257, 337
command processing 187
commands
migrating
version 7.3 to 7.4, 43
version 7.4 to 8.1, 25
common storage, migrating
version 7.3 to 7.4, 48
common storage, migrating
version 7.4 to 8.1, 30
compatibility, migrating
version 7.3 to 7.4
version 7.3 to 7.4
  upward compatibility 49
version 7.4 to 8.1
  version 7.4 to 8.1
  upward compatibility 31
CONAUDIT
  AAOGME00 parameter, modifying 296
considerations for implementing CICS 160
  consoles
    allocating 121
    CSM application 123
    overview 121
Continuous State Manager
  accessing for the first time 229
  alternate database, selecting 227
  calculating repository size 234
  creating a shared repository 233
  End-of-Memory EXEC 228
  EXECs 227
  performance considerations 227
  Rule Set member, selecting 227
Control-O interface, implementing 220
  conventions, documentation 16
CPU time limit, AutoOPERATOR EXECs,
  TIMEXLIM 245
CPU usage, migrating
  version 7.3 to 7.4 48
  version 7.4 to 8.1 30
Created field, Dynamic Parameter Manager 73
CSMALTD8, SSJCL parameter, setting 228
CSMBUILD 227
CSMDOWN 227
CSMEOEM EXEC 228
CSMFINIT 227
CSMUP 227
Current field, Dynamic Parameter Manager 85
Current Parm field, Dynamic Parameter Manager 115
Current SSID field, Dynamic Parameter Manager 115
Current Target field, Dynamic Parameter Manager 115

D

DATAID parameter 160
DB2START 335
DB2TERM 335
DDNAMEs, migrating

version 7.3 to 7.3 41
version 7.4 to 8.1 21
DEBUGMSG
  AAOGME00 parameter, modifying 296
dependent services 171
DFHTST TYPE=RECOVERY 160
DFSAOE00 187
DFSAOE00 exit routine 192
DFSAUE00 exit routine 192
DISP
  AAOARP00 parameter, modifying 80
DIVDEF 335
DIVUTIL 335
documentation changes
  migrating
    version 7.3 to 7.4 47
    version 7.4 to 8.1 29
documentation information 15
DSICMD, DSIPARM member
  NAICMDS 200
  NAIEXEC 200
  NAISTUB 200
DSIDMN, DSIPARM member
  NATASK 201
DSIOPF, DSIPARM member
  NAIPROF 202
  NAIPROFX 202
DTABJCL 335
DUMP data set, BBI-SS PAS 248
dynamic install and uninstall 192
Dynamic Parameter Manager
activating modifications 115

<table>
<thead>
<tr>
<th>Description</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actv</td>
<td>73</td>
</tr>
<tr>
<td>ALERTNV</td>
<td>75</td>
</tr>
<tr>
<td>ALOGACT</td>
<td>101</td>
</tr>
<tr>
<td>ALOGEV</td>
<td>101</td>
</tr>
<tr>
<td>AOPTION</td>
<td>96</td>
</tr>
<tr>
<td>AOREPLAY</td>
<td>101</td>
</tr>
<tr>
<td>APIAUTH</td>
<td>89</td>
</tr>
<tr>
<td>AUDIT</td>
<td>75</td>
</tr>
<tr>
<td>AUDITMSG</td>
<td>89</td>
</tr>
<tr>
<td>AUDITRES</td>
<td>92</td>
</tr>
<tr>
<td>Changed</td>
<td>73</td>
</tr>
<tr>
<td>CICSCEDID</td>
<td>89</td>
</tr>
<tr>
<td>Created</td>
<td>73</td>
</tr>
<tr>
<td>Current</td>
<td>85</td>
</tr>
<tr>
<td>ELAPSLIM</td>
<td>89</td>
</tr>
<tr>
<td>EMLOG</td>
<td>92</td>
</tr>
<tr>
<td>EXABCNT</td>
<td>85</td>
</tr>
<tr>
<td>EXEC</td>
<td>89</td>
</tr>
<tr>
<td>EXHMC</td>
<td>92</td>
</tr>
<tr>
<td>HONORMPF</td>
<td>99</td>
</tr>
<tr>
<td>ID</td>
<td>73</td>
</tr>
<tr>
<td>IMFDATE</td>
<td>99</td>
</tr>
<tr>
<td>IMFJNUM</td>
<td>96</td>
</tr>
<tr>
<td>IMFORGN</td>
<td>102</td>
</tr>
<tr>
<td>IMSAUTH</td>
<td>102</td>
</tr>
<tr>
<td>IMSCMDCN</td>
<td>102</td>
</tr>
<tr>
<td>IMSMSTIM</td>
<td>102</td>
</tr>
<tr>
<td>IMSPLTIM</td>
<td>102</td>
</tr>
<tr>
<td>IMSROUTE</td>
<td>102</td>
</tr>
<tr>
<td>IMSSHTIM</td>
<td>102</td>
</tr>
<tr>
<td>Lib</td>
<td>73</td>
</tr>
<tr>
<td>MAXDLYQ</td>
<td>96</td>
</tr>
<tr>
<td>MAXHIGH</td>
<td>85</td>
</tr>
<tr>
<td>MAXHIGHQ</td>
<td>85</td>
</tr>
<tr>
<td>MAXHOTQ</td>
<td>85</td>
</tr>
<tr>
<td>MAXNORM</td>
<td>85</td>
</tr>
<tr>
<td>MAXNORMQ</td>
<td>85</td>
</tr>
<tr>
<td>MAXSTOR</td>
<td>75</td>
</tr>
<tr>
<td>MAXTPUT</td>
<td>85</td>
</tr>
<tr>
<td>Member</td>
<td>73, 85</td>
</tr>
<tr>
<td>MSGJOBNM</td>
<td>96</td>
</tr>
<tr>
<td>OSPINUM</td>
<td>89</td>
</tr>
<tr>
<td>OSPIPREFIX</td>
<td>89</td>
</tr>
<tr>
<td>PEREXLIM</td>
<td>89</td>
</tr>
<tr>
<td>PREFIX</td>
<td>89</td>
</tr>
<tr>
<td>PUBLISH</td>
<td>75</td>
</tr>
<tr>
<td>RESAUTH</td>
<td>89</td>
</tr>
<tr>
<td>RULEPROT</td>
<td>99</td>
</tr>
<tr>
<td>RULESAVE</td>
<td>99</td>
</tr>
<tr>
<td>RULESCAN</td>
<td>99</td>
</tr>
</tbody>
</table>
Index

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

AAOEXP00 parameter, modifying 92
End-of-Memory (EOM) EXEC, Continuous State Manager 228
EOM (End-of-Memory) EXEC, Continuous State Manager 228
event
defining events 53
Event Activity Statistics panel, modifying 247
event type
See event 53
EXABCNT
AAOEXP00 parameter, modifying 85
EXEC
AAOEXP00 parameter modifying 89, 91, 246
priority EXECs 246
threading EXECs 246
AAOGME00 parameter, modifying 296
Continuous State Manager 227
CSMEOM 228
migrating
version 7.3 to 7.4 18, 36
parameters, Dynamic Parameter Manager 88, 91, 92
priority, Dynamic Parameter Manager 91
REXX, enabling 245
runaway, loop detection 245
thresholds, Dynamic Parameter Manager 84, 85
EXECs, migrating
version 7.3 to 7.4 38
version 7.4 to 8.1 20
EXHMC
AAOEXP00 parameter, modifying 92
exit routines 187
DFSAOE00 192
DFSAQUE0 192
expanding PROFILE pool data set 341
EXTRA, CAODTAB macro, parameter 178
extrapartition transient data queue destinations, selected 177

F
FCD2 transaction 273
FIC2 transaction 273
fields

E
ELAPSLIM, AAOEXP00 parameter, modifying 89
electronic documentation 15
EMLOG

modifying ALERTs thresholds in AAOALSxx 74, 75
modifying Automation Reporter parameters in AAOARPxx 79, 80
modifying EXEC parameters in AAOEXPxx 82, 91
modifying TapeSHARE parameters in AAOEXPxx 109, 110
sample panels
ALERTs Storage Thresholds 75
Automation Reporter parameters 79, 95, 99, 101, 102, 106
AutoOPERATOR Primary Option Menu 70
Confirm Parameter Activation 115
EXEC Parms 88, 92
EXEC Thresholds 84
setting high-priority EXECs 91
TapeSHARE parameters 110
sending
EXEC messages to EMLOG 92
setting
ALERTs storage thresholds 75
Automation Reporter parameters 79
EXEC parameters 88
EXEC thresholds 84
high-priority EXECs 91
TapeSHARE parameters 110

RULESET 99
SELLIM 85
Size 73
SPLXPREF 96
SPLXSTR 96
SPLXVARS 96
SUBXAUTH 89
SUPREPLY 96
TIMEXLIM 89
TSOTIME 89
UCPARMS 89
UCVARS 96
UNITNAME 89
VV.MM 73
WARNLVL1 75, 85
WARNLVL2 75, 85

modifying ALERTs thresholds in AAOALSxx 74, 75
modifying Automation Reporter parameters in AAOARPxx 79, 80
modifying EXEC parameters in AAOEXPxx 82, 91
modifying TapeSHARE parameters in AAOEXPxx 109, 110
sample panels
ALERTs Storage Thresholds 75
Automation Reporter parameters 79, 95, 99, 101, 102, 106
AutoOPERATOR Primary Option Menu 70
Confirm Parameter Activation 115
EXEC Parms 88, 92
EXEC Thresholds 84
setting high-priority EXECs 91
TapeSHARE parameters 110
sending
EXEC messages to EMLOG 92
setting
ALERTs storage thresholds 75
Automation Reporter parameters 79
EXEC parameters 88
EXEC thresholds 84
high-priority EXECs 91
TapeSHARE parameters 110

modifying ALERTs thresholds in AAOALSxx 74, 75
modifying Automation Reporter parameters in AAOARPxx 79, 80
modifying EXEC parameters in AAOEXPxx 82, 91
modifying TapeSHARE parameters in AAOEXPxx 109, 110
sample panels
ALERTs Storage Thresholds 75
Automation Reporter parameters 79, 95, 99, 101, 102, 106
AutoOPERATOR Primary Option Menu 70
Confirm Parameter Activation 115
EXEC Parms 88, 92
EXEC Thresholds 84
setting high-priority EXECs 91
TapeSHARE parameters 110
sending
EXEC messages to EMLOG 92
setting
ALERTs storage thresholds 75
Automation Reporter parameters 79
EXEC parameters 88
EXEC thresholds 84
high-priority EXECs 91
TapeSHARE parameters 110

modifying ALERTs thresholds in AAOALSxx 74, 75
modifying Automation Reporter parameters in AAOARPxx 79, 80
modifying EXEC parameters in AAOEXPxx 82, 91
modifying TapeSHARE parameters in AAOEXPxx 109, 110
sample panels
ALERTs Storage Thresholds 75
Automation Reporter parameters 79, 95, 99, 101, 102, 106
AutoOPERATOR Primary Option Menu 70
Confirm Parameter Activation 115
EXEC Parms 88, 92
EXEC Thresholds 84
setting high-priority EXECs 91
TapeSHARE parameters 110
sending
EXEC messages to EMLOG 92
setting
ALERTs storage thresholds 75
Automation Reporter parameters 79
EXEC parameters 88
EXEC thresholds 84
high-priority EXECs 91
TapeSHARE parameters 110

modifying ALERTs thresholds in AAOALSxx 74, 75
modifying Automation Reporter parameters in AAOARPxx 79, 80
modifying EXEC parameters in AAOEXPxx 82, 91
modifying TapeSHARE parameters in AAOEXPxx 109, 110
sample panels
ALERTs Storage Thresholds 75
Automation Reporter parameters 79, 95, 99, 101, 102, 106
AutoOPERATOR Primary Option Menu 70
Confirm Parameter Activation 115
EXEC Parms 88, 92
EXEC Thresholds 84
setting high-priority EXECs 91
TapeSHARE parameters 110
sending
EXEC messages to EMLOG 92
setting
ALERTs storage thresholds 75
Automation Reporter parameters 79
EXEC parameters 88
EXEC thresholds 84
high-priority EXECs 91
TapeSHARE parameters 110

ELAPSLIM, AAOEXP00 parameter, modifying 89
electronic documentation 15
EMLOG
ALERTs Storage Thresholds panel 75
Automation Reporter Parms panel 79, 80
Dynamic Parameter Manager 73, 115
EXEC PARMS panel 85, 91
TapeSHARE parameters panel 110, 115
FREE
   AAOTSP00 parameter, modifying 110
FST2 QOFF transaction 273
FST2 QON transaction 273

G

GMEACT
   AAOGME00 parameter, modifying 296

H

Help
   online 15
high-priority EXECs, Dynamic Parameter Manager 91
HONORMPF
   AAOPRM00 parameter 307
   AAOPRM00 parameter, modifying 99
HOSTNAME
   AAOGME00 parameter, modifying 296
HST2 transaction (history display) 273

I

IAODEBUG 191
IBAOUE3 340
IBININST member 193
IBUNINST member 195
ICOPY7 340
ICOPY8 340
ICOPY9 340
ICOPYAOI job 192
ID field, Dynamic Parameter Manager 73
IDEL41$2 181
identifying transactions 273
IELOAD 189
IMFDATE
   AAOPRM00 parameter, modifying 99
IMFEXEC CICS commands 172
IMFJNUM
   AAOPRM00 parameter, modifying 96
IMFORGN
   AAOPRM00 parameter, modifying 102
IMFSYS00 187
IMSAUTH
   AAOPRM00 parameter, modifying 102
IMSCMDCN
   AAOPRM00 parameter, modifying 102
IMSID 255
IMSMSTIM
   AAOPRM00 parameter 307
   AAOPRM00 parameter, modifying 102
IMSPRTIM
   AAOPRM00 parameter 307
   AAOPRM00 parameter, modifying 102
IMSSHTIM
   AAOPRM00 parameter, modifying 102
IMSSTART 335
IMSTERM 335
independent IMFEXEC CICS commands 172
initialization exit 187
install, dynamic 192
installation
dynamic 192
INTERVAL
   AAOARP00 parameter, modifying 80
INTRA, CAODTAB macro, parameter 178
intrapartition transient data queue destinations,
selecting 177
IRHIST, AAOARP00 parameter, modifying 80

J

JCL
   dynamic install 193
dynamic uninstall 195
JCL samples 340
JES, start after AutoOPERATOR 248
JES2 job numbers, displaying 248
JES2MAO, BBSAMP data set 248, 335
JES3 support 223
JNL2 transaction (subsystem status) 273

L

LAST SEG LOST 190
Last Update field, Dynamic Parameter Manager 115
Lib field, Dynamic Parameter Manager 73
LINXNUC 340
loop detection, AutoOPERATOR EXECs 245

M

manual start 273
manual stop 273
MAXDLYQ
   AAOPRM00 parameter, modifying 96
MAXHIGH
   AAOEXP00 parameter, modifying 85, 246
MAXHIGHQ
   AAOEXP00 parameter, modifying 85
MAXHOTQ
   AAOEXP00 parameter, modifying 85
MAXNORM
   AAOEXP00 parameter, modifying 85, 246
MAXNORMQ
   AAOEXP00 parameter, modifying 85
MAXSTOR 285
   AAOALS00 parameter, modifying 75
MAXTPUT
   AAOEXP00 parameter, modifying 85
MCTSIZE=xxxx, specifying 247
Member field, Dynamic Parameter Manager 73, 85
message counter table, defining 247
messages
   migrating
      version 7.3 to 7.4 44
      version 7.4 to 8.1 27
migrating

modifying ALERTs thresholds, Dynamic Parameter Manager 74, 75
modifying Automation Reporter parameters, Dynamic Parameter Manager 79, 80
modifying EXEC parameters, Dynamic Parameter Manager 82, 91
modifying TapeSHARE parameters, Dynamic Parameter Manager 109, 115
MPFFILTR
   AAOPRM00 parameter 307
MPFFLTR parameter
      implementing 136
MQEV
   AAOPRM00 parameter 106, 307
MQEVLPRC
   AAOPRM00 parameter 106, 307
MQGINHIB
AAOPRM00 parameter 106, 307
MQSHARE
AAOPRM00 parameter 106, 307
MSGJOBNM
AAOPRM00 parameter, modifying 96
MSGLVL1 keyword
BOTH 184
MTO (IMS master terminal) 184
NONE 184
WTO (OS\xda 390 system console) 184
MTO message capture 187
MVE_LINK
AAOGME00 parameter, modifying 296
MVS consoles See consoles, subsystem, X-MCS 121
MVS subsystem consoles See consoles, subsystem, X-MCS 121

N
NAICMDS 335
NAIINIT 335
NAILOPT00
description 335
DSIPARM member, OPERxx statements 199
NAIPROF 335
NAISTUB, AutoOPERATOR Access NV, executing 203
NAISVAR 335
NAITASK 335
NetView
DSICLD data set
NAIINIT 202
NAIINITX 202
NAISVAR 202
JCL, modifying 199
NETVIEW parameter, AutoOPERATOR Access NV, BBISSP00 207
NEW DSN
AAOARP00 parameter, modifying 80
NOGIVE
AAOTSP00 parameter, modifying 110
NOTAKE
AAOTSP00 parameter, modifying 110

O
OFF INT
AAOARP00 parameter, modifying 80

online Help 15
Open System Procedural Interface (OSPI)
ACB pool 153–155
accessing VTAM applications 153
defining virtual terminals 153
IMFEXEC LOGON 153
naming conventions 154
OSPINUM 155
OSPIPRFX 154
terminal definitions 343
terminal macros 344
OPID= 263
origin 255
OSPI
See Open System Procedural Interface (OSPI) 153
OSPINUM
AAOEXP00 parameter, modifying 89
OSPIPRFX
AAOEXP00 parameter, modifying 89
other changes
migrating
version 7.3 to 7.4 46
version 7.4 to 8.1 27

P
parameters
BBPARM 281
migrating
version 7.3 to 7.4 18, 36, 41
version 7.4 to 8.1 21
PARTNER
AAOTSP00 parameter, modifying 110
PEREXLIM
AAOEXP00 parameter, modifying 89, 245
performance considerations, CSM 227
PM0229W warning message, BB-SS PAS startup 248
PREF
AAOTSP00 parameter, modifying 110
PREFDEV
AAOTSP00 parameter, modifying 110
PREFIX
AAOEXP00 parameter, modifying 89
processing program table (PPT)
modifying 161
use of 273
product
installation 192
PROFILE pool, expanding data set 341
program control table (PCT), use of 161, 273
program list table (PLT) 161
PTFs
migrating
version 7.3 to 7.4 30, 47
publications, related 15
PUBLISH
AAOALS00 parameter, modifying 75

Q
QALIAS
AAOPRM00 parameter 106, 307
QAOCTMCO 335
QAOCTMNC 335

R
RACF security 160
recoverable temporary storage 271
related publications 15
Replaced by field, Dynamic Parameter Manager
115
RESAUTH
AAOEXP00 parameter, modifying 89
reset, CMRSECU table 257
RESLIB 340
RETRYCNT
AAOTSP00 parameter, modifying 110
REXX, support for EXECS 245
Rule Processor application
event types 53
Rule Set, CSM application, selecting 227
RULEPROT
AAOPRM00 parameter 307
AAOPRM00 parameter, modifying 99
rules processor authorization 265
Rules, migrating
version 7.3 to 7.4 18, 36, 37
version 7.4 to 8.1 18
RULESAVE
AAOPRM00 parameter 307
AAOPRM00 parameter, modifying 99
RULESCAN
AAOPRM00 parameter 307
AAOPRM00 parameter, modifying 99
RULESET
AAOPRM00 parameter 307
AAOPRM00 parameter, modifying 99
RULESET parameter 99, 307
RVARTST 335

S
sample JCL 340
sample members
$EXECJOB 335
$EXECTST 335
$TESTDAT 335
AOALRTDF 335
AOALRTPE 335
AOAPPLDF 335
AOAPPLEP 335
AOCMDDF 335
AOEXECDF 335
AOPARMDF 335
CAOTTAB 335
CICSTART 335
CICSTERM 335
CMRSECU 337
CMRUSECU 337
DB2START 335
DB2TERM 335
DIVDEF 335
DIVUTIL 335
DTABJCL 335
IMSSSTART 335
IMSTERM 335
JES2MAO 335
NAICMDS 335
NAIINIT 335
NAILOPT00 335
NAIPROF 335
NAISVAR 335
NAITASK 335
QAOCTMCO 335
QAOCTMNC 335
RVARTST 335
TTABJCL 335

Index 363
CMRASAM member 257
CMRSECU macro 257
CMRSECU module 257
CMRUSECU member 257
considerations 160
display parameters 258
implementing 257
migrating
  version 7.3 to 7.4 48
  version 7.4 to 8.1 30
setting 258
security for BBI, BBPARM member 255
SELLIM
  AAOEXP00 parameter, modifying 85
setting ALERTs storage thresholds, Dynamic Parameter Manager 75
setting Automation Reporter thresholds, Dynamic Parameter Manager 79, 80
setting EXEC parameters, Dynamic Parameter Manager 88, 91, 92
setting EXEC thresholds, Dynamic Parameter Manager 84, 85
setting high-priority EXECs, Dynamic Parameter Manager 91
setting TapeSHARE thresholds, Dynamic Parameter Manager 109, 115
Size field, Dynamic Parameter Manager 73
SMN2 transaction, manual start 273
SPLXPREF
  AAOPRM00 parameter, modifying 96
SPLXSTR
  AAOPRM00 parameter, modifying 96
SPLXVARS
  AAOPRM00 parameter, modifying 96
SSJCL, CSMALTDB parameter, setting 228
storage thresholds, ALERTs, Dynamic Parameter Manager 75
SUB=MSTR, SSJCL command 248
subsystem consoles
  allocating 121
  CSM application 123
subsystem status 273
SUBXAUTH
  AAOEXP00 parameter, modifying 89
SUPREPLY
  AAOPRM00 parameter 307
  AAOPRM00 parameter, modifying 96
SUPREPLY parameter 307
SYS1.HASPSRC, JES2 offsets 248
sysplex variables
  BBPARM member AAOPRMxx 144
  coupling facility 145, 146, 151
  enabling support 143
  specifying default prefix 144
SYSPROG
  security exit 253
T
TapeSHARE parameters
  Dynamic Parameter Manager 109
  implementing 53–57, 61, 125, 127
    activating 128
    password keys 127
    prerequisite 127
    securing 128
TapeSHARE, AAOTSP00, BBPARM member 326
TARGET 255
TCP/IP connection
  establishing for BMC Impact Manager 210
  prerequisites for BMC Impact Manager 210
temporary storage prefix
  CMRI prefix ZAP 271
  existing definition 271
Temporary Storage Table (TST)
  compatibility 160
  use of 271
terminal emulation, VTAM sessions, OSPI application 153
TGTBIMCF
  AAOGME00 parameter, modifying 296
TGTIB
  AAOGME00 parameter, modifying 296
TGTIP
  AAOGME00 parameter, modifying 296
TGTMXSL
  AAOGME00 parameter, modifying 296
TGNAME
  AAOGME00 parameter, modifying 296
TGTRTC
  AAOGME00 parameter, modifying 296
TGTRTI
  AAOGME00 parameter, modifying 296
TIMEXLIM
  AAOEXP00 parameter, modifying 89, 245
TRACE
  AAOTSP00 parameter, modifying 110
TRACEAPP
  AAOGME00 parameter, modifying 296
USERID 255

V
variables, migrating
version 7.3 to 7.4 40
version 7.4 to 8.1 21
variables, sysplex
coupling facility 145
coupling facility, hardware requirements 146
coupling facility, rebuilding 147
coupling facility, sizing 148
coupling facility, structure definition 149
enabling support 143
specifying default prefix 144
VTAM sessions, accessing, OSPI application 153
VV.MM field, Dynamic Parameter Manager 73

W
WARNLVL1
AAOALS00 parameter, modifying 75
AAOEXP00 parameter, modifying 85
WARNLVL2
AAOALS00 parameter, modifying 75
AAOEXP00 parameter, modifying 85

X
XTLOAD utility 341

Z
ZAP, CMRI temporary storage prefix 271
ZAPs for changed transaction IDs 273