

MAINVIEW® AutoOPERATOR™

Customization Guide

Version 6.2

March 15, 2002



Copyright © 2002 BMC Software, Inc., as an unpublished work. All rights reserved.

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

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

Restricted Rights Legend

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 UNPUBLISHED—RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES.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

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

Contacting BMC Software

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

USA and Canada

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

Telephone 713 918 8800 or
800 841 2031

Fax 713 918 8000

Outside USA and Canada

Telephone (01) 713 918 8800

Fax (01) 713 918 8000

Customer Support

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

Support Web Site

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

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

Support via Telephone or E-mail

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

Before Contacting BMC Software

Before you contact BMC Software, have the following information available so that a technical support analyst can begin working on your problem immediately:

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



Contents

Part 1. Migration Considerations..... 1

Chapter 1. Migrating from AutoOPERATOR Version 6.1 to Version 6.2 3

What You Need to Know to Migrate from AutoOPERATOR 6.1 to AutoOPERATOR 6.2 ... 3

Messages That Have Changed 3

Changes Made to Existing Parameters, Rules, and EXECs 4

CSM Compatibility Issues 4

General Questions and Answers about the New Release 5

What Has Changed about Rules 5

What Are the Changes Made to Variables 6

What DD Names Have Changed 6

What Parameters Have Been Added 6

What Else Has Changed 7

How to Secure AutoOPERATOR 6.2 7

How Much CPU Does the New Release Use 7

Does AutoOPERATOR 6.2 Use Common Storage Differently 7

Does AutoOPERATOR 6.2 Use Private Storage Differently 7

Does This Release Communicate with Release 6.1 7

Does Release 6.1 Communicate with This Release 8

Which IBM Software Does AutoOPERATOR 6.2 Require 8

What Other BMC Software Products Will Run with AutoOPERATOR 6.2. 8

What Is Not Supported by Release 6.2 8

What Functions Are Not Supported in the Next Release of AutoOPERATOR 9

What Functions Are Not Supported in Future Releases of AutoOPERATOR 9

Installation Considerations for AutoOPERATOR 6.2 10

Getting New BBKEYS 10

Defining AutoOPERATOR Resources to CICS 10

Migrating by Using the OS/390 and z/OS Installer: Standard Installation 10

AutoCustomization Considerations when Replacing Existing Products 11

Migrating by Using System Maintenance Program (SMP) 11

Chapter 2. Migrating from AutoOPERATOR Version 5.1 to Version 6.2 13

What You Need to Know to Migrate from AutoOPERATOR 5.1 to AutoOPERATOR 6.2 .. 13

Messages That Have Changed 14

Changes Made to Existing Parameters, Rules, and EXECs 15

CSM Compatibility Issues 15

General Questions and Answers about the New Release 17

What Has Changed about Rules 17

What Are the Changes Made to Variables 18

What DD Names Have Changed 19

What Parameters Have Been Added 19

What Else Has Changed 20

How to Secure AutoOPERATOR 6.2 20

How Much CPU Does the New Release Use 20

Does AutoOPERATOR 6.2 Use Common Storage Differently 20

Does AutoOPERATOR 6.2 Use Private Storage Differently 20

Does This Release Communicate with Release 5.1 21

Does Release 5.1 Communicate with This Release 21

Which IBM Software Does AutoOPERATOR 6.2 Require 21

What Other BMC Software Products Will Run with AutoOPERATOR 6.2	21
What Is Not Supported by Release 6.2	21
What Functions Are Not Supported in the Next Release of AutoOPERATOR	22
What Functions Are Not Supported in Future Releases of AutoOPERATOR	22
Installation Considerations for AutoOPERATOR 6.2	23
Getting New BBKEYS	23
Defining AutoOPERATOR Resources to CICS	23
Migrating by Using the OS/390 and z/OS Installer: Standard Installation	23
When Migrating Using System Maintenance Program (SMP)	24

Part 2. Customizing AutoOPERATOR Functions 25

Advanced Installation Checklist	29
--	-----------

Chapter 3. Continuous Operation: Using the Dynamic Parameter Manager 31

Tasks You Can Accomplish with the Dynamic Parameter Manager	31
Accessing the Dynamic Parameter Manager	32
Describing the Fields	34
Qualifying the Display	35
Modifying ALERTs Thresholds in AAOALSxx	36
Setting ALERTs Storage Thresholds	37
Describing Fields	38
Modifying Automation Reporter Application Parameters in AAOARPxx	39
Setting Automation Reporter Parameters	42
Describing Fields	43
Modifying EXEC Parameters in AAOEXPxx	44
Setting EXEC Thresholds	45
Scheduling EXECs	46
Examples	46
Describing Fields	47
Setting EXEC Parameters	50
Describing Fields	51
Setting High-Priority EXECs	53
Modifying TapeSHARE for AutoOPERATOR Parameters in AAOTSPxx	54
Customizing TapeSHARE	56
Describing Fields	57
Dynamically Implementing New Parameter Settings with the Activate Command	62
Describing Fields	63

Chapter 4. MVS Console Considerations 65

Determining How Many Consoles to Allocate	65
Extended MCS (X-MCS) Consoles	65
Allocating X-MCS Consoles without MIGIDs	66
Allocating X-MCS Consoles with MIGIDs	66
Defining X-MCS Consoles to CICS	67
CICS/ESA Targets	67

Chapter 5. Implementing TapeSHARE 69

Overview	69
Prerequisites	69
Step 1: Entering the Password Key for TapeSHARE (Required)	69
Step 2: Activating TapeSHARE (Required)	69
Using TapeSHARE in its Default State	70
Step 3: Securing TapeSHARE (Optional)	70

After TapeSHARE Is Installed	70
Chapter 6. Implementing Automatic Restart Manager (ARM)	71
Overview	71
Implementation Considerations	71
Implementing ARM Support: Default ARM Policy	72
Implementing ARM Support: Creating a User-Defined Policy	72
Enabling ARM Support	73
Stopping AutoOPERATOR When ARM Is Enabled	73
Chapter 7. Implementing Support for MPF	75
Overview	75
Implementation Considerations	76
Setting the MPFFILTR Parameter	76
Chapter 8. Activating the REXX/370 Alternate Library	79
SMP/E Messages	79
Chapter 9. Defining OSPI Virtual Terminals	81
Overview	81
Using the OSPI ACB Pool	81
Naming Conventions	82
Defining the ACB Pool	82
Defining the ACB Pool to AutoOPERATOR	82
OSPI Application Definitions	83
Chapter 10. Implementing BBI-SS PAS to CICS Communication	85
Overview	85
Implementation Considerations	85
Site-Specific Security	85
Temporary Storage Table (TST) Compatibility	86
Specifying Resources to CICS	86
Making the AutoOPERATOR Transactions and Programs Available to CICS/ESA and CICS Transaction Server	86
Modifying the Program List Table (PLT)	87
Modifying CICS Startup JCL	88
Restarting CICS Target Regions	88
AutoOPERATOR for CICS Functions Available after Implementing BBI-SS PAS-to-CICS Communication	89
AutoOPERATOR for CICS Functions Available if BBI-SS PAS-to-CICS Communication is not Implemented	90
Chapter 11. AutoOPERATOR for CICS Optional Tailoring	91
Overview	91
Customizing Terminal Types Eligible for Broadcast Messages (CAOTTAB)	91
Limiting CICS Transient Data Queues Eligible for Rule Processing (CAODTAB)	92
Using the CAODTAB Macro	93
Describing CAODTAB Macro Parameters	93
Examples	93
Chapter 12. Implementing MAINVIEW Products in IMS	97
Deleting Modules from a Prior Release	98
Setting Up BBPARM Data Sets	99
Customizing BBPARM Members for an IMS System	101

Activating MAINVIEW AutoOPERATOR for IMS	102
Enabling Product Initialization	102
Enabling Product Authorization	102
Product Keys	102
Modifying the IMS Control Region JCL	103
Allocating a BBPARM Data Set	103
Giving IMS Access to MAINVIEW AutoOPERATOR for IMS	103
Enabling AO Exit Routines	104
Stopping Message Queuing to the IMS Master Terminal	107
Chapter 13. Implementing AutoRULE	109
Overview	109
Prerequisites	109
Completing the AutoRULE AutoCustomization Step	109
Chapter 14. Implementing AutoOPERATOR Access NV	115
Overview	115
Modifying NetView JCL	116
Creating DSIPARM Member NAIOPT00	116
Modifying DSIPARM Member DSICMD	117
Modifying DSIPARM Member DSIDMN	117
Copying BBSAMP Members to the NetView DSICLD Data Set	117
Modifying DSIPARM Member DSIOPF	117
Verifying Operator Profiles	118
Reviewing NetView Security for Access NV Operator IDs	118
Reviewing the Use of NetView Exit DSIEX02A	118
Reviewing the Use of NetView Exit DSIEX14	119
Modifying BBPARM Member BBISSP00	120
Chapter 15. Implementing AutoOPERATOR ALERT Interface to PATROL Enterprise Manager	121
Overview	121
Establishing a TCP/IP Connection	123
What You Need	123
Defining TCP/IP Addresses in AutoOPERATOR	124
Specifying More Than One TCP/IP Address	125
Resetting BBPARM Member AAOGME00	125
Controlling the AutoOPERATOR and PATROL EM Interface	126
Modifying the TCP/IP Connection with BBPARM Member AAOGME00	128
How AutoOPERATOR and PATROL EM Share ALERTs	128
ALERTs Synchronization	128
Creating and Deleting ALERT and ALERT Queues	129
Chapter 16. Implementing AutoOPERATOR Advanced JES3 Support	131
Using JES3 2.2 and Higher	131
Intercepting Messages	131
Intercepting Commands	132
Implementing the JES Filter	132
Issuing JES3 Commands from AutoOPERATOR	133
Using JES3 Releases Prior to 2.2	133
Intercepting Messages	133
Intercepting Commands	133
Issuing JES3 Commands from AutoOPERATOR	134

Chapter 17. Implementing COMMON STORAGE MONITOR	135
Overview	135
Starting and Stopping COMMON STORAGE MONITOR	136
Starting and Stopping COMMON STORAGE MONITOR as a Procedure	136
Starting and Stopping COMMON STORAGE MONITOR with ISPF	138
Chapter 18. Implementing Continuous State Manager	139
Performance Considerations	139
Additional Implementation Considerations	139
Overriding the Default Group when CSM Is Initiated	139
Subsystem (SS) JCL Parameter CSMAITDB	139
End-of-Memory Initiated CSM EXEC (CSMEOM)	140
Accessing CSM for the First Time	140
Creating a Shared Repository Data Set for CSM	143
Calculating Repository Size for a 3390	145
Migrating a CSM Database from a Previous Release	146
Dealing with Object-Definition Mismatches after Converting	150
Sample JCL for Copying the Shared Repository	151
Chapter 19. Specific AutoOPERATOR Parameters and JCL	153
REXX Support for EXECs	153
EXEC Loop Detection	153
Priority and Threading of EXECs	154
Defining the Message Counter Table	154
Displaying MVS JES2 Job Numbers in Address Space Application	155
Starting BBI-SS PAS before JES	155
IMFJNUM Option and Seven-Digit JES Job Numbers (z/OS v1.2 and Later)	156
Chapter 20. Accessing Advanced SYSPROG Services from AutoOPERATOR	159
Executing AEWS in the AutoOPERATOR Address Space	159

Part 3. Implementing AutoOPERATOR Security

Chapter 21. Authorization for Special AutoOPERATOR Functions	163
Command Authorization	163
Chapter 22. Authorizing EXECs Invoked by an Operator or the Rule Processor	165
Chapter 23. Implementing AutoOPERATOR for CICS Function Security (CMRSECU)	167
CMRSECU TYPE=INITIAL Statement (Required)	168
CMRSECU TYPE=ENTRY Statement by Operator ID	172
CMRSECU TYPE=FINAL Statement (Required)	173

Part 4. Appendixes

Appendix A. Issuing Commands from Terminal Sessions	177
Issuing OS/390 Commands	177
Using Consoles to Issue OS/390 Commands	177
Issuing BBI Commands	177
Issuing IMS Commands	178

Appendix B. Changing the Use of CICS Recoverable Temporary Storage	179
Appendix C. CICS Transaction and Program Names	181
Transaction Entries for MAINVIEW for CICS and AutoOPERATOR for CICS	181
Program Entries for MAINVIEW for CICS and AutoOPERATOR for CICS	183
Release Independent Programs	183
Release Dependent Programs	183
Appendix D. Manual Activation of BBI-SS to CICS Communication	187
Appendix E. BBPARM Data Set Members for AutoOPERATOR	189
BBPARM Member AAOALS00 Parameters (AutoOPERATOR Only)	190
BBPARM Member AAOALT00 Parameters (AutoOPERATOR Only)	192
BBPARM Member AAOARP00 Parameters (AutoOPERATOR Only)	193
BBPARM Member AAOEWD00 Parameters (AutoOPERATOR Only)	194
BBPARM Member AAOEXP00 Parameters (AutoOPERATOR Only)	195
BBPARM Member AAOGME00 Parameters (AutoOPERATOR Only)	199
BBPARM Member AAOMQL00 (AutoOPERATOR Only)	202
BBPARM Member AAONCD00 Parameters (AutoOPERATOR Only)	204
BBPARM Member AAOPRM00 Parameters (AutoOPERATOR Only)	205
BBPARM Members AAORUL00 Parameters (AutoOPERATOR Only)	208
BBPARM Member AAOTRN00 Parameters (AutoOPERATOR Only)	209
BBPARM Member AAOTSP00 Parameters (AutoOPERATOR Only)	210
BBPARM Member BBIVAR00 Parameters (AutoOPERATOR Only)	217
Appendix F. Sample Data Set Members	221
Sample Members for AutoOPERATOR Only	221
Sample Members for AutoOPERATOR for CICS and MAINVIEW for CICS	223
Sample Members for AutoOPERATOR for IMS, MAINVIEW for IMS, and MAINVIEW for DBCTL	224
Appendix G. Expanding the PROFILE Pool Data Set	225
Using the XTLOAD Utility	225
Appendix H. AutoRULE Worksheet	227
Section 1: MPF List Conversion Parameters	227
Section 2: AutoMate Conversion Parameters	228
Section 3: Generating Rules	228
Section 4: Creating Rules for Production Job Abends	229
Section 5: Creating Rules for Production Job Abends	229
Describing AutoRULE Flags	230
Appendix I. How Product Libraries Should Be Used	233
Parameter Library	235
BBPARM	235
UBBPARM	235
IMS-Specific Parameter Library (IMF and AutoOPERATOR for IMS Only)	236
Sample Library	236
BBSAMP	236
UBBSAMP	236
Profile Library	237
BBPROF	237
SBBPROF	237
User BBPROF	238

Procedure Library (AutoOPERATOR Only)	238
BBPROC	238
UBBPROC	238
BBI-2 Product Library Use Summary	239
Appendix J. OSPI Terminal Definitions for CICS and IMS	241
IMS Terminal Definitions	241
CICS Terminal Definitions	241
Autoinstall Terminals	241
Explicit Terminal Definitions	241
RDO Macro	242
Appendix K. Establishing a VTAM Connection for AutoOPERATOR to PATROL EM Communication	245
Defining VTAM ACBs	245
Defining 3270 Printers to VTAM	246
Resetting BBPARM Member AAOALT00	246
Controlling the AutoOPERATOR to PATROL EM Interface	247
Modifying the VTAM Connection with BBPARM Member AAOALT00	248
3287 Data Stream Format for AutoOPERATOR to PATROL EM	249
Diagnostics	251

Part 5. Glossary and Index	253
Glossary	255
Index	267

Figures

1.	AutoOPERATOR PRIMARY OPTION MENU	32
2.	Dynamic Parameter Manager Panel—Example 1	33
3.	Dynamic Parameter Manager Panel—Example 2	36
4.	ALERTs Storage Thresholds Panel	37
5.	Dynamic Parameter Manager Panel—Example 3	40
6.	Automation Reporter Parms Panel	42
7.	Dynamic Parameter Manager Member Panel	44
8.	EXEC Thresholds Panel	45
9.	EXEC Parms Panel	50
10.	Update Non-Unique Parameter List	53
11.	Dynamic Parameter Manager Panel—Example 4	54
12.	TapeSHARE Parameters Panel	56
13.	Confirm Parameter Activation Panel	62
14.	AutoRULE Event Management: Panel 1	110
15.	AutoRULE Event Management: Panel 2	110
16.	AutoRULE Event Management: Compare Option	111
17.	AutoRULE Event Management: Compare Option Completed	112
18.	AutoRULE Event Management: Naming Rule Sets	112
19.	AutoRULE Event Management: Creating a Production Job	113
20.	AutoRULE Event Management: Specifying ALERT Queues	113
21.	AutoRULE Event Management: Successful Completion Panel	114
22.	AutoOPERATOR and PATROL EM Connection Using TCP/IP	121
23.	AutoOPERATOR and PATROL EM Connections Through I/O Concepts X-Direct Protocol Converters	122
24.	\$\$\$CSMON JCL in BBPARM	137
25.	Example of Offloaded Data	142
26.	BBSAMP member INITCSM Sample JCL	144
27.	Calculating Repository Size	145
28.	Example of the CSM Global Overview Panel	146
29.	Example of the CSM Conversion Panel	146
30.	Example of the Group Detail Control Panel	147
31.	Pop-Up Window for New Group Name	148
32.	CSM Conversion Status Pop-Up Panel	148
33.	CSM Conversion Log	148
34.	CSM Data Set Conversion Completed	149
35.	JCL Example for Copying Shared Repository	151
36.	Sample JCL to Execute XTLOAD	225
37.	BBI-2 Data Set Customization	234
38.	Example of Using Resource Definition Online (Part 1 of 2)	242
39.	Example of Using Resource Definition Online (Part 2 of 2)	243

Tables

1.	AutoOPERATOR Advanced Installation Checklist	29
2.	Syntax for the CMRPLT Macro.	87
3.	BBSAMP Jobs to Delete Old Modules	98
4.	BBI Control Commands for AutoOPERATOR and PATROL EM Interface for TCP/IP Connections	126
5.	Object Definitions	150
6.	Action Parameters for the CMRSECU TYPE=INITIAL and TYPE=ENTRY Statements.	169
7.	Display Parameters for CMRSECU TYPE=INITIAL and TYPE=ENTRY Statements.	170
8.	Transaction IDs	181
9.	ZAPs for Transaction ID Changes.	182
10.	Release Independent Programs	183
11.	Release Dependent Programs	184
12.	BBPARAM Data Set Members: AutoOPERATOR Only.	189
13.	BBPARAM Member AAOALS00 - AutoOPERATOR Only.	190
14.	BBPARAM Member AAOALT00 - AutoOPERATOR Only.	192
15.	BBPARAM Member AAOARP00 - AutoOPERATOR Only	193
16.	BBPARAM Member AAOEWD00 - AutoOPERATOR Only.	194
17.	BBPARAM Member AAOEXP00 - AutoOPERATOR Only.	195
18.	BBPARAM Member AAOGME00 - AutoOPERATOR Only.	199
19.	BBPARAM Member AAOMQL00 - AutoOPERATOR Only.	202
20.	BBPARAM Member AAONCD00 - AutoOPERATOR Only	204
21.	BBPARAM Member AAOPRM00 - AutoOPERATOR Only	205
22.	BBPARAM Member AAOTRN00 - AutoOPERATOR Only	209
23.	BBPARAM Member AAOTSP00 - AutoOPERATOR Only	210
24.	BBPARAM Member BBIVAR00 - AutoOPERATOR Only	217
25.	BBSAMP Data Set Members for AutoOPERATOR.	221
26.	BBSAMP Data Set Members for MAINVIEW AutoOPERATOR for CICS and MAINVIEW for CICS.	223
27.	BBSAMP Data Set Members for AutoOPERATOR for IMS, MAINVIEW for IMS, and MAINVIEW for DBCTL.	224
28.	BBI-2 Product Library Configuration	239
29.	BBI Control Commands Available to AutoOPERATOR to PATROL EM Interface for VTAM Connections.	247
30.	BBPARAM Member AAOALT00	248
31.	Output Messages Description	249

How To Use This Manual

This manual contains procedures for customizing the MAINVIEW AutoOPERATOR product (also referred to simply as AutoOPERATOR) to your site's needs. The AutoOPERATOR product is integrated with BBI architecture. This is a base architecture that allows authorized users to interrogate any MVS, CICS, IMS, or DB2 subsystem in a VTAM network from a single terminal.

To install and customize AutoOPERATOR, follow the instructions in the:

1. *OS/390 and z/OS Installer Guide* to load the BMC Software product libraries.
2. *MAINVIEW Common Customization Guide* and the *MAINVIEW Administration Guide* to tailor BBI-based functions.
3. *Using MAINVIEW* guide to learn how to use MAINVIEW and become familiar with the MAINVIEW architecture.
4. *MAINVIEW AutoOPERATOR Customization Guide* to customize AutoOPERATOR for your site's requirements.

Note: You **must** customize the BBI-SS PAS prior to customizing AutoOPERATOR.

This manual is intended for the system programmer who needs to know how to modify the basic BBI product installation to include AutoOPERATOR functions.

How This Manual Is Organized

There are four parts to this manual:

- Part 1. Migration Considerations
Contains release updates that must be considered before using the current release of the product.
- Part 2. Customizing AutoOPERATOR Functions
Contains the instructions for tailoring AutoOPERATOR functions to suit your site's needs.
- Part 3. Implementing AutoOPERATOR Security
Contains the instructions to allow user access to AutoOPERATOR product services.
- Part 4. Appendixes
Contains the appendixes, glossary, and index.

MAINVIEW AutoOPERATOR Product Library

MAINVIEW AutoOPERATOR is available with seven options:

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

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

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

This manual also makes several references to the BBI architecture which provides subsystem communication in its own OS/390 address space. The BBI online environment is described in the:

- *MAINVIEW Common Customization Guide*
- *MAINVIEW Administration Guide*
- *Using MAINVIEW*

Recommended Documentation

Before using this customization guide, the product must be installed as described in the *OS/390 and z/OS Installer Guide*. BBI-based functions can then be modified as described in the *MAINVIEW Common Customization Guide* and the *MAINVIEW Administration Guide*. You might also want to review the *Using MAINVIEW* manual to become familiar with the MAINVIEW architecture.

Related Documentation

You might want to refer to the following IBM documentation:

- *Advanced Communications Function for VTAM (ACF/VTAM)*, SC38-0256
- *IBM System Modification Program Extended*, SC28-1107
- *ISPF and ISPF/PDF Version 2 for MVS Installation and Customization Guide*, SC34-4117
- *Advanced Communications Function for VTAM - Installation and Resource Definition*, SC27-0610

Product Library

The BBI-based products include:

- MAINVIEW AutoOPERATOR
- MAINVIEW for CICS
- MAINVIEW for IMS
- MAINVIEW FOCAL POINT
- MAINVIEW for DB2

Instructions for installing the product libraries for these products are in the *OS/390 and z/OS Installer Guide*. Installation and administration instructions for the BBI-based functions are in the *MAINVIEW Common Customization Guide* and *MAINVIEW Administration Guide*.

Product-specific customization instructions are in:

- MAINVIEW AutoOPERATOR Customization Guide
- MAINVIEW for CICS Customization Guide
- MAINVIEW for IMS Customization Guide
- MAINVIEW FOCAL POINT User Guide
- MAINVIEW for DB2 Customization Guide

The following manuals document the use of BBI General Services:

- *MAINVIEW AutoOPERATOR Basic Automation Guide*
- *MAINVIEW for CICS PERFORMANCE REPORTER User Guide*
- *MAINVIEW for DB2 User Guide*
- MAINVIEW for IMS Online - Analyzers Reference Manual
- MAINVIEW for IMS Online - Monitors and Traces Reference Manual
- Using MAINVIEW

What the Conventions Are

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

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

Part 1. Migration Considerations

Part 1 of this guide addresses migration considerations associated with the customization of AutoOPERATOR.

If you are migrating from AutoOPERATOR version 6.1 to MAINVIEW AutoOPERATOR version 6.2, refer to “Migrating from AutoOPERATOR Version 6.1 to Version 6.2” on page 3.

If you are migrating from AutoOPERATOR version 5.1 to AutoOPERATOR 6.2, refer to “Migrating from AutoOPERATOR Version 5.1 to Version 6.2” on page 13.

Part 1. Migration Considerations 1

Chapter 1. Migrating from AutoOPERATOR Version 6.1 to Version 6.2 3

What You Need to Know to Migrate from AutoOPERATOR 6.1 to AutoOPERATOR 6.2 . . .	3
Messages That Have Changed	3
Changes Made to Existing Parameters, Rules, and EXECs	4
CSM Compatibility Issues	4
General Questions and Answers about the New Release	5
What Has Changed about Rules	5
What Are the Changes Made to Variables	6
What DD Names Have Changed	6
What Parameters Have Been Added	6
What Else Has Changed	7
How to Secure AutoOPERATOR 6.2	7
How Much CPU Does the New Release Use	7
Does AutoOPERATOR 6.2 Use Common Storage Differently	7
Does AutoOPERATOR 6.2 Use Private Storage Differently	7
Does This Release Communicate with Release 6.1	7
Does Release 6.1 Communicate with This Release	8
Which IBM Software Does AutoOPERATOR 6.2 Require	8
What Other BMC Software Products Will Run with AutoOPERATOR 6.2	8
What Is Not Supported by Release 6.2	8
What Functions Are Not Supported in the Next Release of AutoOPERATOR	9
What Functions Are Not Supported in Future Releases of AutoOPERATOR	9
Installation Considerations for AutoOPERATOR 6.2	10
Getting New BBKEYS	10
Defining AutoOPERATOR Resources to CICS	10
Migrating by Using the OS/390 and z/OS Installer: Standard Installation	10
AutoCustomization Considerations when Replacing Existing Products	11
Migrating by Using System Maintenance Program (SMP)	11

Chapter 2. Migrating from AutoOPERATOR Version 5.1 to Version 6.2 13

What You Need to Know to Migrate from AutoOPERATOR 5.1 to AutoOPERATOR 6.2 . .	13
Messages That Have Changed	14
Changes Made to Existing Parameters, Rules, and EXECs	15
CSM Compatibility Issues	15
General Questions and Answers about the New Release	17
What Has Changed about Rules	17
What Are the Changes Made to Variables	18
What DD Names Have Changed	19

What Parameters Have Been Added.	19
What Else Has Changed.	20
How to Secure AutoOPERATOR 6.2.	20
How Much CPU Does the New Release Use	20
Does AutoOPERATOR 6.2 Use Common Storage Differently	20
Does AutoOPERATOR 6.2 Use Private Storage Differently	20
Does This Release Communicate with Release 5.1	21
Does Release 5.1 Communicate with This Release	21
Which IBM Software Does AutoOPERATOR 6.2 Require	21
What Other BMC Software Products Will Run with AutoOPERATOR 6.2	21
What Is Not Supported by Release 6.2	21
What Functions Are Not Supported in the Next Release of AutoOPERATOR.	22
What Functions Are Not Supported in Future Releases of AutoOPERATOR.	22
Installation Considerations for AutoOPERATOR 6.2	23
Getting New BBKEYS.	23
Defining AutoOPERATOR Resources to CICS.	23
Migrating by Using the OS/390 and z/OS Installer: Standard Installation	23
When Migrating Using System Maintenance Program (SMP)	24

Chapter 1. Migrating from AutoOPERATOR Version 6.1 to Version 6.2

Note: This chapter has been updated for Beta but might continue to be revised through the Beta test period.

This chapter contains the following discussions:

- What you need to know to migrate quickly and successfully from AutoOPERATOR 6.1 to AutoOPERATOR 6.2
- General questions and answers about the current release
- Installation considerations for AutoOPERATOR 6.2

What You Need to Know to Migrate from AutoOPERATOR 6.1 to AutoOPERATOR 6.2

The following sections include important discussions about:

- What messages have changed
- What changes have been made to existing parameters, Rules, and EXECs
- CSM 6.1 to CSM 6.2 compatibility

Messages That Have Changed

Messages changes since the AutoOPERATOR 6.1 release are as follows:

- The following three TapeSHARE message IDs have changed:
 - DS0501I has been changed to DS4000I.
 - DS1201I has been changed to DS4200I.
 - DS1401I has been changed to DS4300I.

If you have any automation that is triggered from these messages, you must convert these Rules to use the new message IDs.

- The following message has changed:

DX9501I RULE: XXXXXXXX SUSPENDED AFTER YYY MATCHES; RULE SET:
ZZZZZZZZ

This message contains the new fields RULE SET and *ZZZZZZZZ*, which represent the Rule Set name.

- There are 2 new messages:

AU6021E JES JOB NUMBER GREATER THAN 99,9990

This message appears the first time 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 AutoOPERATOR.

AU6020W7 DIGIT JES JOB NUMBER DETECTED

This message is issued the first time AutoOPERATOR detects a job number greater than 99,999. For more information, refer to “IMFJNUM Option and Seven-Digit JES Job Numbers (z/OS v1.2 and Later)” on page 156.

Changes Made to Existing Parameters, Rules, and EXECs

All existing AutoOPERATOR 6.1 parameter values are honored in this release, and you can copy your parameter members from your AutoOPERATOR 6.1 BBI-SS PAS and use them for your AutoOPERATOR 6.2 PAS.

However, you must 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).

In addition, you can run your automation without any manual modifications by copying your existing Rules and EXECs to your new AutoOPERATOR 6.2 PAS.

There are some compatibility limitations between AutoOPERATOR 6.1 Rules and 6.2 Rules. Refer to “What Has Changed about Rules” on page 5.

For additional information about new parameters or new BBPARM members, refer to “What Parameters Have Been Added” on page 6.

CSM Compatibility Issues

There are no compatibility issues when you migrate from AutoOPERATOR 6.1 to 6.2, but there is a new primary command. The PROFile command on the Global Overview panel allows you to select the names of the groups that are to be displayed in the Global Overview. For more information, refer to the *MAINVIEW AutoOPERATOR Basic Automation Guide*.

In addition, objects can be moved from an inactive to an active group within the same repository. This allows users who need to recover started task objects from failed LPARs in the SYSPLEX. Moves from an inactive group can be triggered either from the Global Overview display, with the CSM command line interface or by scheduling CSMACT on the BBI-PAS that the object is to move to. For more information, refer to the *MAINVIEW AutoOPERATOR Basic Automation Guide*.

General Questions and Answers about the New Release

The following sections contain discussions about migrating from AutoOPERATOR 6.1 to AutoOPERATOR 6.2.

What Has Changed about Rules

Using Rules instead of EXECs for certain automation tasks has proven to reduce CPU usage by as much as 90 percent.

Note: The Rules and Rule Sets that you created on an AutoOPERATOR 6.1 PAS can run without changes on AutoOPERATOR 6.2.

With AutoOPERATOR 6.2, the following changes apply to your AutoOPERATOR 6.1 Rules and Rule Sets:

- A terminal session connected to an AutoOPERATOR 6.1 PAS can target an AutoOPERATOR 6.2 PAS (and vice versa) and can enable, disable, browse, edit, and otherwise manipulate AutoOPERATOR 6.2 Rules.
- When viewing a 6.1 Rule from an AutoOPERATOR 6.2 PAS, all help panels will contain information specific to what is available to a 6.2 Rule.
- The 6.2 Rule-creation panels can be used to add new version 6.1 Rules with the following conditions:

The AutoOPERATOR 6.2 Rule-creation panels contain fields supported in AutoOPERATOR version 6.2. Use of some of these fields when creating a version 6.1 Rule will return error messages when the Rule is created. When the other fields are used although no error message is displayed, the Rule might not fire, or might fire incorrectly, such as when using the new REWORD CMD field for CMD Rules.

Other changes made to Rules in AutoOPERATOR 6.2 are as follows:

- A new field REWORD CMD has been added to the Action Specification Panel for CMD events. With this field, you can enter a new command that will replace the original command and the response will return to the originating console. Prior to this release, you were restricted to rejecting the command and reissuing the command CONSOLE 00. Now, the response will return to the originating console.
- The Filter Criteria panel has been modified so that you can specify a match rate for a Rule Set. When Rules within a specified Rule Set exceed the match rate, the Rule can be suspended or disabled, or the Rule will fire but the Rule's actions will not be taken until the rate falls below the threshold. This modification allows you to specify a match rate for all Rules within the Rule Set that do not have individual match rates specified.
- The *Ignore leading plus* field has been added to the selection criteria for MSG types. This feature allows messages issued from authorized and non-authorized programs to be trapped by a single Rule.

Note: If you currently use the distributed Rules ALRMSTRT and ALRMSTOP or you have written Rules that create (or delete) an AutoOPERATOR ALERT when AutoOPERATOR receives an MVALARM, you should review the automation that uses these Rules, especially if you have also specified ALRTRCVE=YES in BBPARM member BBISSP00. If this parameter is set to YES, AutoOPERATOR

will automatically create (or delete) ALERTs and therefore, you might receive duplicate ALERTs.

You might avoid this situation by disabling ALRMSTRT and ALRMSTOP (or your other Rules) if ALRTRCVE is set to YES.

What Are the Changes Made to Variables

There are new variables returned when you use the IMFEXEC WAITLIST command. The variables return the name, date and time of the waiting EXEC.

There are also new ALRT.x. variables returned with the AOEXEC SYSINFO command. These variables might contain either the value YES or NO, indicating whether a particular subsystem has ALRTRCVE=YES specified in its BBISSPxx member.

Another new variable returned by AOEXEC SYSINFO is IMFXCFCGP. When returned, it contains the value of the XCF group of the addressed systems.

For more information about these variables, refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* and *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.

What DD Names Have Changed

No changes were made to existing DD names in AutoOPERATOR 6.2.

What Parameters Have Been Added

The following changes were made to BBPARM members in AutoOPERATOR 6.2:

- BBPARM member BBISSP00 can now have a user-defined suffix, which can be specified during BBI-SS PAS initialization. The default suffix is 00. Specifying different member name suffixes allows multiple BBI-SS PASs to process different configuration members from the same BBPARM library. Refer to the *MAINVIEW Common Customization Guide* for information about specifying suffixes for BBISSP00.
- BBPARM member BBISSP00 has a new parameter, ALRTRCVE=YES | NO (where NO is the default). This parameter specifies whether the AutoOPERATOR subsystem is eligible to receive ALERTs from other BMC Software products using the AOAnywhere AOEXEC ALERT command. Refer to the *MAINVIEW Common Customization Guide* for information about parameters set in BBPARM member BBISSP00.
- BBPARM member BBISSP00 has a new parameter, XCFGROUP= xxxxxxxx where xxxxxxxx is a user-specified one to eight character XCF group name.

Note: The AutoOPERATOR BBI-PASes that are in different XCF Groups are unable to communicate with each other using AOAnywhere. If you want to prevent 'test' AOAnywhere callers from accessing 'production' AutoOPERATORS, SAF security is a better solution than using XCFGROUP.

Also note that the BBISSPxx ALRTRCV=yes setting is incompatible with the XCFGROUP parameter. Do not specify the XCFGROUP parameter in any BBI-PAS that has ALRTRCV=yes.

- BBPARM member BBISSP00 has a new parameter, JESCNCT=YES|NO which allows you to use JES functions even if the BBI-SS PAS is started with SUB=MSTR specified or before JES is started. When JESCNCT=YES is specified, you can use the IMFEXEC JESALLOC and JESSUBM commands to allocate and submit JES jobs.

What Else Has Changed

BBSAMP member QAOSMTP1 has been updated to make modifications simpler. This sample EXEC shows how to send an e-mail through an SMTP mail server and is documented in the *MAINVIEW AutoOPERATOR Solutions Guide*. You should review QAOSMTP1 to determine whether you prefer to implement your current version of the QAOSMTP1 EXEC or upgrade to release 6.2.

When you apply PTFs BPO5552, BPO5580 and BPO5563, this release will support JES job numbers greater than 99,999 for JES2 and JES3 jobs. For more information, refer to “IMFJNUM Option and Seven-Digit JES Job Numbers (z/OS v1.2 and Later)” on page 156.

How to Secure AutoOPERATOR 6.2

AutoOPERATOR 6.2 allows you to implement security for AutoOPERATOR with the external security manager (ESM) installed at your site (just as AutoOPERATOR 6.1 does). While you may continue to use BMC Software proprietary security (documented in *Implementing Security for MAINVIEW Products*), you might also find that you want to control security of AutoOPERATOR through your ESM.

This release includes new or additional security for the CSM application and the AOAnywhere AOEXEC statements. Refer to the *Implementing Security for MAINVIEW Products* for more information.

Some BBPARM members contain EXECs that may aid in the definition and permission of AutoOPERATOR resources. For the names and functions of these EXECs, refer to “Sample Members for AutoOPERATOR Only” on page 221.

How Much CPU Does the New Release Use

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

Does AutoOPERATOR 6.2 Use Common Storage Differently

AutoOPERATOR 6.2 does not use common storage differently than AutoOPERATOR 6.1.

Does AutoOPERATOR 6.2 Use Private Storage Differently

AutoOPERATOR 6.2 does not use private storage differently than AutoOPERATOR 6.1.

Does This Release Communicate with Release 6.1

Any BBI-SS PAS with an AutoOPERATOR 6.2 product activated can schedule an ALERT or EXEC to any AutoOPERATOR 6.1 BBI-SS PAS. In addition, an AutoOPERATOR 6.2 terminal session can access AutoOPERATOR 6.1 panels.

You can install AutoOPERATOR 6.2 on your test system and use it to review production operation until you are satisfied with the product’s stability and functionality. Then you can migrate AutoOPERATOR 6.2 through your other systems, one at a time, at a pace that is acceptable to your staff.

Note: Be sure to review compatibility restrictions in “What Has Changed about Rules” on page 5.

Does Release 6.1 Communicate with This Release

ALERTs, EXECs, and Rules can be sent from an AutoOPERATOR 6.1 BBI-SS PAS to an AutoOPERATOR 6.2 BBI-SS PAS.

Which IBM Software Does AutoOPERATOR 6.2 Require

Refer to Chapter 1, “Installation Prerequisites”, in the *MAINVIEW Installation Requirements Guide* for information about IBM software (and levels) that are required to run AutoOPERATOR 6.2.

What Other BMC Software Products Will Run with AutoOPERATOR 6.2

These release levels and high of the following BMC Software products are supported by AutoOPERATOR 6.2:

- SYSPROG Services release 3.2
- MAINVIEW for CICS release 5.3.00, 5.3.01, 5.4.00, 5.4.0.1, and 5.5.00
- IMF release 3.1
- MAINVIEW for DBCTL release 2.1
- MAINVIEW for MVS release 2.2

What Is Not Supported by Release 6.2

AutoOPERATOR 6.2 does not support any IBM software release that is not supported by IBM. This list includes

- IMS/VS versions 1.3 and 2.2
- IMS/ESA version 3.1 and 4.1
- Any CICS release earlier than CICS/ESA 4.1.0
- CICS Transaction Server 1.1.0
- MVS/XA
- MVS/ESA versions 3.x , 4.x, and 5.x
- NetView versions 1.2, 1.3 and 2.1
- TSO/E versions 1.x, 2.1.0, 2.1.1, 2.2.0, 2.3.0, 2.3.1, 2.4, and 2.5
- TCP/IP 3.1

Within the AutoOPERATOR product, support for the IMFEXEC CMD (schedule an EXEC) has been dropped in this release of AutoOPERATOR. This function has been replaced by the IMFEXEC SELECT(execname) command. For more information about IMFEXEC SELECT(execname), refer to the chapter “Using the IMFEXEC Statements” in either the

*MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs or the
MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs.*

What Functions Are Not Supported in the Next Release of AutoOPERATOR

The CSMALTDDB parameter in the SS JCL will not be supported in the next release of AutoOPERATOR. Refer to “Additional Implementation Considerations” on page 139 for information about the CSMALTDDB parameter.

What Functions Are Not Supported in Future Releases of AutoOPERATOR

BMC Software plans to discontinue the use of documented variables by the Continuous State Manager application. When these documented variables are discontinued, BMC Software will attempt to identify all customer implementations that are dependent on these variables to provide an alternate method of functionality.

Installation Considerations for AutoOPERATOR 6.2

This section contains technique-specific installation considerations for migrating from previous releases of AutoOPERATOR to AutoOPERATOR version 6.2.

Getting New BBKEYS

If a new release of AutoOPERATOR is installed on the same CPU where a currently licensed copy resides, the same set of BBKEYS is required as the current copy. ***There is no need to request a new set of BBKEYS.*** You are required to obtain a new set of BBKEYS only when changing or upgrading CPUs.

Refer to the *MAINVIEW Common Customization Guide* for more information about acquiring BBKEYS.

Defining AutoOPERATOR Resources to CICS

If you have customized your CICS regions for use with a previously released version of AutoOPERATOR, the CICS resource definitions for your BMC Software products must be refreshed.

This update is accomplished by customizing and executing the BMC Software job CMRCSDES in the BBSAMP data set.

No other changes are needed except changing the CICS execution JCL to reflect the names of the new BBPARM data sets.

Migrating by Using the OS/390 and z/OS Installer: Standard Installation

The *OS/390 and z/OS Installer Guide* refers to two installation methods:

- Standard installation provides a fast IEBCOPY installation process. This method offers an optional step that creates a predefined SMP/E environment for SMP/E maintenance. The Standard installation method requires less expertise than an SMP/E installation, but it offers less control over the SMP/E environment.
- SMP/E is an industry-standard installation program that provides a variety of capabilities for unloading products and maintenance. SMP/E provides more flexibility than the Standard installation method for setting up and using an SMP/E environment, but it is more complicated and time consuming.

If you are migrating by using standard installation with the BMC Software OS/390 and z/OS Installer, read the following section on implementing the JES2MAO exit again.

If you are replacing any AutoOPERATOR release and ***all*** of the following statements are true, you must implement the JES2MAO exit again:

- Your site is licensed for the AutoOPERATOR for OS/390 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” in “Specific AutoOPERATOR Parameters and JCL” on page 153, for additional information about the JES2MAO exit.

AutoCustomization Considerations when Replacing Existing Products

When replacing the existing BMC Software products, the following conditions apply:

- All existing products that are replaced must be customized again.
- Any modifications or additions to the UBBPARM data set that were performed manually after running AutoCustomization on the existing system need to be performed as necessary on the new UBBPARM data set after it is created by AutoCustomization on the new system.

Refer to “Customize the Installed Products” in the *OS/390 and z/OS Installer Guide* for additional information about AutoCustomization.

Migrating by Using System Maintenance Program (SMP)

Read this section if you are using System Maintenance Program (SMP) to migrate.

AutoOPERATOR 6.2 *must not* be installed into a set of SMP zones that contains any release of the BMC Software products listed in “What Other BMC Software Products Will Run with AutoOPERATOR 6.2” on page 9 without also installing the minimum release level of those products.

Chapter 2. Migrating from AutoOPERATOR Version 5.1 to Version 6.2

This chapter contains the following discussions:

- What you need to know to migrate quickly and successfully from AutoOPERATOR 5.1 to AutoOPERATOR 6.2
- General questions and answers about the current release
- Installation considerations for AutoOPERATOR 6.2

You should read this chapter in conjunction with the MAINVIEW AutoOPERATOR version 6.2 Release Notes.

What You Need to Know to Migrate from AutoOPERATOR 5.1 to AutoOPERATOR 6.2

The following describes tasks you must complete to migrate from AutoOPERATOR 5.1 to AutoOPERATOR 6.2

- If you plan to use any of the following new or modified applications in AutoOPERATOR 6.2, you must perform the manual customization specified in “Step 25: Create a BBI-SS PAS Start Procedure”, which is described in the *MAINVIEW Common Customization Guide*.

These three applications use the Generalized Message Exchange (GME) which now requires the BMC Software Common TCP/IP Interface (GTS):

- AutoOPERATOR COMMAND/POST (ACE), which strengthens the integration between AutoOPERATOR and PATROL Enterprise Manager (PATROL EM) by removing the need for a printer device and introducing simple, two-way communication between the two products.
- *Command MQ On Ramp (COR)*, which simplifies the deployment of *Command MQ* through a Web-based application that automatically configures distributed MQSeries queue managers to match the topology needed by AutoOPERATOR for MQSeries and MAINVIEW for MQSeries.
- Automation Power Line (APL), which allows you to inquire queue manager names, start and stop queue managers, and display queue manager attributes from an AutoOPERATOR EXEC without using MQSeries channels.

For more details about activating GTS and GME, refer to “Implementing AutoOPERATOR ALERT Interface to PATROL Enterprise Manager” on page 121.

- In this release, MAINVIEW AutoOPERATOR for MQSeries has been enhanced where you can now
 - Open MQSeries queues for automation exclusively
 - Perform catch-up processing where AutoOPERATOR has the ability to process existing messages on a queue when AutoOPERATOR connects to the queue manager

To use these new features, you must set the OPEN parameter in BBPARM member AAOMQL00, which is documented in “BBPARM Member AAOMQL00 (AutoOPERATOR Only)” on page 202 in this manual and also in the *MAINVIEW AutoOPERATOR for MQSeries Installation and User Guide*.

In addition, due to compatibility issues between MAINVIEW AutoOPERATOR for MQSeries 6.2 and the products PATROL for MQ - Operator version 2.2.00 and PATROL for MQ Administrator version 3.1.00, you may have to make changes to ensure compatibility if you are using MAINVIEW AutoOPERATOR for MQSeries with these other products. Refer to the *MAINVIEW AutoOPERATOR for MQSeries Installation Guide and User Guide* for more information.

- If you have linked or copied any AutoOPERATOR modules into non-AutoOPERATOR libraries, you must perform customization for those areas again for this release.

For example, you may need to update CICS table definitions for the AutoOPERATOR for CICS option. Refer to “Defining AutoOPERATOR Resources to CICS” on page 23 for more information.

Or you may need to relink the AOIEXIT, which is required for AutoOPERATOR for IMS (except for IMS 5.1 or later). Refer to “Implementing MAINVIEW Products in IMS” on page 97, for information about implementing products into IMS.

The following sections include important discussions about:

- What messages have changed
- What changes have been made to existing parameters, Rules, and EXECs
- CSM 5.1 to CSM 6.2 compatibility

Messages That Have Changed

Messages changes since the AutoOPERATOR 5.1 release are as follows:

- The wording of the following message has changed:

IM9462I START GME COMMAND COMPLETE FOR: XXXX

to

IM9462I START GME COMMAND ISSUED FOR: XXXX

This new message indicates that a START command was issued but not necessarily completed.

If you have any automation that is triggered from the wording within message IM9462I, you might need to revise it to use the new message wording.

- The wording of the following message has changed:

GM6522E TCP/IP STEPLIB MISSING/INVALID

to

GM6522E GTS ENVIRONMENT NOT ENABLED

If you have any automation that is triggered from the wording within message GM6522E, you might need to revise it to use the new message wording.

- The following three TapeSHARE message IDs have changed:

– DS0501I has been changed to DS4000I.

- DS1201I has been changed to DS4200I.
- DS1401I has been changed to DS4300I.

If you have automation that is triggered from these messages, you must convert these Rules to use the new message IDs.

- The following message has changed:

DX9501I RULE: XXXXXXXXX SUSPENDED AFTER YYY MATCHES; RULE SET:
ZZZZZZZZ

This message contains the new text RULE SET and *ZZZZZZZZ*, which represent the Rule Set name.

- There are 2 new messages:

AU6021E JES JOB NUMBER GREATER THAN 99,9990

This message appears the first time 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 AutoOPERATOR.

AU6020W7 DIGIT JES JOB NUMBER DETECTED

This message is issued the first time AutoOPERATOR detects a job number greater than 99,999. For more information, refer to “IMFJNUM Option and Seven-Digit JES Job Numbers (z/OS v1.2 and Later)” on page 156.

Changes Made to Existing Parameters, Rules, and EXECs

All existing AutoOPERATOR 5.1 parameter values are honored in this release, and you can copy your parameter members from your AutoOPERATOR 5.1 BBI-SS PAS and use them for your AutoOPERATOR 6.2 PAS.

However, you must 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).

In addition, you can run your automation without any manual modifications by copying your existing Rules and EXECs to your new AutoOPERATOR 6.2 PAS.

There are some compatibility limitations between AutoOPERATOR 5.1 Rules and 6.2 Rules. Refer to “What Has Changed about Rules” on page 17.

For additional information about new parameters or new BBPARM members, refer to “What Parameters Have Been Added” on page 19.

CSM Compatibility Issues

If you are currently using the AutoOPERATOR 5.1 version of the Continuous State Manager (CSM) application, you must convert your 5.1 database to operate with the 6.2 version of CSM. Complete conversion instructions are documented in “Implementing Continuous State Manager” on page 139.

In addition, objects can be moved from an inactive to an active group within the same repository. This allows users who need to recover started task objects from failed LPARs in the SYSPLEX. Moves from an inactive group can be triggered either from the Global

Overview display, with the CSM command line interface or by scheduling CSMACT on the BBI-PAS that the object is to move to. For more information, refer to the *MAINVIEW AutoOPERATOR Basic Automation Guide*.

The panels and navigation through them in the CSM application has changed a great deal since the AutoOPERATOR 5.1 release. It is recommended that you review the documentation for CSM in the *MAINVIEW AutoOPERATOR Basic Automation Guide*.

General Questions and Answers about the New Release

The following sections contain discussions about migrating from AutoOPERATOR 5.1 to AutoOPERATOR 6.2.

What Has Changed about Rules

Using Rules instead of EXECs for certain automation tasks has proven to reduce CPU usage by as much as 90 percent.

With AutoOPERATOR 6.2, the following changes apply to your AutoOPERATOR 5.1 Rules and Rule Sets:

- The restriction for Alarm events has been removed.

The Rules and Rule Sets you created on an AutoOPERATOR 5.1 PAS can run without changes on AutoOPERATOR 6.2 except, in previous releases, to process Alarm events with a Rule you were required to have product option keys to receive Alarms from different sources. For example, to receive an Alarm from MAINVIEW for IMS, the AutoOPERATOR for IMS product option key was required. In AutoOPERATOR 6.2, this restriction for Alarm events has been removed.

- A terminal session connected to an AutoOPERATOR 5.1 PAS can target an AutoOPERATOR 6.2 PAS (and vice versa) and can enable, disable, browse, edit, and otherwise manipulate AutoOPERATOR 6.2 Rules with the exception of Rules created for event type MQS (MQSeries) as follows:
 - You can use the 6.2 Rule creation panels to update 5.1 MQSeries Rules with attributes that are available in AutoOPERATOR 5.1.
 - You cannot use the 6.2 Rule creation panels to add variable dependencies that use Hex operators to a 5.1 Rule. For more information about Hex operators, refer to Chapter 6, “Creating Rules for Events: Using the Rule Creation Panels” in the *MAINVIEW AutoOPERATOR Basic Automation Guide*.
 - When viewing a Rule with the 5.1 Rules panels, you cannot view or update any 6.2 Rule that uses Hex operators.
 - You cannot use the 5.1 Rules panels to view or update any 6.2 Rule that was created for the event type MQS.
 - When viewing a 5.1 Rule with the 6.2 Rules, all help panels will contain information specific to what is available to a 5.1 Rule.

The 6.2 Rule-creation panels can be used to add new version 5.1 Rules with the following conditions:

The AutoOPERATOR 6.2 Rule-creation panels contain fields supported in AutoOPERATOR version 6.2. Use of some of these fields when creating a version 5.1 Rule will return error messages when the Rule is created. Although no error message is displayed when the other fields are used, the Rule might not fire, or might fire incorrectly, such as when using attributes in the SUB, LEN, or OP fields next to the MSGID, CORRELI D, MSG BUFFER and QUEUE ID fields. In this case, the data will become part of the selection criteria for that field. When attempting to write AutoOPERATOR 5.1 Rules from an AutoOPERATOR 6.2

Rule creation panel, the previously mentioned fields must contain data entered in the Value column only.

For an MQEVENT, if you specify anything other than the SYSTEM.ADMIN.*.EVENT queue names or blanks, no error message are issued but the Rule will never fire.

If the Rule did not fire and you suspect a compatibility error, you can look at the BBI Journal (use option L) and find the time frame when the Rule was supposed to fire to see if there are any syntax error messages listed.

Other changes made to Rules in AutoOPERATOR 6.2 are as follows:

- A new field REWORD CMD has been added to the Action Specification Panel for CMD events. With this field, you can enter a new command that will replace the original command.
- The Filter Criteria panel has been modified so that you can specify a match rate for a Rule Set. When Rules within a specified Rule Set exceed the match rate, the Rule can be suspended or disabled, or the Rule might fire but the Rule's actions will not be taken until the rate falls below the threshold. This modification allows you to specify a match rate for all Rules within the Rule Set that do not have individual match rates specified.
- The *Ignore leading plus* field has been added to the selection criteria for MSG types. This additional feature allows messages issued from authorized and non-authorized programs to be trapped by a single Rule.

Note: If you currently use the distributed Rules ALRMSTRT and ALRMSTOP or you have written Rules that create (or delete) an AutoOPERATOR ALERT when AutoOPERATOR receives an MVALARM, you should review the automation that uses these Rules, especially if you have also specified ALRTRCVE=YES in BBPARAM member BBISSP00. If this parameter is set to YES, AutoOPERATOR will automatically create (or delete) ALERTs and therefore, you might receive duplicate ALERTs.

You might be able to avoid this situation by disabling ALRMSTRT and ALRMSTOP (or your other Rules) if ALRTRCVE is set to YES.

What Are the Changes Made to Variables

There are two new variables in AutoOPERATOR 6.2: IMFDDNAM and IMFETYPE. For more information about these variables, you can refer to the variable documentation in the following books:

- *MAINVIEW AutoOPERATOR Basic Automation Guide*
- *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs*
- *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*

There are new variables returned when you use the IMFEXEC WAITLIST command. The variables return the name, date, and time of the waiting EXEC.

There are also new ALRT.x. variables returned with the AOEXEC SYSINFO command. These variables might contain either the value YES or NO, indicating whether a particular subsystem has ALRTRCVE=YES specified in its BBISSPxx member.

Another new variable returned by AOEXEC SYSINFO is IMFXCFCGP. When returned, it contains the value of the XCF group of the addressed systems.

For more information about these variables, refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* and *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.

In addition, major additions were made in AutoOPERATOR 6.2 to variables for automating MQSeries events. Refer to the *MAINVIEW AutoOPERATOR for MQSeries Installation and User Guide* for more information.

What DD Names Have Changed

No changes were made to existing DD names in AutoOPERATOR 6.2.

What Parameters Have Been Added

The following changes were made to BBPARM members in AutoOPERATOR 6.2:

- AutoOPERATOR 6.2 contains a new BBPARM member BBIVAR00. This member allows you to preset SHARED variables when the BBI-SS PAS is initialized. Refer to “BBPARM Member BBIVAR00 Parameters (AutoOPERATOR Only)” on page 217.
- There is a new BBPARM member BBTTCP00 that contains the parameters that control the BMC Software Generic TCP/IP interface and a new parameter, GTS=xx in BBPARM member BBISSP00 (where xx is the suffix for BBPARM member BBTTCPxx).

For more information about BBPARM member BBTTCP00 and the GTS parameter in BBISSP00, refer to Appendix C in the *MAINVIEW Common Customization Guide*.

- BBPARM member AAOPRM00 has a new parameter, MQEVLPRC=, to support coexistence with the MQSeries Event Listener.
- BBPARM member AAOMQL00 also contains new parameters for the AutoOPERATOR for MQSeries product option.

Refer to the *MAINVIEW AutoOPERATOR for MQSeries Installation and User Guide* for more information about the new parameters for BBPARM members AAOPRM00 and AAOMQL00.

- BBPARM member AAOALS00 has a new parameter, PUBLISH=, that was added to support whether AutoOPERATOR will (by default) publish ALERTs to PATROL EM consoles that are attached through GME. Possible values are as follows:
 - ADD: publish, but do not replace a previously published ALERT
 - REPLACE: publish, but delete a previously published ALERT before adding the new ONE.
 - NO: do not publish ALERTs

The default is ADD.

- BBPARM member BBISSP00 can now have a user-defined suffix, which can be specified during BBI-SS PAS initialization. The default suffix is 00. Specifying different member name suffixes allows multiple BBI-SS PASs to process different configuration members from the same BBPARM library. Refer to the *MAINVIEW Common Customization Guide* for information about specifying suffixes for BBISSP00.

- BBPARM member BBISSP00 has a new parameter, ALRTRCVE=YES | NO (where NO is the default). This parameter specifies whether the AutoOPERATOR subsystem is eligible to receive ALERTs from other BMC Software products using the AOAnywhere AOEXEC ALERT command. Refer to the *MAINVIEW Common Customization Guide* for information about parameters set in BBPARM member BBISSP00.
- BBPARM member BBISSP00 has a new parameter, XCFGROUP= xxxxxxxx where xxxxxxxx is a user-specified one to eight character XCF group name.
- BBPARM member BBISSP00 has a new parameter, JESCNCT=YES|NO which allows you to use JES functions even if the BBI-SS PAS is started with SUB=MSTR specified or before JES is started. When JESCNCT=YES is specified, you can use the IMFEXEC JESALLOC and JESSUBM commands to allocate and submit JES jobs.

What Else Has Changed

This release adds support of 7-digit job numbers for JES2 and JES3 jobs. With this support, you can have numbers up to 999,999 with room to expand one digit further. For more information, refer to “IMFJNUM Option and Seven-Digit JES Job Numbers (z/OS v1.2 and Later)” on page 156.

How to Secure AutoOPERATOR 6.2

With AutoOPERATOR 6.2, you can implement security for AutoOPERATOR with the external security manager (ESM) installed at your site (just as AutoOPERATOR 5.1 does). While you may continue to use BMC Software proprietary security (documented in *Implementing Security for MAINVIEW Products*), you might also find that you want to control security of AutoOPERATOR through your ESM.

This release includes new or additional security for the CSM application and the AOAnywhere AOEXEC statements. Refer to the *Implementing Security for MAINVIEW Products* for more information.

Some BBPARM members contain EXECs that may aid in the definition and permission of AutoOPERATOR resources. For the names and functions of these EXECs, refer to “Sample Members for AutoOPERATOR Only” on page 221.

How Much CPU Does the New Release Use

Automation for MQSeries events has been optimized to reduce CPU usage.

Does AutoOPERATOR 6.2 Use Common Storage Differently

In AutoOPERATOR 6.2, automation areas for MQSeries have been moved to private storage from Extended Common Storage Area (ECSA).

Does AutoOPERATOR 6.2 Use Private Storage Differently

In AutoOPERATOR 6.2, automation areas for MQSeries have been moved to private storage from Extended Common Storage Area (ECSA).

Does This Release Communicate with Release 5.1

Any BBI-SS PAS with an AutoOPERATOR 6.2 product activated can schedule an ALERT or EXEC to any AutoOPERATOR 5.1 BBI-SS PAS. In addition, an AutoOPERATOR 6.2 terminal session can access AutoOPERATOR version 5.1 panels.

You can install AutoOPERATOR 6.2 on your test system and use it to review production operation until you are satisfied with the product's stability and functionality. Then you can migrate AutoOPERATOR 6.2 through your other systems, one at a time, at a pace that is acceptable to your staff.

Note: Be sure to review compatibility restrictions in “What Has Changed about Rules” on page 17.

Does Release 5.1 Communicate with This Release

ALERTs, EXECs, and Rules can be sent from an AutoOPERATOR 5.1 BBI-SS PAS to an AutoOPERATOR 6.2 BBI-SS PAS.

Which IBM Software Does AutoOPERATOR 6.2 Require

Refer to Chapter 1, “Installation Prerequisites”, in the *MAINVIEW Installation Requirements Guide* for information about IBM software (and levels) that are required to run AutoOPERATOR 6.2.

What Other BMC Software Products Will Run with AutoOPERATOR 6.2

These release levels and high of the following BMC Software products are supported by AutoOPERATOR 6.2:

- SYSPROG Services release 3.2
- MAINVIEW for CICS release 5.3.00, 5.3.01, 5.4.00, 5.4.01, and 5.5.00
- IMF Release 3.1
- MAINVIEW for DBCTL release 2.1
- MAINVIEW for MVS release 2.2

What Is Not Supported by Release 6.2

AutoOPERATOR 6.2 does not support any IBM software release that is not supported by IBM. This includes

- IMS/VS versions 1.3, 2.2
- IMS/ESA version 3.1 and 4.1
- Any CICS release earlier than CICS/ESA 4.1.0
- CICS Transaction Server 1.1.0
- MVS/XA
- MVS/ESA versions 3.x , 4.x and 5.x

- NetView versions 1.2, 1.3, 2.1
- TSO/E versions 1.x, 2.1.0, 2.1.1, 2.2.0, 2.3.0, 2.3.1, 2.4, and 2.5
- TCP/IP 3.1

Within the AutoOPERATOR product, support for the IMFEXEC CMD (schedule an EXEC) has been dropped in this release of AutoOPERATOR. This function has been replaced by the IMFEXEC SELECT(execname) command. For more information about IMFEXEC SELECT(execname), refer to the chapter “Using the IMFEXEC Statements” in either the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.

What Functions Are Not Supported in the Next Release of AutoOPERATOR

The CSMALTDDB parameter in the SS JCL will not be supported in the next release of AutoOPERATOR. Refer to “Additional Implementation Considerations” on page 139 for information about the CSMALTDDB parameter.

What Functions Are Not Supported in Future Releases of AutoOPERATOR

BMC Software plans to discontinue the use of documented variables by the Continuous State Manager application. When these documented variables are discontinued, BMC Software will attempt to identify all customer implementations that are dependent on these variables to provide an alternate method of functionality.

Installation Considerations for AutoOPERATOR 6.2

This section contains technique-specific installation considerations for migrating from previous releases of AutoOPERATOR to AutoOPERATOR Version 6.2.

Getting New BBKEYS

If a new release of AutoOPERATOR is installed on the same CPU as a presently licensed copy, the same set of BBKEYS is required as the current copy. ***There is no need to request a new set of BBKEYS.*** You are required to obtain a new set of BBKEYS only when changing or upgrading CPUs.

Refer to the *MAINVIEW Common Customization Guide* for more information about acquiring BBKEYS.

Defining AutoOPERATOR Resources to CICS

If you have customized your CICS regions for use with a previously released version of AutoOPERATOR, the CICS resource definitions for your BMC Software products must be refreshed.

This update is accomplished by customizing and executing the BMC Software job CMRCSDES in the BBSAMP data set.

No other changes are needed except changing the CICS execution JCL to reflect the names of the names of the BBPARM data sets.

Migrating by Using the OS/390 and z/OS Installer: Standard Installation

The *OS/390 and z/OS Installer Guide* refers to two installation methods:

- Standard installation provides a fast IEBCOPY installation process. This method offers an optional step that creates a predefined SMP/E environment for SMP/E maintenance. The standard installation method requires less expertise than an SMP/E installation, but it offers less control over the SMP/E environment.
- SMP/E is an industry-standard installation program that provides a variety of capabilities for unloading products and maintenance. SMP/E provides more flexibility than the standard installation method for setting up and using an SMP/E environment, but it is more complicated and time consuming.

Read this section if you are migrating using standard installation with the BMC Software OS/390 and z/OS Installer.

- Reimplementing the JES2MAO exit

If you are replacing any AutoOPERATOR release and ***all*** of the following statements are true:

- Your site is licensed for the AutoOPERATOR for OS/390 option.
- You elected to implement the JES2MAO exit.
- The JES2MAO exit is linked to the existing system's BBLINK library.

you must reimplement the JES2MAO exit.

Refer to “Displaying MVS JES2 Job Numbers in Address Space Application” in “Specific AutoOPERATOR Parameters and JCL” on page 153, for additional information on the JES2MAO exit.

- AutoCustomization considerations when replacing existing products

When replacing the existing BMC Software products:

- All existing products that are replaced must be recustomized.
- Any modifications or additions to the UBBPARM data set that were performed manually after running AutoCustomization on the existing system are performed as necessary on the new UBBPARM data set after it is created by AutoCustomization on the new system.

Refer to “Customize the Installed Products” in the *OS/390 and z/OS Installer Guide* for additional information about AutoCustomization.

When Migrating Using System Maintenance Program (SMP)

Read this section if you are using System Maintenance Program (SMP) to migrate.

The current release of AutoOPERATOR Version 6.2 **must not** be installed into a set of SMP zones that contains any release of the BMC Software products listed in “What Other BMC Software Products Will Run with AutoOPERATOR 6.2” on page 21 without also installing the minimum release level of those products.

Part 2. Customizing AutoOPERATOR Functions

Part 2 details both mandatory and optional customization steps.

Advanced Installation Checklist	29
Chapter 3. Continuous Operation: Using the Dynamic Parameter Manager	31
Tasks You Can Accomplish with the Dynamic Parameter Manager	31
Accessing the Dynamic Parameter Manager	32
Describing the Fields	34
Qualifying the Display	35
Modifying ALERTs Thresholds in AAOALSxx	36
Setting ALERTs Storage Thresholds	37
Describing Fields	38
Modifying Automation Reporter Application Parameters in AAOARPx	39
Setting Automation Reporter Parameters	42
Describing Fields	43
Modifying EXEC Parameters in AAOEXPxx	44
Setting EXEC Thresholds	45
Scheduling EXECs	46
Examples	46
Describing Fields	47
Setting EXEC Parameters	50
Describing Fields	51
Setting High-Priority EXECs	53
Modifying TapeSHARE for AutoOPERATOR Parameters in AAOTSPxx	54
Customizing TapeSHARE	56
Describing Fields	57
Dynamically Implementing New Parameter Settings with the Activate Command	62
Describing Fields	63
Chapter 4. MVS Console Considerations	65
Determining How Many Consoles to Allocate	65
Extended MCS (X-MCS) Consoles	65
Allocating X-MCS Consoles without MIGIDs	66
Allocating X-MCS Consoles with MIGIDs	66
Defining X-MCS Consoles to CICS	67
CICS/ESA Targets	67
Chapter 5. Implementing TapeSHARE	69
Overview	69
Prerequisites	69
Step 1: Entering the Password Key for TapeSHARE (Required)	69
Step 2: Activating TapeSHARE (Required)	69
Using TapeSHARE in its Default State	70
Step 3: Securing TapeSHARE (Optional)	70
After TapeSHARE Is Installed	70
Chapter 6. Implementing Automatic Restart Manager (ARM)	71
Overview	71
Implementation Considerations	71
Implementing ARM Support: Default ARM Policy	72
Implementing ARM Support: Creating a User-Defined Policy	72

Enabling ARM Support	73
Stopping AutoOPERATOR When ARM Is Enabled	73
Chapter 7. Implementing Support for MPF	75
Overview	75
Implementation Considerations	76
Setting the MPFFILTR Parameter	76
Chapter 8. Activating the REXX/370 Alternate Library	79
SMP/E Messages	79
Chapter 9. Defining OSPI Virtual Terminals	81
Overview	81
Using the OSPI ACB Pool	81
Naming Conventions	82
Defining the ACB Pool	82
Defining the ACB Pool to AutoOPERATOR	82
OSPI Application Definitions	83
Chapter 10. Implementing BBI-SS PAS to CICS Communication	85
Overview	85
Implementation Considerations	85
Site-Specific Security	85
Temporary Storage Table (TST) Compatibility	86
Specifying Resources to CICS	86
Making the AutoOPERATOR Transactions and Programs Available to CICS/ESA and CICS Transaction Server	86
Modifying the Program List Table (PLT)	87
Modifying CICS Startup JCL	88
Restarting CICS Target Regions	88
AutoOPERATOR for CICS Functions Available after Implementing BBI-SS PAS-to-CICS Communication	89
AutoOPERATOR for CICS Functions Available if BBI-SS PAS-to-CICS Communication is not Implemented	90
Chapter 11. AutoOPERATOR for CICS Optional Tailoring	91
Overview	91
Customizing Terminal Types Eligible for Broadcast Messages (CAOTTAB)	91
Limiting CICS Transient Data Queues Eligible for Rule Processing (CAODTAB)	92
Using the CAODTAB Macro	93
Describing CAODTAB Macro Parameters	93
Examples	93
Chapter 12. Implementing MAINVIEW Products in IMS	97
Deleting Modules from a Prior Release	98
Setting Up BBPARM Data Sets	99
Customizing BBPARM Members for an IMS System	101
Activating MAINVIEW AutoOPERATOR for IMS	102
Enabling Product Initialization	102
Enabling Product Authorization	102
Product Keys	102
Modifying the IMS Control Region JCL	103
Allocating a BBPARM Data Set	103
Giving IMS Access to MAINVIEW AutoOPERATOR for IMS	103

Enabling AO Exit Routines	104
Stopping Message Queuing to the IMS Master Terminal	107
Chapter 13. Implementing AutoRULE	109
Overview	109
Prerequisites	109
Completing the AutoRULE AutoCustomization Step	109
Chapter 14. Implementing AutoOPERATOR Access NV	115
Overview	115
Modifying NetView JCL	116
Creating DSIPARM Member NAIOPT00	116
Modifying DSIPARM Member DSICMD	117
Modifying DSIPARM Member DSIDMN	117
Copying BBSAMP Members to the NetView DSICLD Data Set	117
Modifying DSIPARM Member DSIOPF	117
Verifying Operator Profiles	118
Reviewing NetView Security for Access NV Operator IDs	118
Reviewing the Use of NetView Exit DSIEX02A	118
Reviewing the Use of NetView Exit DSIEX14	119
Modifying BBPARM Member BBISSP00	120
Chapter 15. Implementing AutoOPERATOR ALERT Interface to PATROL Enterprise Manager	121
Overview	121
Establishing a TCP/IP Connection	123
What You Need	123
Defining TCP/IP Addresses in AutoOPERATOR	124
Specifying More Than One TCP/IP Address	125
Resetting BBPARM Member AAOGME00	125
Controlling the AutoOPERATOR and PATROL EM Interface	126
Modifying the TCP/IP Connection with BBPARM Member AAOGME00	128
How AutoOPERATOR and PATROL EM Share ALERTs	128
ALERTs Synchronization	128
Creating and Deleting ALERT and ALERT Queues	129
Chapter 16. Implementing AutoOPERATOR Advanced JES3 Support	131
Using JES3 2.2 and Higher	131
Intercepting Messages	131
Intercepting Commands	132
Implementing the JES Filter	132
Issuing JES3 Commands from AutoOPERATOR	133
Using JES3 Releases Prior to 2.2	133
Intercepting Messages	133
Intercepting Commands	133
Issuing JES3 Commands from AutoOPERATOR	134
Chapter 17. Implementing COMMON STORAGE MONITOR	135
Overview	135
Starting and Stopping COMMON STORAGE MONITOR	136
Starting and Stopping COMMON STORAGE MONITOR as a Procedure	136
Starting and Stopping COMMON STORAGE MONITOR with ISPF	138
Chapter 18. Implementing Continuous State Manager	139

Performance Considerations	139
Additional Implementation Considerations.....	139
Overriding the Default Group when CSM Is Initiated.....	139
Subsystem (SS) JCL Parameter CSMALTDDB	139
End-of-Memory Initiated CSM EXEC (CSMEOM)	140
Accessing CSM for the First Time	140
Creating a Shared Repository Data Set for CSM	143
Calculating Repository Size for a 3390.....	145
Migrating a CSM Database from a Previous Release	146
Dealing with Object-Definition Mismatches after Converting.....	150
Sample JCL for Copying the Shared Repository	151
Chapter 19. Specific AutoOPERATOR Parameters and JCL	153
REXX Support for EXECs	153
EXEC Loop Detection.....	153
Priority and Threading of EXECs	154
Defining the Message Counter Table.....	154
Displaying MVS JES2 Job Numbers in Address Space Application	155
Starting BBI-SS PAS before JES.....	155
IMFJNUM Option and Seven-Digit JES Job Numbers (z/OS v1.2 and Later).....	156
Chapter 20. Accessing Advanced SYSPROG Services from AutoOPERATOR.....	159
Executing AEWS in the AutoOPERATOR Address Space.....	159

Advanced Installation Checklist

Use these checklists to decide which customization steps you need to do now and which ones you will do later.

Table 1. AutoOPERATOR Advanced Installation Checklist

Complete any of the following steps for the options of AutoOPERATOR that you will use at your site.

- Dynamically modify parameters in BBPARM members, see page 31
- Implement TapeSHARE for AutoOPERATOR, see page 69
- Implement support for IBM's Automatic Restart Manager (ARM) facility, see page 71
- Implement support for IBM's Message Processing Facility (MPF), see page 75
- Activate the REXX/370 Alternate Library, see page 79
- Define OSPI virtual terminals, see page 81
- Implement BBI-SS PAS-to-CICS communication, see page 85
- Implementing MAINVIEW Products in IM, see page 97
- Implement AutoRULE, see page 109
- Implement MAINVIEW AutoOPERATOR Access NV, see page 115
- Implement AutoOPERATOR advanced JES3 support, see page 131
- Implement Common Storage Monitor, see page 135

The following steps are optional and allow you to further customize your system.

- Change user authorization and security; see *Implementing Security for MAINVIEW Products*
- Modify or add terminal sessions; see *MAINVIEW Administration Guide* and *Using MAINVIEW*
- Set up additional subsystems; see *MAINVIEW Common Customization Guide*
- Define cross-system parameters; see *MAINVIEW Common Customization Guide*
- Modify General Services; see *Using MAINVIEW*
- Optionally, tailor AutoOPERATOR for CICS, see page 91
- Specify AutoOPERATOR parameters and JCL, see page 153
- Display JES2 job numbers in the address space application, see page 155 (MAINVIEW AutoOPERATOR for OS/390 option only)
- Access SYSPROG Services from AutoOPERATOR, see page 159

Chapter 3. Continuous Operation: Using the Dynamic Parameter Manager

The Dynamic Parameter Manager (DPM) is a panel-driven application that allows users to view or modify parameters in BBPARM members from a TS session. The BBPARM members currently supported by DPM are:

AAOALSxx	AutoOPERATOR ALERT thresholds
AAOARPxx	AutoOPERATOR Automation Reporter parameters
AAOEXPxx	AutoOPERATOR EXEC parameters
AAOTSPxx	TapeSHARE for AutoOPERATOR parameters

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**.

Tasks You Can Accomplish with the Dynamic Parameter Manager

Using the Dynamic Parameter Manager, you can:

- Control virtual storage consumption

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

- Control CPU & I/O usage

By using DPM to dynamically adjust the CPU & I/O throttles in AAOEXPxx.

- Dynamically implement previously defined AAOALSxx, AAOARPxx, AAOEXPxx, and AAOTSPxx members

By dynamically implementing previously defined members, 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 data base.

The Automation Reporter application (and how to activate it with the Dynamic Parameter Manager) is described in the *MAINVIEW AutoOPERATOR Basic Automation Guide*.

- Customize the way the TapeSHARE for AutoOPERATOR component performs tape device sharing automation among your OS/390 images

The TapeSHARE for AutoOPERATOR component, also called TapeSHARE, allows you to automate the sharing of tape devices between OS/390 images. For more information about installing TapeSHARE, refer to “Implementing TapeSHARE” on page 69; 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 Figure 1) to see a list of the AAOALSxx, AAOARPxx, AAOEXPxx, and AAOTSPxx members in BBPARM.

```

BMC Software ----- PRIMARY OPTION MENU ----- AutoOPERATOR
OPTION  =====>                                     DATE   -- 01/02/01
                                                         TIME   -- 12: 22: 53
                                                         USERID -- BAOMXY2
                                                         MODE   -- ISPF 4. 8

Operator Workstations:
  1  ALERT Management                                5  NetView Resources
  2  MVS Resources                                  6  TapeSHARE
  3  CICS Resources                                 7  MQSeries
  4  IMS Resources

Automation:
  8  Basic and Advanced Automation                  9  Dynamic Parameter Manager

General Services:
  C  Service Refresh Cycle Setup                    K  Current PF Key Assignments
  L  Display Journal                                T  Tutorials
  M  Display Messages and Codes                     X  Terminate

                                                         PF1/13  HELP  PF3/15: EXIT

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

```

Figure 1. AutoOPERATOR PRIMARY OPTION MENU

When you select Option 9, the Dynamic Parameter Manager panel is shown (Figure 2).

```

BMC Software ----- Dynamic Parameter Manager ----- AutoOPERATOR
COMMAND ==>
Primary command: Add
Date --- 01/02/09
Time --- 11:41:36

LC CMDS --- (S)elect, (A)ctivate

Cmd  Member          Lib VV.MM  Created      Changed      Size  ID
---  -
_   AAOALS00          1  01. 11  01/01/08  01/01/28 20:12    4   JDB1
_   AAOALS01          1  00. 01  01/01/08  01/01/08 13:30    3   JDB1
_   AAOARP01          1  00. 01  01/01/08  01/01/08 13:30    3   JDB1
_   AAOEXP00          1  02. 26  01/01/12  01/01/28 20:12   27   JDB1
_   AAOTSP00          1  02. 01  01/01/12  01/01/28 20:12   12   JDB1
***** END OF MEMBERS *****

```

Figure 2. Dynamic Parameter Manager Panel—Example 1

Use this display to see a list of all the AAOALSxx, AAOARPxx, AAOEXPxx, and AAOTSPxx members in BBPARM where

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

From this display, you can

- Use the line command Select in the Cmd column to select a member to view and modify the parameters, if you choose.

For more information about selecting an AAOALSxx member, refer to “Modifying ALERTs Thresholds in AAOALSxx” on page 36.

For more information about selecting an AAOARPxx member, refer to “Modifying Automation Reporter Application Parameters in AAOARPxx” on page 39.

For more information about selecting an AAOEXPxx member, refer to “Modifying EXEC Parameters in AAOEXPxx” on page 44.

For more information about selecting an AAOTSPxx member, refer to “Modifying TapeSHARE for AutoOPERATOR Parameters in AAOTSPxx” on page 54.

- 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 may 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, or AAOTSPxx member.

For more information about activating your changes, refer to “Dynamically Implementing New Parameter Settings with the Activate Command” on page 62.

- Use the primary command Add to add a new AAOALSxx, AAOARPxx, AAOEXPxx, or AAOTSPxx member. To add new member(s), type one of the following:

ADD AAOALSxx

ADD AAOARPxx

ADD AAOEXPxx

ADD AAOTSPxx

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

ADD AAOEXP02

Describing the Fields

The fields for the Dynamic Parameter Manager panel are:

Field	Description
Cmd	Allows you to enter the Select or Activating line command.
Member	Is the list of members that can be <ul style="list-style-type: none">• AAOALSxx for ALERTs thresholds• AAOARPxx for Automation Reporter operation• AAOEXPxx for EXEC thresholds• AAOTSPxx for TapeSHARE operation
Lib	Is the library volume number.
VV.MM	Indicates the version level (VV) and modification level (MM).
Created	Is the date the member was created.
Changed	Is the date and time the member was last edited.
Size	Is the size (in number of lines) of the member.
ID	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 jobname of the subsystem used

Qualifying the Display

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:

Cmd	Member	Lib	VV	MM	Created	Changed	Size	ID
	AAOEXP							

The output shows only those members with the prefix AAOEXP.

Modifying ALERTs Thresholds in AAOALSxx

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

For information about...	See page...
Activating and modifying the Automation Reporter application	39
Modifying values for AutoOPERATOR EXECs	44
Customizing TapeSHARE	54

To modify an AAOALSxx member, type S in the Cmd field of the Dynamic Parameter Manager panel. Figure 3 shows an example of the panel.

```

BMC Software ----- Dynamic Parameter Manager ----- AutoOPERATOR
COMMAND ==>>                                     TGT ==>> BBSYSA
Primary command: Add                               Date --- 01/02/19
                                                    Time --- 11:41:36

LC CMDS --- (S)elect, (A)ctivate

Cmd  Member          Lib VV.MM  Created      Changed      Size  ID
---  -
s    AAOALS00         1 01.11   01/03/08   01/03/28 20:12    4   JDB1
-    AAOALS01         1 00.01   01/03/08   01/03/08 13:30    3   JDB1
-    AAOARP01         1 00.01   01/01/08   01/01/08 13:30    3   JDB1
-    AAOEXP00         1 02.26   01/05/12   01/03/28 20:12   27   JDB1
-    AAOTSP00         1 02.01   01/07/12   01/03/28 20:12   12   JDB1
***** END OF MEMBERS *****

```

Figure 3. Dynamic Parameter Manager Panel—Example 2

Use this display to see a list of all the AAOALSxx, AAOARPxx, AAOEXPxx, and AAOTSPxx members in BBPARM where

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

From this display, you can

- Use the Select line command in the Cmd column to select a member to view and modify the parameters, if you choose.
- Use the Activate line command when you want to dynamically change the BBPARM member that your system is currently using to manage either EXEC thresholds, ALERTs storage, or the Automation Reporter or TapeSHARE applications.

The Activate command displays the Confirm Parameter Activation panel; see Figure 13 on page 62. This function allows you to verify that you want to dynamically change and use a new AAOALSxx, AAOARPxx, AAOEXPxx, or AAOTSPxx member.

For more information about activating your changes, refer to “Dynamically Implementing New Parameter Settings with the Activate Command” on page 62.

- Use the primary command Add to add a new AAOALSxx, AAOARPxx, AAOEXPxx, or AAOTSPxx member. To add a new member(s), type one of the following:

ADD AAOALSxx

ADD AAOARPxx

ADD AAOEXPxx

ADD AAOTSPxx

on the COMMAND line, where xx is a new suffix for a new member. For example, if you want to add a new member named AAOALS02, type:

ADD AAOALS02

Setting ALERTs Storage Thresholds

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

Figure 4 shows an example of this panel.

Member		Current	
MAXSTOR	1000	1000	Maximum storage (ext private) in K bytes
WARNLVL1	60	60	Initial threshold (percent)
WARNLVL2	75	75	Secondary threshold (percent)
ALERTNV	NO	NO	Default RETAIN setting for Alerts which have no explicit RETAIN setting. Requires a Warm Start for change to be effective.
Publish	ADD	ADD	Default PUBLISH setting for Alerts. Used for Alerts which have no explicit PUBLISH setting.

Press END to save changes, CANCEL to cancel changes

Figure 4. ALERTs Storage Thresholds Panel

This panel provides two columns of data:

- On the left, the value of the parameters in the AAOALSxx member for the target BBI-SS PAS

You can input 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

This side is display-only.

Describing Fields

Following is a description of the fields on this panel.

Field Name	Description
MAXSTOR	<p>Is the maximum amount of BBI-SS PAS extended private virtual storage to be allocated to ALERTs.</p> <p>Specify the storage amount in kilobytes and thousandths of megabytes. The default is zero, which means no checking is done.</p>
WARNLVL1	<p>The first warning threshold level for queued ALERTs.</p> <p>The default is 60%. This means that if MAXSTOR=1K, a warning message is issued when the amount of virtual storage allocated to queued ALERTs reaches .6K. The message looks like:</p> <pre>AU6100W ALERT STORAGE USAGE 60% OVER 1ST LEVEL (60%)</pre>
WARNLVL2	<p>Is the second warning threshold level for queued ALERTs.</p> <p>The default is 75%. This 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:</p> <pre>AU6100W ALERT STORAGE USAGE 75% OVER 2ND LEVEL (75%)</pre> <p>When the maximum amount of storage is used (as set by MAXSTOR), two additional messages are produced. The first is:</p> <pre>EM0022E ERROR PROCESSING .. DEMOAL22 .. ALERT NOT ADDED Q IS FULL</pre> <p>where DEMOAL22 is the name of the ALERT that was not added to the ALERT queue because the maximum storage limit has been reached.</p> <p>The second message is:</p> <pre>EM0020E - IMFEXEC ALERT DASD5. 11: 57: 28 ' RESERVE IN PROGRESS. . .</pre> <p>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.</p> <p>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:</p> <pre>AU6200I ALERT STORAGE USAGE 58%, NO LONGER OVER THRESHOLD</pre>

ALERTNV

Controls the default setting for the RETAIN parameter of IMFEXEC ALERT statements.

The default is NO. Specifying YES changes the default of all generated ALERTs to IMFEXEC ALERT...RETAIN(YES) which means that all generated ALERTs will be retained in disk storage across BBI-SS PAS restarts and OS/390 IPLs.

Specifying NO means that all ALERTs will not be retained across BBI-SS PAS restarts or OS/390 IPLs. When NO is specified in AAOALSxx, you can still override the setting and create non-volatile ALERTs by specifying RETAIN(YES) on a specific IMFEXEC ALERT statement. This means only that ALERT will be retained across BBI-SS PAS restarts and OS/390 IPLs.

Note: When an ALERT is targeted to a different BBI-SS PAS, the ALERT defaults to the ALERTNV setting of the target BBI-SS PAS.

When you change the setting on the ALERTNV parameter, the BBI-SS PAS must be warm started for the change to take effect.

PUBLISH

Controls whether AutoOPERATOR should (by default) publish ALERTs to PATROL EM consoles that are attached through GME.

Possible values are as follows:

ADD: publish, but do not replace a previously published ALERT

REPLACE: publish, but delete a previously published ALERT before adding the new one.

NO: do not publish ALERTs

The default is ADD.

Modifying Automation Reporter Application Parameters in AAOARPxx

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

For information about...	See page...
Modifying storage values for AutoOPERATOR ALERTs	36
Modifying values for AutoOPERATOR EXECs	44
Customizing TapeSHARE	54

To modify an AAOARPx member, type S in the Cmd field of the Dynamic Parameter Manager panel. Figure 5 shows an example of the panel.

```

BMC Software ----- Dynamic Parameter Manager ----- AutoOPERATOR
COMMAND ==>>
Primary command: Add
Date --- 01/02/19
Time --- 11:41:36

LC CMDS --- (S)elect, (A)ctivate

Cmd  Member      Lib VV.MM  Created      Changed      Size  ID
---  ---
-   AAOALS00      1  01.11  01/03/08  01/03/28 20:12      4   JDB1
-   AAOALS01      1  00.01  01/03/08  01/03/08 13:30      3   JDB1
-   AAOARP01      1  00.01  01/01/08  01/01/08 13:30      3   JDB1
-   AAOEXP00      1  02.26  01/05/12  01/03/28 20:12     27   JDB1
-   AAOTSP00      1  02.01  01/07/12  01/03/28 20:12     12   JDB1
***** END OF MEMBERS *****

```

Figure 5. Dynamic Parameter Manager Panel—Example 3

Use this display to see a list of all the AAOALSxx, AAOARPx, AAOEXPxx, and AAOTSPxx members in BBPARM where

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

From this display, you can

- Use the line command Select in the Cmd column to select a member to view and modify the parameters, if you choose.
- Use the line command Activate when you want to dynamically change the BBPARM member that your system is currently using to manage EXEC thresholds, ALERTs storage, or the Automation Reporter or TapeSHARE applications.

The Activate command displays the Confirm Parameter Activation panel; see Figure 13 on page 62. This function allows you to verify that you want to dynamically change and use a new AAOALSxx, AAOARPx, AAOEXPxx, or AAOTSPxx member.

For more information about activating your changes, refer to “Dynamically Implementing New Parameter Settings with the Activate Command” on page 62.

- Use the primary command Add to add a new AAOALSxx, AAOARPx, AAOEXPxx, or AAOTSPxx member. To add a new member(s), type one of the following:

ADD AAOALSxx

ADD AAOARPx

ADD AAOEXPxx

ADD AAOTSPxx

on the COMMAND line, where xx is a new suffix for a new member. For example, if you want to add a new member named AAOARP02, type:

ADD AAOARP02

Setting Automation Reporter Parameters

Use the Automation Reporter Params panel to see and modify parameters for the Automation Reporter application and control how often data is collected, name a data set to hold the data, when the collected data is offloaded to the data set, and so on.

Figure 6 shows an example of this panel.

```

BMC Software ----- Automation Reporter Params ----- AutoOPERATOR
COMMAND ==>>>                                     TGT --- MS@1
                                                    DATE --- 01/02/28
                                                    TIME --- 13:39:59

                Member                Current

New DSN ==>>>                Current ===

Di sp    ==>>>      MOD                Dataset disposition (SHR, MOD)

Interval ==>>>      15                Interval minutes (10 - 1440, or 0)

Off int  ==>>>      1440              Offload interval minutes (60 - 1440)

Alert Queue ==>>>  MAIN                Alert Queue

Press END to save changes, CANCEL to cancel changes
  
```

Figure 6. Automation Reporter Params Panel

This panel provides two columns of data:

- On the left, the value of the parameters in the AAOARPxx member for the target BBI-SS PAS
You can input 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

This side is display-only.

The first time you bring up this panel, the display-only field `Current ===` is blank and all the values in the display-only side of this panel (the right side) are blank.

This is because the first time you bring up the Automation Reporter, there are no current data set values to display in these fields. After you have used the Automation Reporter once, the next time you access this panel, the `Current ===` field and all the display fields will show the values you previously entered.

The left side of the panel will show the default values in BBPARM member AAOARPxx, as shown in this panel.

Describing Fields

Following is a description of the fields on this panel.

Field Name	Description
New DSN	<p>Specify the name of the sequential data set you have pre-allocated.</p> <p>The user-defined data set should be a fixed block sequential data set and the maximum record length 580 bytes. This data set must be pre-allocated before the Automation Reporter can be activated.</p>
Disp	<p>Specify the disposition of the user-defined sequential data set.</p> <p>Valid disposition values are SHR (share) and MOD (modify), which is the default value. Specifying SHR causes the Automation Reporter to overwrite existing data. Specifying MOD causes the Automation Reporter to append data to the end of the data set from each offload cycle.</p>
Interval	<p>Specify the collection interval (in minutes) for the Automation Reporter to collect data.</p> <p>For example, if you specify 60, then every 60 minutes the Automation Reporter will record the data in the subsystem for the resources and activities.</p> <p>Valid collection intervals range from 10 minutes to 1440 minutes. The default is 15 minutes.</p>
Off Int	<p>Specify the offload interval (in minutes) for the Automation Reporter to offload the data it has collected to the data set.</p> <p>For example, if you specify 120, then every 2 hours the Automation Reporter writes the data it has collected to the sequential data set.</p> <p>Valid offloading intervals are 60 minutes to 1440 minutes. The default is 1440 (or once a day).</p>
Alert Queue	<p>Specify the name of an AutoOPERATOR ALERT queues to which any AutoOPERATOR ALERTs created by the Automation Reporter will be queued to.</p> <p>For example, if you specify ARALRT, then ALERTs will be queued to a queue named ARALRT.</p> <p>Any valid AutoOPERATOR ALERT queue name can be entered in this field (default queue name is MAIN). You do not have to enter the name of a pre-existing ALERT queue. You can enter the name of a new queue and it will be created for you when the Automation Reporter is activated and ALERTs are generated.</p>

Modifying EXEC Parameters in AAOEXPxx

BBPARM member AAOEXPxx contains parameters that determine the resource throttle settings and the environment for AutoOPERATOR EXECs.

For information about...	See page...
Modifying storage values for AutoOPERATOR ALERTs	36
Activating and modifying the Automation Reporter application	39
Customizing TapeSHARE	54

To modify an AAOEXPxx member, enter S in the Cmd field next to a member on the Dynamic Parameter Manager panel (refer to Figure 2 on page 33).

Figure 7 shows an example of the panel that is displayed when you select an AAOEXPxx member called AAOEXP00.

```

BMC Software ----- Dynamic Parameter Manager Member ----- AutoOPERATOR
COMMAND ==>>>                                     TGT --- BBSYSA
                                                    Date --- 01/02/19
                                                    Time --- 11:41:44

      Member Name: AAOEXP00 - AutoOPERATOR EXEC Specifications

LC CMDS --- (S)elect

Cmd   Panel Description
-    EXEC Thresholds
-    EXEC Parameters
  
```

Figure 7. Dynamic Parameter Manager Member Panel

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

- EXEC Thresholds
 - Allows you to 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 45.

- EXEC Parameters

Allows you to specify parameters for your EXECs themselves; for example:

- Whether EXEC parameters will be translated to uppercase
- What the maximum number of OSPI ACBs will be

For more information about selecting EXEC PARMS, refer to “Setting EXEC Parameters” on page 50.

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 you want to be able to support, and so on.

Figure 8 shows an example of the EXEC Thresholds panel.

BMC Software		EXEC Thresholds		AutoOPERATOR
COMMAND ==>>				TGT --- BBSYSA
				DATE --- 01/02/19
				TIME --- 11:41:52
	Member	Current		
EMABCNT ==>>	5	5	# of ABENDS allowed before EM terminated	
EXABCNT ==>>	3	3	# of ABENDS allowed before EXEC disabled	
MAXNORM ==>>	5	5	Maximum concurrent Normal EXECs	
MAXHIGH ==>>	3	3	Maximum concurrent High Priority EXECs	
MAXNORMQ ==>>	5	5	Maximum Normal EXECs Queued	
MAXHIGHQ ==>>	3	3	Maximum High Priority EXECs Queued	
WARNLVL1 ==>>	65	65	Initial threshold (percent)	
WARNLVL2 ==>>	80	80	Secondary threshold (percent)	
SELLIM ==>>	10	10	Maximum depth for SELECT with WAIT(YES)	
MAXTPUT ==>>	0	0	Maximum TPUT allowed in each EXEC	
Press END to save changes, CANCEL to cancel changes				

Figure 8. EXEC Thresholds Panel

This panel provides two columns of data:

- On the left, the value of the parameters in the AAOEXPxx member for the target BBI-SS PAS
You can input 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
This side is display-only.

Scheduling EXECs

The EXEC Thresholds panel allows you to 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 WARNLVL1 and WARNLVL2 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 there are 75 EXECs scheduled and when 112 EXECs scheduled.

```
WARNLVL1  50% OF 150 = 75
WARNLVL2  75% OF 150 = 112
```

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

```
WARNLVL1  50% OF 100 = 50
WARNLVL2  75% 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 WARNLVL2 fields in “Describing Fields” for more information.

Describing Fields

Following is a description of the fields on this panel.

Field Name	Description
------------	-------------

Member	Specifies the values for the new AAOEXPxx member.
---------------	---

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:

- The AAOEXPxx member used during the most recent BBI-SS PAS initialization
- The most recently applied AAOEXPxx member

EMABCNT	Is the maximum number of abends the EXEC Manager can sustain before it shuts down and requires a restart of the BBI-SS PAS. When the EXEC Manager stops, no EXECs can run in the system. Valid values range from 1 to 99. The default is 5.
----------------	---

Use this threshold to avoid recursive or persistent abends that demand attention.

Abends in individual EXECs do not count against this threshold. The count is cumulative from the time the target BBI-SS PAS was started. The reason for this is that generally, abends in individual EXECs are more frequent and are caused by coding errors within REXX or coding errors within an IMFEXEC statement. These abends usually affect only 1 or 2 EXECs or a set of similar EXECs. Multiple failures of this type are not serious enough to warrant shutting off all automation but are serious enough to warrant shutting off the failing EXECs.

However, EXEC Manager abends are much less frequent and far more serious. They can usually be recovered from, but if several occur, it is better to stop AutoOPERATORAutoOPERATOR and restart it to avoid further errors.

EXABCNT	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.
----------------	--

MAXNORM	Is the maximum number of EXECs that can execute concurrently on Normal priority threads. Valid values range from 1 to 99. The default is 1.
----------------	---

MAXHIGH	Is the maximum number of EXECs that can execute concurrently on high-priority threads. Valid values range from 1 to 99. The default is 5.
----------------	---

MAXNORMQ 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. 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:

```
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:

```
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. 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:

```
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:

```
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.

WARNLVL1 Specifies a preliminary percentage of EXECs in a queue that, when met, generates a warning message. Valid values range from 0% to 99%. 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 there are 60 EXECs queued to the High-priority queue. The message looks like:

```
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%. 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 there are 75 EXECs queued to the high-priority queue. The message looks like:

```
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. The default is 0.

This parameter is designed to control recursive calls (where EXECA schedules EXECB, that 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.

MAXTPUT Is the maximum number of TSO TPUTs allowed for each execution of an EXEC. Valid values range from 0 to 999. 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 JOURNALLING SUSPENDEED, BACKLOG HAS EXCEEDED MAXIMUM
```

and subsequent TPUTs are ignored until the EXEC terminates.

Setting EXEC Parameters

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

Figure 9 shows an example of the panel.

```

BMC Software ----- EXEC Parms ----- AutoOPERATOR
COMMAND ==>
                                                    TGT --- BBSYSA
                                                    DATE --- 01/02/19
                                                    TIME --- 11:42:09

                Member          Current
TIMEXLIM ==>      0              0  Maximum CPU time limit for an EXEC
PEREXLIM ==>      0              0  Maximum CPU percentage usage for an EXEC
TSOTIME  ==>     30             30  Timeout value for attached TSO commands

UCPARMS  ==> NO                NO   Translate EXEC parms to UPPERCASE
AUDITMSG ==> YES               YES  Write JRNL msg for BLDL and submit EXEC
OSPINUM  ==>      5              5   Maximum # of OSPI ACBs
OSPIPRFX ==> OSPI              OSPI 4 character OSPI ACB prefix
UNITNAME ==> BABDA             BABDA UNITNAME used for allocating a dataset
PREFIX   ==> JDB1              JDB1  Prefix used for allocation a dataset
SUBXAUTH ==> DEFER             DEFER RACF/JOBNAME/DEFER - IMFSUBEX security

REXX     ==> YES               YES   REXX support

EXEC     ==>                  (Enter ? to see high priority EXECs)

Press END to save changes, CANCEL to cancel changes

```

Figure 9. EXEC Parms Panel

Use this panel to continue setting parameters for how single EXECs will 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 AAOEXPxx member for the target BBI-SS PAS
You can input 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
This side is display-only.

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 53 for more information about specifying EXEC names for high-priority queuing.

Describing Fields

Following is a description of the fields on this panel.

Field Name	Description
TIMEXLIM	<p>Is the maximum CPU time limit allowed for an EXEC to use while executing. Valid values range from 0 to 9999. The default is 0.</p> <p>Set this parameter to ensure that any EXEC does not execute limitlessly.</p>
PEREXLIM	<p>Is the maximum CPU percentage usage allowed for an EXEC to use while executing. Valid values range from 0 to 99. The default is 0.</p> <p>Set this parameter to ensure that any EXEC does not use a disproportionate amount of the CPU time.</p>
TSOTIME	<p>Is the timeout value (in seconds) for attached TSO commands. Valid values range from 0 to 9999. The default is 1200.</p> <p>Set this parameter to limit the amount of time a TSO command associated with an EXEC can use.</p>
UCPARMS	<p>Specifies that parameters passed to an EXEC must be converted to uppercase characters. Valid values are YES and NO. The default is YES. All CLIST EXECs require uppercase characters for parameters.</p>
AUDITMSG	<p>Specifies if 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.</p> <p>Valid values are YES and NO. The default is YES.</p>
OSPINUM	<p>Specifies the maximum number of OSPI ACBs allowed. Valid values range from 0 to 999. The default is 0.</p>
OSPIPRFX	<p>Specifies the default 4-character OSPI ACB prefix. Valid values are any 4-character string. The default is OSPI.</p>
UNITNAME	<p>Is the default UNITNAME used for allocating a data set. Valid values are any 8-character string. The default is SYSALLDA.</p>
PREFIX	<p>Is the prefix used for allocating a data set in an EXEC if the data set name is not enclosed in quotes. Valid values are any 7-character string. The default is blank, which refers to the subsystem ID.</p>

SUBXAUTH Specifies one of the following:

Name	Meaning
DEFER	Defers using the jobname 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 jobname for all batch job security checks.

The default is DEFER.

REXX Shows the current REXX= setting. Valid values are YES or NO. The default is YES.

Restriction
<p>When REXX=YES, if the total of the MAXNORM and MAXHIGH values exceeds the number of available REXX Environment Blocks (REBs), the number of available REBs will determine how many EXECs can execute simultaneously. When REXX=NO, the total number of EXECs that may be scheduled concurrently is limited by the number of REXX Environment Blocks defined in IBM module IRXANCHR. The number of REBs is user-modifiable.</p> <p>The number of EXECs that may be scheduled concurrently is equal to $(x - 3) / 2$ 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:</p> <p>EM0002W EXEC MULTITASKING EXCEEDS SITE LIMITS</p> <p>In addition, further scheduling of EXECs is inhibited if this limit is reached.</p>

EXEC Allows 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 53 for more information about this panel.

CAUTION
<p>This application does not allow the dynamic change of the REXX= parameter. To change the REXX= parameter, you must terminate and restart the BBI-SS PAS.</p>

Setting High-Priority EXECs

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

Figure 10 shows an example of the panel.

```
BMC Software ----- Update Non-Unique Parameter List ----- AutoOPERATOR
COMMAND ==>
Parameter: EXEC=      Enter ADD xxxx to add new entries          TGT --- BBSYSA
LC CMDS --- (D)el ete                                         DATE --- 01/02/19
                                                                TIME --- 11:27:27

- EXEC1
- EXEC2
- EXEC3
- EXEC4
- EXECABCD
- EXECEFGH
***** END OF ENTRIES *****
```

Figure 10. Update Non-Unique Parameter List

To add an EXEC to for High priority execution, type:

ADD *execname*

where *execname* is the 1-8 character name of the EXEC 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 you want to delete.

Modifying TapeSHARE for AutoOPERATOR Parameters in AAOTSPxx

BBPARM members AAOTSPxx contain the parameters that you can modify to customize the way the TapeSHARE for AutoOPERATOR component automates sharing tape devices among your OS/390 images.

Remember, the TapeSHARE for AutoOPERATOR 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*.

For information about...	See page...
Modifying storage values for AutoOPERATOR ALERTs	36
Activating and modifying the Automation Reporter application	39
Modifying values for AutoOPERATOR EXECs	44

To modify an AAOTSPxx member, type S in the **Cmd** field of the Dynamic Parameter Manager panel.

Figure 11 shows an example of the panel.

```

BMC Software ----- Dynamic Parameter Manager ----- AutoOPERATOR
COMMAND ==>                                     TGT ==> BBSYSA
Primary command: Add                               Date --- 01/02/19
                                                    Time --- 11:41:36

LC CMDS --- (S)elect, (A)ctivate

Cmd  Member          Lib VV.MM  Created      Changed      Size  ID
---  ---
_   AAOALS00          1 01.11  01/03/08  01/03/28 20:12    4   JDB1
_   AAOALS01          1 00.01  01/03/08  01/03/08 13:30    3   JDB1
_   AAOARP01          1 00.01  01/01/08  01/01/08 13:30    3   JDB1
_   AAOEXP00          1 02.26  01/05/12  01/03/28 20:12   27   JDB1
S   AAOTSP00          1 02.01  01/07/12  01/03/28 20:12   12   JDB1
***** END OF MEMBERS *****

```

Figure 11. Dynamic Parameter Manager Panel—Example 4

Use this display to see a list of all the AAOALSxx, AAOARPxx, AAOEXPxx, and AAOTSPxx members in BBPARM where

- The AAOALSxx members contain parameters that determine how much extended storage AutoOPERATOR ALERTs use on your system
- The AAOARPxx members contain parameters that control how the Automation Reporter application operates
- The AAOEXPxx members contain parameters and thresholds that determine how efficiently AutoOPERATOR EXECs execute on your system

- The AAOTSPxx members contain parameters that control how TapeSHARE operates

From this display, you can:

- Use the line command **Select** in the **Cmd** column to select a member to view and modify the parameters if you choose.
- Use the line command **Activate** when you want to dynamically change the **BBPARAM** member that your system is currently using to manage **EXEC** thresholds, **ALERTs** storage, or the **Automation Reporter** or **TapeSHARE** applications.

The **Activate** command displays the **Confirm Parameter Activation** panel; see **Figure 13** on page 62. This function allows you to verify that you want to dynamically change and use a new **AAOALSxx**, **AAOARPxx**, **AAOEXPxx**, or **AAOTSPxx** member.

For more information about activating your changes, refer to “**Dynamically Implementing New Parameter Settings with the Activate Command**” on page 62.

- Use the primary command **Add** to add a new **AAOALSxx**, **AAOARPxx**, **AAOEXPxx**, or **AAOTSPxx** member. To add a new member(s), type one of the following:

ADD AAOALSxx

ADD AAOARPxx

ADD AAOEXPxx

ADD AAOTSPxx

on the **COMMAND** line, where **xx** is a new suffix for a new member. For example, if you want to add a new member named **AAOTSP02**, type:

ADD AAOTSP02

Customizing TapeSHARE

Use the TapeSHARE Parameters panel to see and modify parameters for the TapeSHARE application and control how TapeSHARE automates sharing tape devices among your OS/390 images.

Figure 12 shows an example of this panel.

```

BMC Software ----- TapeSHARE Parameters ----- AutoOPERATOR
COMMAND ==>>>                                     TGT --- DSHD
                                                    DATE --- 01/0201//19
                                                    TIME --- 16:27:51

                Member      Current

PARTNER      ==>>>                * LIST * (Enter ? for a partner list)

TIMEOUT      ==>>>      120      120      Subsystem response timeout value
RETRYCNT     ==>>>       2       2       Times to retry a take request
RETRYINT     ==>>>      30      30      Interval between take retries
ACTION       ==>>>  DEFAULT  DEFAULT Allocation failure action

NOGIVE       ==>>>
NOTAKE       ==>>>                (Enter ? for a device list)
                (Enter ? for a device list)

PREF         ==>>>  NO         NO         Enable / Force preferencing
PREFDEV      ==>>>
                (Enter ? for a device list)

FREE         ==>>>  DEALLOC   DEALLOC   Deallocation procedure name
TRACE        ==>>>  NO         NO         Start / Stop TapeSHARE tracing

Press END to save changes, CANCEL to cancel changes
  
```

Figure 12. TapeSHARE Parameters Panel

This panel provides two columns of data:

- On the left, the value of the parameters in the AAOTSPxx member for the target BBI-SS PAS
You can input 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
This side is display-only.

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 Fields

Following is a description of the fields on this panel.

Field Name	Description
PARTNER	<p>Specify a BBI-SS PAS name (that can be associated with either one or a range of tape devices) that is a TapeSHARE PLEX partner.</p> <p>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.</p> <p>Example 1:</p> <p>PARTNER=SYSA</p> <p>In this example, SYSA is a partner to this image.</p> <p>Example 2:</p> <p>PARTNER=SYSC, 0123</p> <p>In this example, SYSC is a partner to this image and the device whose address is 0123 cannot be GIVEN to partner SYSC.</p> <p>Example 3:</p> <p>PARTNER=SYSB, 0120- 0127</p> <p>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.</p> <p>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.</p>
TIMEOUT	<p>Specify the amount of time (in seconds) to wait for the partners to respond to a request for devices.</p> <p>When this time is reached and a successful GIVE has not completed, an Allocation Failed Event occurs unless you have specified a number of retries on the RETRYCNT parameter. See below.</p> <p>Example:</p> <p>TIMEOUT=20</p> <p>Valid values are 0 - 999 seconds; 120 seconds is the default.</p>

RETRYCNT Specify the 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 three retries is specified. TapeSHARE will attempt 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.

Valid values are 0–10; 2 retries is the default.

RETRYINT Specify the number of seconds TapeSHARE waits for devices between attempts after an initial attempt has failed.

For example, suppose:

- The **RETRYCNT=** parameter is set to 3 retries
- The **RETRYINT=** parameter is set to 45 seconds
- 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

Valid values are 0 - 300; 30 seconds is the default.

ACTION Specify the action TapeSHARE should take when an Allocation Failed Event occurs because a successful GIVE cannot be performed.

Possible values and their definitions are:

- | | |
|----------------|--|
| DEFAULT | Allows the installation's default action to occur |
| WTOR | Issues a WTOR to the operator |
| | If the site does not have a defined default action, this is the default. |
| CANCEL | Cancels the job |
| NOHOLD | Allows the job to wait without holding resources |
| HOLD | Allows the job to wait while holding resources |

Example:

ACTI ON=WTOR

NOGIVE

Specify a list of device addresses (or a range of addresses) that should not be GIVEN. to other request devices to TAKE.

Note: This does not prevent you from manually GIVING a NOGIVE device.

If a device address is not associated with this parameter, TapeSHARE assumes there are no devices 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.

For 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 will not attempt to GIVE or TAKE this device.

The default value is none.

NOTAKE

Specify 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.

For 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 will not attempt to `GIVE` or `TAKE` this device.

The default value is none.

PREF

Specify whether or not 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 will be `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

Specify a tape device address (or a range of tape device addresses) followed by either an I for input-only or 0 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, 0

or

PREFDEV=0150-015F, 0

Default value is none.

FREE

For use with MVS SP4.x only

Specify 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.

TRACE

Specify that diagnostic information is written to the BBITST DD statement.

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).

Dynamically Implementing New Parameter Settings with the Activate Command

The Dynamic Parameter Manager's 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 DPM panel, examples of which are shown in Figure 2 on page 33 and Figure 3 on page 36.

Figure 13 on page 62 shows the Confirm Parameter Activation panel that is displayed after the Activate line command is entered.

```
BMC Software ----- Confirm Parameter Activation ----- AutoOPERATOR
COMMAND ==>>>                                         TGT --- BBSYSA
                                                         DATE --- 01/02/19
                                                         TIME --- 11: 43: 13

Current TARGET:      SYSB
Current SSID:        A022

Last Parm:           AAOEXP00 (Values may have been temporarily modified)
Replaced by:         AAOEXP00
Last update:         01/05/19 11: 42
User:                MKY1

Instructions:

  Press ENTER to confirm activation request.
    (The parameters in the current SS will be dynamically updated)

  Press END  to cancel activation request.
```

Figure 13. 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.

Important

CAUTION This application does not allow the dynamic change of the REXX= parameter. To change the REXX= parameter, you must terminate and restart the BBI-SS PAS.

Describing Fields

Following is a description of the fields on this panel.

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.

Chapter 4. MVS Console Considerations

MVS commands can be issued by AutoOPERATOR terminal session (TS) users, EXECs, or Rules. In some cases, a response to the command needs to be returned to the issuer. AutoOPERATOR uses MVS consoles to perform this function.

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

For more information about how TS users use consoles, refer to “Issuing Commands from Terminal Sessions” on page 177. For more information about how EXECs use consoles, refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* and the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs* where the IMFEXEC CMD statement is documented.

This chapter describes how to allocate MVS consoles to AutoOPERATOR subsystems which are used by TS users and EXECs.

The Os/390 consoles used by AutoOPERATOR are Extended MCS (X-MCS) consoles

Determining How Many Consoles to Allocate

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

The number of consoles allocated to AutoOPERATOR controls the number of EXECs that can issue MVS commands and receive a response concurrently. A console is allocated exclusively to an EXEC while a command requesting a response is being processed.

If all of AutoOPERATOR's consoles are in use, a new command (EXEC) must wait until a console becomes available. This could adversely impact the performance of the requesting EXEC and, possibly, the entire AutoOPERATOR subsystem.

For this reason, BMC Software recommends allocating one console for each EXEC thread defined in AutoOPERATOR. Refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* and the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs* for information about EXEC threads.

Note: The Continuous State Manager (CSM) application requires a minimum of three consoles.

Extended MCS (X-MCS) Consoles

Unlike subsystem consoles, MVS definitions are not required for X-MCS consoles. AutoOPERATOR (and other products or subsystems such as NetView) dynamically request that MVS create X-MCS consoles as they are needed. There is no limit on the number of X-MCS consoles created in an MVS system or sysplex.

Prior to MVS Version 4, all consoles have only a 1-byte console ID. Beginning with MVS Version 4, all consoles (subsystem, MCS, and X-MCS) have a 4-byte console ID and an 8-byte console name. AutoOPERATOR creates all X-MCS console names using the format:

SSI Dnnnn

where:

SSID Is the BBI-SS PAS identifier name

nnnn Is a number from 1 to the total number of X-MCS consoles created

Note: You must ensure that no other application uses these console names.

Some consoles may have a 1-byte console ID in addition to the new 4-byte console ID. For example, MCS consoles (defined in the CONSOLxx member of SYS1.PARMLIB) continue to have a 1-byte console ID in addition to the 4-byte console ID and 8-byte console name. However, X-MCS consoles usually do not have a 1-byte console ID; they have only a 4-byte console ID.

This means that applications that interface with consoles specified in the CONSOLxx member of SYS1.PARMLIB do not have to be updated to understand 4-byte console IDs.

Applications that will interface with X-MCS consoles need to be updated to understand 4-byte console IDs. You may have some applications **that do not yet understand 4-byte console IDs**. To remain compatible with these applications, MVS allows some X-MCS consoles to have a 1-byte migration ID (MIGID) specified. Therefore, X-MCS consoles that have a MIGID can interface with applications that have not yet been updated.

In addition, within a sysplex, MVS limits the number of X-MCS consoles with MIGIDs. For this reason, AutoOPERATOR does not request a MIGID for all X-MCS consoles it creates.

Refer to “Allocating X-MCS Consoles without MIGIDs” on page 66 for information about allocating X-MCS consoles without MIGIDs. Refer to “Allocating X-MCS Consoles with MIGIDs” on page 66 for information about allocating X-MCS consoles with MIGIDs.

Allocating X-MCS Consoles without MIGIDs

The number of X-MCS consoles that AutoOPERATOR creates without a migration ID is specified with the CONSOLES parameter in BBPARM member BBISSP00. Refer to the *MAINVIEW Common Customization Guide* for more information about specifying values for the CONSOLES parameter in BBPARM member BBISSP00.

Allocating X-MCS Consoles with MIGIDs

The total number of X-MCS consoles with a MIGID that AutoOPERATOR creates is specified using the MIGCONS parameter in BBPARM member BBISSP00.

MVS imposes a limit of 150 X-MCS consoles with a MIGID within a sysplex. In addition to the considerations in “Determining How Many Consoles to Allocate” on page 65, you also must consider this maximum limit when deciding how many X-MCS consoles with an MIGID to specify on the MIGCONS parameter. Refer to the *MAINVIEW Common Customization Guide* for more information about specifying MIGIDs with the MIGCONS parameter in BBPARM member BBISSP00.

AutoOPERATOR always allocates one extra X-MCS console with an MIGID to support terminal-session users. Refer to “Issuing Commands from Terminal Sessions” on page 177 for information about TS users using consoles with MIGIDs to issue commands.

Defining X-MCS Consoles to CICS

CICS will not interface with a console unless the console is defined by using CICS Resource Definition. For CICS/ESA version 4.1 and CICS Transaction Server, refer to “CICS/ESA Targets” on page 67.

CICS/ESA Targets

CICS/ESA version 4.1.0 (and later) customers need to define all AutoOPERATOR X-MCS console names. The console names use the format:

SSIDnnnn

where:

SSID Is the BBI-SS PAS identifier name

nnnn Is a number from 1 to the total number of X-MCS consoles created

Chapter 5. Implementing TapeSHARE

This chapter describes how to implement the TapeSHARE for 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.**

Important Note
To use TapeSHARE, you must make sure the OS/390 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 OS/390 when it initializes.

Prerequisites

To use TapeSHARE, the only prerequisites are that you have a valid AutoOPERATOR for OS/390 password key and that you have a password key for the TapeSHARE for AutoOPERATOR component. See “Step 1: Entering the Password Key for TapeSHARE (Required)” for more information about password keys.

Step 1: Entering the Password Key for TapeSHARE (Required)

1. TapeSHARE requires that you have a password key to use it.

Once you have a key, you must enter it in the BBPARM member BBKEYS. Refer to the section “Specify Product Option Password Keys” in the *MAINVIEW Common Customization Guide* for information about acquiring and using a password key for TapeSHARE.

2. Go to “Step 2: Activating TapeSHARE (Required)”.
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 (Required)

After the password key is entered into BBPARM member BBKEYS, all you need to do is restart the BBI-SS PAS and TapeSHARE is activated for that image. **You do not have to restart OS/390.**

For every image you want to include in a TapeSHARE PLEX, you must add a password key to the BBPARM member BBKEYS for that image and re-start the BBI-SS PAS (as described in “Step 1: Entering the Password Key for TapeSHARE (Required)”).

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 for AutoOPERATOR Parameters in AAOTSPxx” on page 54.

Step 3: Securing TapeSHARE (Optional)

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

Security for all AutoOPERATOR options is described in the BMC Software document *Implementing Security for MAINVIEW Products*.

Securing access to TapeSHARE is the same as securing access to any of the AutoOPERATOR applications, which is described in “Chapter 4, Reviewing Product Resources” of the *Implementing Security for MAINVIEW Products* manual.

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

After TapeSHARE Is Installed

Once TapeSHARE is installed, you can let it perform all 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.

Chapter 6. Implementing Automatic Restart Manager (ARM)

This chapter describes implementation considerations for the Automatic Restart Manager (ARM) and how to implement support for it.

Important Note

The Automatic Restart Manager (ARM) was introduced in MVS/ESA Version 5, Release 2. For ARM support to be available, you must be using MVS/ESA 5.2 (or higher) and AutoOPERATOR 4.1 (or higher).
--

Overview

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

ARM support is available in either single or multisystem SYSPLEX using couple data sets. By implementing support for ARM, you can ensure that AutoOPERATOR will be restarted quickly and automatically after an abend with no operator intervention required.

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

If your system is using MVS/ESA 5.2 or higher, you can choose to implement ARM support for the AutoOPERATOR. This eliminates the need for a secondary AutoOPERATOR system (or Elan) to monitor AutoOPERATOR availability and can simplify AutoOPERATOR restarts by allowing MVS/ESA to monitor and restart AutoOPERATOR.

Implementation Considerations

To implement ARM support for AutoOPERATOR, you must modify the ARMPOLCY parameter in BBPARM member BBISSP00 (see “Enabling ARM Support” on page 73). There are three choices:

- Do not implement ARM support.
- Implement ARM support with the default ARM policy, see 72.
- Implement ARM support with your own user-designed policy, see 72.

| 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 AutoOPERATOR, ARM will attempt to restart an abended 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 AutoOPERATOR subsystem originally. After the third attempt, if 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”.

Implementing ARM Support: Creating a User-Defined Policy

You may 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, AutoOPERATOR provides support for a user-defined ARM restart policy. “Enabling ARM Support” 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 OS/390 image at one time.

Enabling ARM Support

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

ARMPOLCY=[NO | YES | xxxxxxxx]

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.

xxxxxxx

Specifies a 1- to 8-character name of a user-created policy to be used to control BBI-SS PAS restarts

Once 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.

AutoCustomization provides a step for ARM support implementation. The *MAINVIEW Common Customization Guide* describes how to implement ARM support during Manual Customization.

Stopping AutoOPERATOR When ARM Is Enabled

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

- Use the STOP | P command for a normal shutdown of AutoOPERATOR; for example:

```
P ao_ssi d
```

- Use the CANCEL | C or FORCE command if you unexpectedly need to cancel AutoOPERATOR; for example:

```
C ao_ssi d
```

or

```
FORCE ao_ssi d
```

where ao_ssi d is the subsystem ID of the AutoOPERATOR you want to stop or cancel.

Chapter 7. Implementing Support for MPF

This chapter describes how 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 AutoOPERATOR provided the HONORMPF=(YES|NO) parameter, which determined how 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 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 AutoOPERATOR automation.

In AutoOPERATOR 4.1, you can specify that 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:

SUP=(YES|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 76.

Implementation Considerations

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

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

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

For more information, refer to “Setting the MPFFILTR Parameter” on page 76.

Setting the MPFFILTR Parameter

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

`MPFFILTR=(AUTO=YES|NO|token,SUP=YES|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:

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

You can select up to four of the possible combinations. Once you use any of the combinations, 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 AutoOPERATOR. The following examples describe some of the possible MPFFILTR settings.

Example 1

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 AutoOPERATOR automation. AutoOPERATOR ignores all other messages that have different AUTO and SUP settings; they are not eligible for AutoOPERATOR automation.

Example 2

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 AutoOPERATOR. AutoOPERATOR ignores all other messages that have different AUTO and SUP settings; they are not eligible for AutoOPERATOR automation.

Example 3

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 AutoOPERATOR.

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 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=BO OL, SUP=NO
AUTO=NETVIEW, SUP=YES

Example 4

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 will be eligible for automation regardless of the SETTING. This is equivalent to setting MPFFILTR to:

```
mpffiltr=(AUTO=YES, SUP=*)
```

AutoOPERATOR ignores all other WTOs that have different AUTO and SUP settings; they are not eligible for AutoOPERATOR automation.

Chapter 8. Activating the REXX/370 Alternate Library

This chapter describes how to implement the REXX/370 Alternate Library.

If your site does not have the IBM library for SAA REXX/370 installed and you want to run compiled REXX programs in a BBI-SS PAS, you must follow the instructions in this chapter to implement the REXX/370 Alternate Library.

If the IBM library for SAA REXX/370 is available, no customization is required on the AutoOPERATOR side. Please refer to the IBM documentation for any steps required to make the library accessible to the BBI-SS PAS.

To activate the REXX/370 Alternate Library:

1. Review the sample job distributed in BBSAMP member BBREXALT.

This job activates the REXX/370 Alternate Library. The REXX/370 Alternate Library must be activated at sites that need to execute compiled REXX programs in an BBI-SS PAS **but do not have the IBM library for SAA REXX/370 (program number 5695-014) available.**

Important
This job should be not be used at sites where the IBM library for SAA REXX/370 is available.

2. Edit the job, if necessary.
3. Follow all customization instructions.
4. Submit the job.

The REXX/370 Alternate Library is installed.

SMP/E Messages

You may receive the following SMP/E messages when you install compiled REXX programs. These are benign messages and you can ignore them.

For SMP/E release 1.4, you may receive:

```
GIM4441 ASSEMBLY FOR SRC aaaaaaa in SYSMOD BP0nnnn  
      WILL NOT BE DONE - RELATED MOD ENTRY NOT FOUND
```

For SMP/E release 1.5 and higher, you may receive:

```
GIM44402W aaaaaaaaa WAS NOT ASSEMBLED FOR SRC aaaaaaa  
      IN SYSMOD BP0nnnn BECAUSE THERE IS NO MOD  
      ENTRY FOR aaaaaaa. SMP/E CANNOT DETERMINE  
      THE TARGET LIBRARY FOR THE ASSEMBLER OUTPUT.
```

These messages are issued because compiled REXX elements are distributed as ++SRC elements.

Chapter 9. Defining OSPI Virtual Terminals

This chapter describes defining OSPI virtual terminals.

Overview

AutoOPERATOR provides the Open Systems Procedural Interface (OSPI) 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, AutoOPERATOR has logon capabilities and complete access to any VTAM application's data stream. In this way, 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 241 for more information.

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

- “Defining the ACB Pool” on page 82 contains instructions for defining OSPI virtual terminals to VTAM.
- “Defining the ACB Pool to AutoOPERATOR” on page 82 contains instructions for defining OSPI virtual terminals to AutoOPERATOR.
- “OSPI Application Definitions” on page 83 contains information about local terminal definitions that might be required for some applications.

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 describing the OSPI Scripting facility and the IMFEXEC LOGON command in the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* and the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs* for information about requesting a specific ACB.

Naming Conventions

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

PPPPnnnn

where:

PPPP Is the 4-character pool prefix

nnnn Is a sequential number from 0000 to 9999

The default pool prefix is OSPI. You can specify a different prefix 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 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:

```
OSPI      VBUILD TYPE=APPL
OSPI 0000 APPL AUTH=(ACQ, NOTSO, VPACE)
OSPI 0001 APPL AUTH=(ACQ, NOTSO, VPACE)
```

The number of ACBs defined determines the maximum number of concurrent OSPI sessions 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 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 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 241](#) 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.

Chapter 10. 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 AutoOPERATOR for CICS, use the *MAINVIEW for CICS Customization Guide* and perform the steps described in:

- "Chapter 3: Standard Implementation Procedures"

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

Certain AutoOPERATOR for CICS functions require BBI-SS PAS to CICS communication. However, there are numerous AutoOPERATOR for CICS functions available to you if you do not implement BBI-SS PAS-to-CICS communication. See "AutoOPERATOR for CICS Functions Available after Implementing BBI-SS PAS-to-CICS Communication" on page 89 for additional information.

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

- Review implementation considerations (page 85).
- Specify resources to CICS (page 86).
- Modify CICS startup JCL (page 88).
- Restart CICS target regions (page 88).

To begin implementing BBI-SS PAS-to-CICS communication, go to "Implementation Considerations" on page 85.

Implementation Considerations

This section provides information you should consider when you are implementing MAINVIEW for CICS or MAINVIEW 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 RACF or 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 Appendix B, “Changing the Use of CICS Recoverable Temporary Storage” on page 179.

Specifying Resources to CICS

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

Making the AutoOPERATOR Transactions and Programs Available to CICS/ESA and CICS Transaction Server

The member CMRCSDES in the **prefix.BBSAMP** data set contains JCL that refers to other **prefix.BBSAMP** members that contain in-line PCT and PPT definitions.

Use this JCL to define BMC Software transactions and programs required for BBI-SS PAS to CICS communication by doing the following:

1. Copy CMRCSDES from **prefix.BBSAMP** to a data set that is not a BMC Software product target or distribution library. 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 modify the JCL accordingly.
3. Submit the JCL and check the output. Be sure the return code is not higher than 4.
4. Verify that the BMC Software transaction IDs do not conflict with any existing transaction IDs.

If a conflict exists, and the existing transaction IDs cannot be changed, see Appendix C, “CICS Transaction and Program Names” on page 181.

The AutoOPERATOR transactions and programs are now available to CICS.

Go to “Modifying the Program List Table (PLT)” on page 87.

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 OS/390 console.

Refer to Appendix D, “Manual Activation of BBI-SS to CICS Communication” on page 187 for more information.

Assembly Note
To assemble the table changes, concatenate BBSAMP with SYSLIB in your site's CICS table assembly JCL (normally, DFHAUPLE).

Existing PLT

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

```
DFHPLT TYPE=FINAL
```

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

```
CMRPLT RELEASE=41 | 12 | 13 | 21 | 22
```

Table 2 describes the parameters for the CMRPLT macro.

Table 2. Syntax for the CMRPLT Macro

Term	Parameters	Description
RELEASE	41 12 13 21 22	Specifies the release of the CICS system that will use the resulting table (for example, CICS 4.1.0, 2.1.0, 2.2.0).
SUFFIX	BI	<p>Specifies the suffix for the new PLT.</p> <p>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 which uses the PLT.</p> <p>When SUFFIX is used, TYPE=INITIAL and TYPE=FINAL statements are generated automatically.</p>

3. Assemble the newly revised PLT.

The PLT is modified.

No existing 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:

```
CMRPLT RELEASE=41 | 12 | 13 | 21 | 22
```

Table 2 describes the parameters for this macro.

2. Assemble the newly created PLT.
3. Add the following parameter to the CICS initialization override parameters:

```
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 88.

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:

```
//BBI PARM 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 will communicate with a BBI-SS PAS but for which the procedure in “Modifying the Program List Table (PLT)” on page 87 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 OS/390 console.

Refer to “Manual Activation of BBI-SS to CICS Communication” on page 187 for more information.

5. 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 implementation of BBI-SS PAS to CICS communication is now complete.

AutoOPERATOR for CICS Functions Available after Implementing 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:

ACQUIRE TERMINAL	Acquire a VTAM-supported terminal.
ALLOC	Allocate a data set to a CICS region.
CEMT	Issue a CICS master terminal command.
CHAP	Change a task's priority in a CICS region.
CLOSE	Close a file in a CICS region.
DISABLE	Disable a CICS resource.
DROP	Decrease the use count of a CICS program.
DUMPDB	Prepare a database for dumping.
ENABLE	Make a CICS resource available for use.
FREE	Deallocate a data set from a CICS region.
INSERVE	Place a CICS resource in service.
KILL TASK(PURGE, FORCEPURGE)	Terminate a task in a CICS region by task number.
KILL TERM	Terminate a task in a CICS region by term ID.
LOAD	Load a program in a CICS region.
NEWCOPY	Load a new version of a program in a CICS region.
OPEN	Open a file in a CICS region.
OUTSERVE	Place a CICS resource out of service.

PURGE	Purge a resource in a CICS region.
RECOVERYDB	Prepare a database for recovery.
RELEASE TERMINAL	Release a VTAM-supported terminal.
STARTDB	Start a database.
STOPDB	Stop a database.

AutoOPERATOR for CICS Functions Available if BBI-SS PAS-to-CICS Communication is not Implemented

If you choose **not** to implement BBI-SS PAS-to-CICS communication, you can still use 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:

ALLOC (LOCAL only)	Allocate a data set to the BBI-SS.
ALTER	Change CICS task-related throttles.
ALTERVS	Change virtual storage in the CICS region.
DISABLE (TRAN, DEST, PROGRAM)	Disable a CICS resource.
ENABLE (TRAN, DEST, PROGRAM)	Make a CICS resource available for use.
FREE (LOCAL only)	Deallocate a data set from the BBI-SS.
KILL TASK (KILL, WITH DUMP, FORCE)	Terminate a task in a CICS region by task number.
QUERY	Invoke a MAINVIEW for CICS service.
SPURGE	Change the spurge value for CICS transactions.

Chapter 11. AutoOPERATOR for CICS Optional Tailoring

This section describes optional steps for tailoring AutoOPERATOR for CICS to your site's requirements.

Overview

All steps described in this section are optional, and the full functionality of AutoOPERATOR for CICS is available whether or not they are performed.

The AutoOPERATOR for CICS optional tailoring steps are:

- Customizing Terminal Types Eligible for Broadcast Messages (page 91)
- Limiting CICS Transient Data Queues Eligible for Rule Processing (page 92)

If customized to do so, 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 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 AutoOPERATOR for CICS terminal table so that it includes the terminal types used at your site.

The CAOTTAB macro is used to generate the 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:

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, AutoOPERATOR for CICS intercepts any data written to any CICS extra partition transient data destination. If customized to do so, 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” 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 93.
4. Submit the newly created JCL; check the output to insure that all steps complete with a return code of 0.

Describing CAODTAB Macro Parameters

The following list describes input parameters for the CAODTAB macro:

Parameter	Possible values
-----------	-----------------

EXTRA	(YES NO,INCLUDE EXCLUDE)
--------------	--------------------------

INTRA	(YES NO,INCLUDE EXCLUDE)
--------------	--------------------------

QUEUES	[(name1),(name2),(namen)...]
---------------	------------------------------

The QUEUES= parameter is optional. Use the QUEUES= parameter when you want to explicitly include or exclude a specific queue(s) by queue name.

Examples

The following examples show how to use different combinations of the parameters on the CAODTAB macro invocation.

Example 1

To specify 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) then all extrapartition transient data messages are included.

Example 2

To specify only intrapartition transient data messages will be eligible for processing by the Rule Processor:

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) then all intrapartition transient data messages are included.

Example 3

To specify both intrapartition and extrapartition transient data messages will be eligible for processing by the Rule Processor:

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) then all extrapartition and intrapartition transient data messages are included.

Example 4

To specify no transient data messages from any queue will be eligible for processing by the Rule Processor:

EXTRA=(NO) , INTRA=(NO)

Example 5

To specify only intrapartition transient data messages will be eligible for processing by the Rule Processor:

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 6

To specify only intrapartition transient data messages will be eligible for processing by the Rule Processor:

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.

Chapter 12. Implementing MAINVIEW Products in IMS

This chapter describes how to install and implement BMC Software 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 of these products and you have not performed AutoCustomization. AutoCustomization tailors your products automatically.

You can refer to the manual customization steps in this chapter if you need help during AutoCustomization.

Note: If you have multiple IMSs, you may want to allocate a BBPARM data set that is unique to an IMS, as described in “Setting Up BBPARM Data Sets” on page 99 and “Modifying the IMS Control Region JCL” on page 103.

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 Common Customization Guide* 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 3 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 3. BBSAMP Jobs to Delete Old Modules

IMS Release	MVIMS 3.1 MVDBC 2.1 AO 3.1, AO 4.1	MVIMS 3.2 MVDBC 3.2 AO 4.1	MVIMS 3.3.mm MVDBC 3.3.mm AO 5.1, AO 6.1, AO 6.2
IMS 5.1	IDEL31\$5	IDEL32\$5	IDEL33\$5
IMS 6.1	n/a	IDEL32\$6	IDEL33\$6
IMS 7.1	n/a	n/a	IDEL33\$7

To delete the old modules:

1. From Table 3, select the delete job that corresponds to the BMC Software 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 OS/390.

Setting Up BBPARM Data Sets

The customization instructions in this chapter refer to the following data sets:

hilevel.ibbparm
hilevel.UBBPARM
hilevel.BBPARM

hilevel The high-level data set name qualifier used at your site

ibbparm 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 and IMFECPO0.

The *ibbparm* data set must be allocated; it is not created through AutoCustomization.

Note: If the only members that require customization for an IMS are IMFSYS00, IMFECPO0, 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 IMFECPO0 member, or both, include them in UBBPARM, and customize them to suit your needs (as explained below in UBBPARM).

UBBPARM A parameter data set that is tailored from the distributed BBPARM data set and is shared by all IMS systems

If you used AutoCustomization, you can use the UBBPARM data set created by AutoCustomization. If you did not use AutoCustomization, allocate UBBPARM, copy the distributed BBPARM data set to it, and tailor UBBPARM to suit your needs.

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:

imsidSYS (where *imsid* is the four-character IMS ID code)

You can also include a copied, renamed version of IMFECPO0 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:

imsidECP (where *imsid* is the four-character IMS ID code)

If you do not need to customize *other* UBBPARM members for an IMS system, you do not need to create and allocate an *ibbparm* member for that system.

BBPARM The target BBPARM data set distributed by BMC Software

Important

Throughout the MVAAO books, 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 normally referred to as BBPARM member IMFSYS00, and the Event Collector parameter is normally referred to as BBPARM member IMFEC00, even though the members may 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
1. 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:

SUBSYS. 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 BBIJNT00 member of the UBBPARM data set or an *ibbparm* data set.

MSGLVL1. Set the message level to MTO, WTO, BOTH, or NONE.

KEYWARN. 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.)

AOEINIT. This parameter is used with the DFSAOE00 initialization call. (See BBPARM member IMFSYS00 for more information.)

AOEEXIT and AOIEXIT. 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 104 for more information.

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.

Activating MAINVIEW AutoOPERATOR for IMS

To activate MAINVIEW AutoOPERATOR for IMS, you need to 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 use product keys to enable MAINVIEW AutoOPERATOR for IMS product authorization.

Product Keys

If you have licensed MAINVIEW AutoOPERATOR for IMS and you want to use a product key to activate the product, specify product key IAO in BBPARM member BBKEYS.

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:

```
//IMFPARM DD DSN=hi level . i bbparm  
// DD DSN=hi level . UBBPARM  
// DD DSN=hi level . 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:

- ICOPY8 for IMS 5.1
- ICOPY9 for IMS 6.1
- ICOPY1 for IMS 7.1

Then edit the JCL and run the job. This job copies the appropriate BMC Software modules from the BBLINK library to the site-authorized library, such as IMS RESLIB. You must rerun the JCL each time you apply BMC Software service.

Enabling AO Exit Routines

You can skip this section if you have a prior version of MAINVIEW AutoOPERATOR for IMS installed.

IMS gives control to AO exit routines to do initialization processing and to do message processing. BMC Software does not supply an AO exit routine to do initialization processing. If you have your own routine, specify its one- to eight-character load module name in the AOEINIT parameter in an *imsidSYS* member in the UBBPARM data set or an IMFSYS00 member in an *ibbparm* data set.

BMC Software 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 IBM's *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 (if any)
- automatically start up MVIMS 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

On entry to DFSAOE00 with AOE0FUNC=1, MVIMS loads and executes any user exit specified in AOEINIT=xxxxxxx 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.

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 *imsidSYS* member in UBBPARM or an IMFSYS00 member in an *ibbparm* data set.

For 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.

For 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: 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

For 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:

```
AOIEXIT=(USEMINE,RC)  
AOIEXIT=IGNOREME
```

Enhancement to the IMS DFSAOE00 Interface: 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: When you install your user exit with a BMC Software exit, you should consider the following information:

- If the BMC Software exit is executed before the user exit and the user exit's return code cancels the processing of additional message segments of a multisegment 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 timestamp sequence by MAINVIEW AutoOPERATOR.
- When the user exit is executed before the BMC Software 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 multisegment 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 Software 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 Software routines. If an abend occurs in one of these exits, only that routine is disabled; the other routine and IMS itself are not affected.

Stopping Message Queuing to the IMS Master Terminal

There are two ways to stop message queuing to the IMS master terminal:

- Physically turn off the master terminal (this requires BBI Logging).
AutoOPERATOR detects that the master terminal is not operable and will cancel any messages that are queued.
- Create an AutoOPERATOR IMS-initiated Rule to suppress messages from the master terminal. Refer to the section on Rules in the *MAINVIEW AutoOPERATOR Basic Automation Guide*.

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 V
NET, ID=nodename, INACT, FORCE

Chapter 13. Implementing AutoRULE

This chapter describes how to implement the optional AutoOPERATOR AutoCustomization step, “Run AutoRULE for AutoOPERATOR, Event Management Utility”.

Overview

AutoRULE is an optional advanced customization step for AutoOPERATOR that allows you to automatically generate AutoOPERATOR Rules from your existing Message Processing Facility (MPF) lists and AutoMate Rules.

You can also use the BMC Software-supplied KnowledgeBases to generate AutoOPERATOR Rules. The generated Rules perform automation such as suppressing messages from the console, replying to write-to-operator messages (WTORs), and issuing OS/390 commands.

Prerequisites

To use AutoRULE, you must first fill in the worksheet (refer to Appendix H, “AutoRULE Worksheet” on page 227). The worksheet should be used in conjunction with the AutoCustomization steps because you will need the data from the filled-in worksheet to proceed through the AutoCustomization steps.

Completing the AutoRULE AutoCustomization Step

The following describes the panels you must complete while executing the AutoCustomization step, “Run AutoRULE for AutoOPERATOR, Event Management Utility”. You will need the information from the filled-in worksheet to respond to the AutoCustomization steps.

1. Select the step titled “Run AutoRULE for AutoOPERATOR, Event Management Utility” to implement AutoRULE from AutoCustomization.

Figure 14 is displayed.

```
AUTOOPERATOR ----- AutoRULE Event Management ----- CUSTOMIZATION
COMMAND ==>

AutoRULE is an optional AutoOPERATOR event management utility. It allows
you to create and maintain AutoOPERATOR Rulesets. The Rulesets are built
using the supplied KnowledgeBases, input from MVS MPF lists, and AutoMate
Rules. AutoRULE can be run multiple times, but should be run at least once
each time you receive a KnowledgeBase.

All AutoRULE panels have context sensitive help and each step is optional.

Please ensure you have reviewed the AutoRULE documentation and fill in the
supplied worksheets prior to executing this step.

Would you like to run AutoRULE? at this time ==> YES (Yes/No)

  If you reply NO, this step will be bypassed.

  If you reply YES, you will invoke AutoRULE Event Management

Press ENTER to continue, HELP for more information, or END to exit.
```

Figure 14. AutoRULE Event Management: Panel 1

2. To proceed, enter YES at .

In Figure 15, you are prompted to select the conversion process you want to complete.

```
AUTOOPERATOR ----- AutoRULE Event Management ----- CUSTOMIZATION
OPTION ==>

Please complete the steps applicable to your site      User ID : BAOMXY1
in the order shown below. Each step may be rerun    Time   : 15:20
as often as needed before moving on to the next.    Date   : 96/04/25

                                                    Last Run on
                                                    -----
M MPF      Convert MPF List                       3 Apr 1996 13:07:34
A AUTOMATE Convert AutoMate Rules                 25 Apr 1996 13:03:09
C COMPARE  Compare & Delete Duplicates            10 Apr 1996 11:04:29
R RULES    Generate AutoOPERATOR Rules            25 Apr 1996 12:49:55
O OUTPUT   Browse Output

Press END to return, HELP for more information or select an OPTION
```

Figure 15. AutoRULE Event Management: Panel 2

3. To convert MPF automation to AutoOPERATOR Rules, choose the step Convert MPF List (). You will need the information from Section 1 of the worksheet to complete this step.

Or

4. To convert AutoMate automation to AutoOPERATOR Rules, choose the step Convert AutoMate Rules (). You will need the information from Section 2 of the worksheet to complete this step.

You must choose and complete one of these steps before you complete any of the other steps presented on this panel.

5. Once the conversion you chose is completed, select the step **Compare and Delete Duplicates** (see in Figure 15).

Figure 16 is displayed.

```
AUTOOPERATOR ----- AutoRULE Event Management ----- CUSTOMIZATION
COMMAND ==>>

Line commands: / to select a member to process, E to edit, or B to browse.

Note that highlighted members must be viewed before being selected.

MVS Console Suppression: (Ind.)  MVS Console Actions: (Ind.)
  MVS Conservative List          MVS WTOR Replies
  MVS Aggressive List           MVS Commands
  CICS Messages                 CICS WTOR Replies
MVS Console Alerts:             Non-MVS Alerts:
  CICS Alerts                   CICS TDQ Alerts
  MVS Alerts                   BBI Journal Alerts
  Production Job Alerts         IMS AOI Exit Alerts
  Tape Mounts & Errors         DB2 Alerts
MVS Command Handling:          Converted User Automation (created)
  MVS Command Rejection        MPF List 3 Apr 1996
  User Commands                AutoMate Rules 25 Apr 1996

Other member name (optional):

Press END to return, HELP for more information or enter GO to proceed.
```

Figure 16. AutoRULE Event Management: Compare Option

Use this panel to select AutoRULE KnowledgeBase members to be included in your newly generated Rule Set. Each of the members is for a specific type of message and action type.

Before selecting members, you can browse or edit a member. Use the B or E line command to do this. The (Ind.) column indicates which members have been browsed or edited.

When you edit a member, you can change the flags to create a different type of Rule. For example, you might want to delete some of the automatic replies from the OS/390 Commands & Replies member. For a list of flags and their meanings, refer to “Describing AutoRULE Flags” on page 230.

When the Compare step has completed and you have reviewed the error messages that are displayed, Figure 17 is displayed.

```

AUTOOPERATOR ----- AutoRULE Event Management ----- CUSTOMIZATION
COMMAND ==>

          Compare & sort has successfully completed.

                What to do now:

In any cases where there were conflicting Rules for an event, your
converted MPF or AutoMate Rule was placed ahead of the distributed
AutoRULE Rule.  If you have changed your mind after reviewing the
results of the Compare & Sort step (in member ARLISTC ), you can
delete your Rule and rerun this step.

If you are happy with the results, go on to the next step:

Next, run Rules.  This will generate AutoOPERATOR Rules from your
converted Rules and the distributed KnowledgeBase Rules.

Compare & Sort has also created a sorted list of messages that will
be handled by the new Rules (in the member ARSORTED ).  You can
print this member for future reference.

Press END or ENTER to continue

```

Figure 17. AutoRULE Event Management: Compare Option Completed

6. Follow the instructions listed on this panel.

If the compare and sort step completed successfully, proceed to the step Generate AutoOPERATOR Rules (in Figure 15).

The Generate AutoOPERATOR Rules step generates AutoOPERATOR Rules in Rule Sets that are named RULRUxxx, where xxx is a user-specified three-character suffix.

7. Specify the suffixes for the Rule Sets. Enter the information from Section 3 of the worksheet on this panel.

```

AUTOOPERATOR ----- AutoRULE Event Management ----- CUSTOMIZATION
COMMAND ==>

Ruleset Names:                (3-character suffix ID)

MSGs starting with A - G      MSGs starting with H - L
MSGs starting with M - S      MSGs starting with T - Z
Journal Events                 CICS TDQ Events
IMS Events                     Command Events

Rules Author                   (For Documentation Fields)
Display on Console             YES or NO
Catch-all Rules              ENABLED DISABLED SAVE or NONE

STCs to Exclude

Userids to Allow

Press Enter to continue, HELP for more information, or END to return.

```

Figure 18. AutoRULE Event Management: Naming Rule Sets

8. Press ENTER; Figure 19 is displayed. Enter the information from Section 4 of the worksheet on this panel.

```

AUTOOPERATOR ----- AutoRULE Event Management ----- CUSTOMIZATION
COMMAND ==>

Define a production job, using the following Rule fields.
Job name . . . . . Userid . . . . . RACF Group
Job class. . . . . Accounting Info . . . . .

Variable-name          Op      Variable-value          AND/OR

Press ENTER to continue, END or CANCEL to exit without creating Rules.
HELP for more information.

```

Figure 19. AutoRULE Event Management: Creating a Production Job

This panel specifies the parameters that tell AutoRULE how to recognize a production job, and this information is used for creating Rules for production job abends.

The bottom portion of the panel allows you to specify variable dependencies that function exactly as the Variable Dependencies panel does during AutoOPERATOR Rules creation (refer to the *MAINVIEW AutoOPERATOR Basic Automation Guide* for information about creating Rules).

9. Press ENTER; Figure 20 is displayed. Enter the information from Section 5 of the worksheet on this panel.

```

AUTOOPERATOR ----- AutoRULE Event Management ----- CUSTOMIZATION
COMMAND ==>

Specify User IDs and Alert Queues for problem notification.

TSO User IDs for Notification          Operator Alert Queue Names
MVS Problems          KMZ1          MVS Alerts          MVS
CICS Problems        BAOKMZ1       CICS Alerts         CICS
IMS Problems         BAOKMZ1       IMS Alerts          IMS
DB2 Problems         BAOKMZ1       DB2 Alerts          DB2
Network Problems    BAOKMZ1       Network Alerts      NET
DASD Problems       BAOKMZ1       Tape Alerts         TAPE
Unauthorized Cnds   BAOKMZ1       Batch Alerts        BATCH
Automation Problems CMF6          All Other Alerts    MAIN

OptionalParms
Alerts Target Name          Leave blank for no Alert routing
Default Beeper Name          Elan only
Default Beeper Info
(continued)

Press ENTER to continue, END or CANCEL to exit without creating Rules.
HELP for more information.

```

Figure 20. AutoRULE Event Management: Specifying ALERT Queues

This panel contains the parameters that tell AutoRULE how to direct ALERTs after they are created. The user IDs allow you to specify a TSO user ID to be notified when an ALERT is created. To specify more than one TSO user ID, separate the IDs with spaces.

10. Press ENTER; Rules are generated and Figure 21 is displayed.

```
AUTOOPERATOR ----- AutoRULE Event Management ----- CUSTOMIZATION
COMMAND =====>

                          AutoRULE Rules Generation Successfully Completed

                          What to do now:

If you didn't get the results you expected, you can go back and fix
the errors and rerun this or any other step. The listing you just
viewed is in member ARLIST in the UBBPARAM dataset.

UBBPARAM member MPFNEW is a replacement version of MPF list. After
you implement the new Rules, copy it to SYS1.PARMLIB with the same
name as the current MPF list (MPFLSTxx). Then issue the MVS MPF
list reset command.

Use the OUTPUT option to aid in completing your Event Management
conversion process.

Press END or ENTER to continue
```

Figure 21. AutoRULE Event Management: Successful Completion Panel

The new UBBPARAM members that have been created by AutoRULE are:

- ARLIST, which contains the results of the AutoMate conversion

If any AutoMate Rules were not converted by AutoRULE, you will have to convert them by manually creating AutoOPERATOR Rules for them. Refer to the *MAINVIEW AutoOPERATOR Basic Automation Guide* for information about creating AutoOPERATOR Rules.
- MPFNEW, which contains a new listing that you can use to replace the existing MPFLSTxx member in SYS1.PARMLIB

To use the new member, make a copy of the existing MPFLSTxx member. Next, copy MPFNEW to SYS1.PARMLIB and rename it to the previous MPFLSTxx member name. Last, issue the MVS MPF list reset command.
- ARSORTED, which contains a list of the events from which the Rules were generated

You may want to print the contents of this member to have as a reference document of the events that now have Rules.
- Various new Rule Sets named RULRUxxx, where xxx is a user-specified three-character suffix

To use these new Rule Sets, enter the new Rule Set names in BBPARAM member AAOPRMxx on the RULESET parameter.

The AutoCustomization step for AutoRULE is complete.

Chapter 14. Implementing AutoOPERATOR Access NV

This chapter describes how to install the MAINVIEW AutoOPERATOR Access NV option into NetView and how to define the default NetView system to AutoOPERATOR.

Overview

To implement the AutoOPERATOR Access NV option for NetView and define the default NetView system to 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.

Task description	Page number
____ 1. Modify NetView JCL	116
____ 2. Create member NAIOPT00 in the NetView DSIPARM data set	116
____ 3. Make backup copies of DSIPARM members DSICMD, DSIDMN, and DSIOPF	N/A ^a
____ 4. Modify DSIPARM member DSICMD	117
____ 5. Modify DSIPARM member DSIDMN	117
____ 6. Copy members NAIINIT and NAISVAR into the NetView DSICLD data set	117
____ 7. Modify member DSIOPF in DSIPARM	117
____ 8. Verify operator profiles	118
____ 9. Review NetView security for Access NV operator IDs	118
____ 10. Review use of DSIEX02A exit	118
____ 11. Review use of DSIEX14 exit	119
____ 12. Modify member BBISSP00 in the BBPARM data set	120
____ 13. Restart NetView to implement JCL and parameter changes	N/A ^a
____ 14. Restart the BBI-SS PAS to implement parameter changes	N/A ^a

^aThese steps do not have individual descriptions in this chapter.

Note: If NetView is started before JES (SUB=MSTR), and you want to obtain dumps from the NetView product address space, you must complete the following steps:

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 NetView address space JCL, pointing to the dump data sets 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 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 AutoOPERATOR programs and parameters available in the NetView address space, modify the NetView JCL:

1. Add the AutoOPERATOR BBLINK library to the NetView STEPLIB concatenation.
2. Add the following new DD statements to the NetView JCL:

```
//BBI PARM DD ... DSN=hi l vl . BBPARM, DISP=SHR  
//BBI LOAD DD ... DSN=hi l vl . BBLINK, DISP=SHR
```

Creating DSIPARM Member NAIOPT00

Create member NAIOPT00 in the NetView DSIPARM data set to define:

- The BBI-SS PAS with which NetView will communicate
 - OSTs used to execute IMFEXEC NetView commands
1. Copy AutoOPERATOR BBSAMP member NAIOPT00 to the NetView DSIPARM data set.
 2. Edit NAIOPT00, changing the SSID parameter to specify the subsystem ID of the BBI-SS PAS that you want to process NAIEXEC commands issued from NetView EXECs.

Note: NAIEXEC commands can be sent to a different BBI-SS PAS using the TARGET keyword on the command. However, all NAIEXEC commands are initially routed to the specified BBI-SS PAS and then, if necessary, routed to another BBI-SS PAS using SS PAS-to-SS PAS communication.

3. Add 19 (or less) OPERxx statements to define the operator station tasks (OSTs) used to execute the IMFEXEC NETVIEW commands issued by AutoOPERATOR EXECs.

You must specify at least one OPERxx statement; multiple statements do not need to be sequential.

The operator IDs specified should be used solely for Access NV. A user should not log on to NetView with any of these IDs.

Modifying DSIPARM Member DSICMD

Modify the DSIPARM member DSICMD:

1. Append the contents of BBSAMP member NAICMDS to the end of your existing DSICMD member

This adds definitions for the two NetView command processors, NAIEXEC and NAISTUB, used by Access NV.

Access NV uses two command processors that must be defined to NetView in the DSICMD member in the DSIPARM data set.

Modifying DSIPARM Member DSIDMN

Some Access NV functions run as an optional subtask (OPT) under NetView. Modify the DSIPARM member DSIDMN to define this OPT:

1. Insert the contents of BBSAMP member NAITASK before the END statement of your existing DSIDMN member.

This adds a definition for the NetView optional subtask (OPT), NATASK, 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 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 may 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:

1. Copy BBSAMP members NAIINITX, NAIINIT, and NAISVAR into the DSICLD data set.

Modifying DSIPARM Member DSIOPF

The DSIOPF member in DSIPARM allows you to associate passwords and profiles with operator IDs.

Modify member DSIOPF 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. If necessary, add operator IDs for TS users who will use Access NV.

The NetView operator ID for a TS user is the same as the BBI-TS ID. The profile specified for these IDs must be similar to the example profile in BBSAMP member NAIPROF. See the following section for more information.

Verifying Operator Profiles

All Access NV controlled operator station tasks (OSTs) **must** execute an initialization CLIST which contains the NAISTUB command. This applies to TS users as well as the OPERxx IDs used for IMFEXEC NETVIEW commands.

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 NAILOPT00
 - 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 DSICLDA data set to include the statements from either NAINITX or NAINIT and use:
 - NAIINITX for OPERxx IDs that were specified in NAILOPT00
 - NAIINIT for TS operator IDs

Reviewing NetView Security for Access NV Operator IDs

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 an AutoOPERATOR EXEC (IMFEXEC NETVIEW), the NetView operator ID is chosen using a least-recently used algorithm from the pool of OPERxx IDs defined in DSIPARM member NAILOPT00. Standard NetView security applies to the operator ID.

All OPERxx IDs must have the maximum amount of authority required by any AutoOPERATOR EXEC since a pool of operator IDs is used for processing IMFEXEC NETVIEW commands. In addition, these operator IDs must have authority to issue NetView START and STOP commands.

Reviewing the Use of NetView Exit DSIEX02A

AutoOPERATOR Access NV uses the NetView exit DSIEX02A. This exit processes line mode terminal and console output and allows messages to be modified.

If you do not currently have a DSIEX02A exit, no action is required to install the AutoOPERATOR exit. If you already use this exit, do the following steps to make sure that

your existing DSIEX02A exit and the 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=the library where your renamed exit resides
//SYSLMOD     DD  DSN=A0BBLINK
CHANGE  DSIEX02A(DSIEX02B)
INCLUDE  OLDLIB(DSIEX02B)
INCLUDE  SYSLMOD(DSIEX02A)
ORDER  DSIEX02A
ENTRY  DSIEX02A
NAME  DSIEX02A(R)
```

AutoOPERATOR Access NV first passes control to DSIEX02B. If DSIEX02B has modified the incoming message, AutoOPERATOR Access NV processes the modified message. If the exit indicates that the message should be discarded, AutoOPERATOR Access NV returns control to NetView with a return code 4.

Reviewing the Use of NetView Exit DSIEX14

AutoOPERATOR Access NV uses the NetView exit DSIEX14. This exit intercepts LOGOFF requests and performs clean-up processing for Access NV controlled OSTs. LOGOFF requests for OSTs that are not controlled by Access NV are ignored.

If you already use this exit, follow the steps below to make sure that your existing DSIEX14 exit and the AutoOPERATOR Access NV DSIEX14 exit can coexist:

1. Rename the existing DSIEX14 load module to DSIEX14B.
2. Link the two exits by using the following linkage editor statements:

```
//OLDLIB      DD  DSN=the library where your renamed exit resides
//SYSLMOD     DD  DSN=A0BBLINK
CHANGE  DSIEX14(DSIEX14B)
INCLUDE  OLDLIB(DSIEX14B)
INCLUDE  SYSLMOD(DSIEX14)
ORDER  DSIEX14
ENTRY  DSIEX14
NAME  DSIEX14(R)
```

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.

Modifying BBPARM Member BBISSP00

Add a NETVIEW= specification to BBPARM member BBISSP00. This 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 work station 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 jobname on the IMFEXEC NETVIEW command.

Chapter 15. Implementing AutoOPERATOR ALERT Interface to PATROL Enterprise Manager

This chapter describes how to customize AutoOPERATOR to send ALERTs to the PATROL Enterprise Manager (PATROL EM) outboard processor.

For information about receiving AutoOPERATOR ALERTs on the PATROL EM workstation, refer to the PATROL EM user documentation.

Overview

There are two ways to implement sending ALERTs from AutoOPERATOR to PATROL EM:

- By implementing a TCP/IP-based interface between AutoOPERATOR and PATROL EM (see Figure 22)
- By establishing a VTAM connection to PATROL EM using a unique ACB and an I/O Concepts X-Direct protocol converter (see Figure 23 on page 122)

The following figures show the two different configurations when AutoOPERATOR is enabled to send ALERTs to PATROL EM

Figure 22. AutoOPERATOR and PATROL EM Connection Using TCP/IP

Note in this configuration that there is no distance limit between the locations of the PATROL EM systems and the AutoOPERATOR systems. In Figure 23 on page 122, there are significant physical distance limitations.

Figure 23. AutoOPERATOR and PATROL EM Connections Through I/O Concepts X-Direct Protocol Converters

Note in this configuration that there is a limit of 1 km between AutoOPERATOR and the I/O Concepts protocol converter, and there is a 10 m limit between PATROL EM and AutoOPERATOR. Therefore, AutoOPERATOR cannot send ALERTs if it is located more than 1 km away, and PATROL EM cannot send ALERTs if it is located more than 10 m away.

For more information about these two ways, refer to:

- “Establishing a TCP/IP Connection” on page 123.
- Appendix K, “Establishing a VTAM Connection for AutoOPERATOR to PATROL EM Communication” on page 245.

This feature requires that you install the latest version of the ConnectAAO Compatibility Patch.

Establishing a TCP/IP Connection

The following sections describe how to enable AutoOPERATOR to share ALERTs with PATROL EM via a TCP/IP connection.

What You Need

To establish a TCP/IP connection between AutoOPERATOR and PATROL EM, you must have:

- At least one BBI-SS PAS where AutoOPERATOR 6.1 or later is running
- At least one PATROL EM system that has been configured to accept ALERTs from AutoOPERATOR 6.1 or later

For information about how to do this, refer to the PATROL EM documentation.

- MVS/ESA TCP/IP 3.2 or later or Interlink TCPAccess 5.2 installed in the MVS system

You must also perform the following steps:

1. In BBPARM member BBISSP00, specify

```
GTS=xx
```

where `xx` is the suffix of your BBTTCp`xx` member (see the next step).

2. (Optional for MAINVIEW AutoOPERATOR version 6.1 and later, and **required** for Command MQ for S/390, Version 3.), create BBPARM member BBTTCp00 (refer to “BBPARM Member BBTTCp00” in the *MAINVIEW Common Customization Guide* for more information about the BBTTCp00 member.)

If you choose not to complete this step, specify `GTS=00` in BBPARM member BBISSP00 (see Step 1).

3. Create BBPARM member BBTTCp`xx` in your copy of the BBPARM library, where `xx` is 00 or the suffix that you specify with the `GTS=` parameter in BBPARM member BBISSP00. A sample member can be found in “hi-level.BBSAMP(BBTTCp`xx`)”.

BBPARM member BBTTCp00 defines your TCP/IP environment and has the following statements:

```
TCPNAME=TCP/IP started task
```

```
STACK=IBM32 | ILINK52
```

where

`TCPNAME` is set to the started task name of your TCP/IP job

`STACK` is set based on the version of TCP/IP that you are using. If you are running TCP/IP version 3.2 or later, use `IBM32`. If you are using Interlink TCPAccess 5.2, use `ILINK52`.

BMC Software recommends that you not specify any parameters in member BBTTCp`xx`. The parameters are designed mostly for testing or support purposes. When parameters are not specified, the BBI-SS PAS chooses the most current TCP/IP stack that is found active on the system.

When these steps are completed, you must

- Create a BBPARM member AAOGME00

This can be done through the AutoCustomization step, “Customize AutoOPERATOR ALERTs Publishing” or through the manual customization step which is documented in “Defining TCP/IP Addresses in AutoOPERATOR” on page 124.

The AAOGME00 member allows AutoOPERATOR to use the General Messages Exchange (GME) facility to share AutoOPERATOR ALERTs with PATROL EM. The local GME node requires network addresses and port numbers to connect to remote GME nodes.

- Define TCP/IP addresses in BBPARM member AAOGME00

Once the AAOGME00 member is created, you must specify the IP address of each PATROL EM system in the TGTIP= parameter for communications to be enabled. For more information, refer to “Defining TCP/IP Addresses in AutoOPERATOR”.

- Define the port number on which the PATROL EM server is listening

Once the AAOGME00 member is created, you can (optionally) set the TGTPORT= parameter. For more information, refer to “Defining TCP/IP Addresses in AutoOPERATOR”.

You can set additional parameters in AAOGME00 with the Dynamic Parameter Manager application. For information about additional parameters, refer to “Modifying the TCP/IP Connection with BBPARM Member AAOGME00” on page 128 or Appendix E, “BBPARM Data Set Members for AutoOPERATOR” on page 189.

- Reset the BBPARM member AAOGME00.

Defining TCP/IP Addresses in AutoOPERATOR

Note: If you completed the AutoCustomization step “Customize AutoOPERATOR Alerts Publishing”, you do not have to complete the steps in this section.

To define TCP/IP addresses:

1. Create a member in your UBBPARM data set named AAOGME_{xx} where _{xx} is a two-character suffix associated with the BBI-SS PAS you are enabling to share ALERTs with PATROL EM.

2. Specify (mandatory):

TGTIP=IP_ADDRESS

where IP_ADDRESS is the name of the location of a server to which GME will connect.

GME allows IP addresses to be expressed in:

- IP v4 (dotted decimal)
- IP v6
- Internet host domain name; for example: `www.remote_host_domain_name.com`

If a host domain name is used, it is your responsibility to define that name in a domain name server (DNS).

3. Specify (optional):

TGTPORT=xxxxx

where xxxxx is a one- to five-digit decimal number specifying the port number on which the PATROL EM server is listening. The GME facility attempts to connect to the remote GME node using this port number.

The default port number is 7051.

Specifying More Than One TCP/IP Address

Note that you may specify as many TGTPIP and TGTPORT parameters in BBPARM member AAOGME00 as you need. Any parameters set prior to the first TGTPIP parameter are attributed to the local GME node.

The TGTPIP parameter denotes definitions for remote GME nodes and the TGTPORT parameter should appear in column 1. Each subsequent TGTPIP parameter signifies settings for a different remote GME node. For example:

```
TGTPIP=137. 72. 4. 15, TGTPORT=12345  
TGTPIP=172. 18. 14. 255, TGTPORT=67890
```

Additional parameters can be set for each TGTPIP address, each separated by a comma. Settings can continue onto additional lines; for example:

```
TGTPIP=137. 72. 4. 15, TGTPORT=12345, TGTNAME=remote_GME_node_name_here,  
TGTHB=999
```

For information about additional parameters, refer to “BBPARM Member AAOGME00 Parameters (AutoOPERATOR Only)” on page 199.

Resetting BBPARM Member AAOGME00

After the TGTPIP and TGTPORT (optional) parameters are set in the BBPARM member AAOGME00, you must activate the new GME connection by issuing the following BBI control command:

- . RESET PARM AAOGMEnn, RESYNC

Controlling the AutoOPERATOR and PATROL EM Interface

Following is a list of BBI control commands that you can use to control the AutoOPERATOR and PATROL EM interface.

Table 4. BBI Control Commands for AutoOPERATOR and PATROL EM Interface for TCP/IP Connections

Command	Keyword	Parameter	Description
DISPLAY D	GME G	CONN C PUBLISH P RECEIVE R TRACE T TRACE,nnnn T,nnnn no parm	<p>Provides a summary status of the local GME node.</p> <p>CONN Displays status of GME connections. A specific connection is displayed by specifying the GME NAME.</p> <p>PUBLISH Displays details of PUBLISHED GME subjects.</p> <p>RECEIVE Displays details of the GME applications that are registered to receive messages.</p> <p>TRACE Displays a summary of the trace records by class.</p> <p>TRACE,nnnn Displays up to 10 trace records starting with the specified trace record sequence number.</p> <p>no parm Displays status of the local GME node.</p>
RESET E	GME G	nodename	<p>Stops and starts communications with a GME server node using the parameters specified from the active BBPARM member AAOGMExx.</p> <p>Possible specifications for nodename are:</p> <p>TGTNAME Is the name of the GME server node whose connection is to be stopped and started. Typically, this will be the workstation ID of the system where the remote GME server node is running.</p> <p>IP Address:Port Is a combination of the IP address and the port number connected by a colon (:).</p> <p>Hostname:Port Is a combination of the hostname and the port number connected by a colon (:).</p>

Table 4. BBI Control Commands for AutoOPERATOR and PATROL EM Interface for TCP/IP Connections (Continued)

Command	Keyword	Parameter	Description
RESET E	PARM P	AAOGME _{xx} RECYCLE AAOGME _{xx} RESYNC	Reinitializes the parameters specified by AAOGME00. Possible options are to specify: RECYCLE Stops the GME environment, disabling all connections, and restarts the GME environment, re-establishing all connections. RESYNC Stops any active GME connections that might exist (but have been removed from the active BBPARM member AAOGME _{xx}) and starts any new connections that may have been added to AAOGME _{xx} since the recycle or startup of AutoOPERATOR.
START S	GME G	nodename	Starts communications with a GME server node using connection parameters read from the active BBPARM member AAOGME00. For a description of possible values for nodename, refer to the description for RESET GME nodename in this table.
STOP P	GME G	nodename	Stops communication with a GME server node. For a description of possible values for nodename, refer to the description for RESET GME nodename in this table.

For a description of all the available BBI control commands, refer to the *MAINVIEW Administration Guide* where the complete list of BBI control commands is documented.

Modifying the TCP/IP Connection with BBPARM Member AAOGME00

Use BBPARM member AAOGME to customize how the GME facility operates. BBPARM member AAOGME00 is used by the AutoOPERATOR General Messages Exchange (GME) to identify which GME nodes receive messages from and send messages to AutoOPERATOR, define the local GME node, and control which instrumentation events are traced.

Refer to “BBPARM Member AAOGME00 Parameters (AutoOPERATOR Only)” on page 199 where the parameters are described.

Also refer to the “BBPARM Member BBTTCP00 Parameters” in the *MAINVIEW Common Customization Guide* for more information about the BBTTCP00 member, which defines your TCP/IP environment.

How AutoOPERATOR and PATROL EM Share ALERTs

The following describes how:

- AutoOPERATOR and PATROL EM synchronize their ALERT displays
- ALERTs and ALERT queues are created and deleted between the two products
- PATROL EM issues commands via AutoOPERATOR

once the connection between AutoOPERATOR and PATROL EM is completed.

ALERTs Synchronization

AutoOPERATOR synchronizes its set of ALERTs with PATROL EM’s set of ALERTs when one of the following events occurs:

1. AutoOPERATOR starts and PATROL EM is already active and the initial connection is made to PATROL EM
2. AutoOPERATOR starts but PATROL EM is not already active. Synchronization occurs when AutoOPERATOR detects that PATROL EM has started.
3. An inactive connection between AutoOPERATOR and PATROL EM becomes active.

The process of synchronizing is as follows:

1. AutoOPERATOR initializes.
2. AutoOPERATOR sends ALERTs destined for PATROL EM.
3. PATROL EM deletes any ALERTs which were deleted in AutoOPERATOR while it was not connected to PATROL EM.
4. PATROL EM deletes all ALERTs in AutoOPERATOR that PATROL EM deleted when it was not connected with AutoOPERATOR.

Creating and Deleting ALERT and ALERT Queues

The following table describes what happens when AutoOPERATOR creates or deletes an ALERT destined for PATROL EM and what happens on the PATROL EM side.

When...	Then...
An AutoOPERATOR Rule (or EXEC) generates a new ALERT destined for PATROL EM	PATROL EM creates the ALERT in the specified queue.
An AutoOPERATOR Rule (or EXEC) generates an ALERT to replace an existing ALERT on PATROL EM	PATROL EM deletes all existing ALERT(s) that match the queue regardless of which GME node sent them. PATROL EM creates the ALERT on the specified queue.
An AutoOPERATOR Rule (or EXEC) deletes an ALERT or ALERT queue that resides on PATROL EM	PATROL EM deletes the specified ALERT or deletes all ALERTs it received from AutoOPERATOR's GME node that belong to the specified queue.
PATROL EM deletes an ALERT that resides on AutoOPERATOR	AutoOPERATOR deletes the specified ALERT.

Chapter 16. Implementing AutoOPERATOR Advanced JES3 Support

Unlike JES2, JES3 manages its own message traffic and command execution internally and may use dedicated consoles to interact with the operator.

Depending on the JES3 release, these messages and commands might never become visible to the subsystem interface which AutoOPERATOR uses to drive the Rule Processor. Note that no additional customization is required to process OS/390 messages and commands. Therefore, some additional customization steps are necessary in order to fully manage a JES3 environment and perform automation tasks with Rules on JES3 messages and commands.

JES3 releases prior to 2.2 must be handled differently than later versions. The following sections explain the necessary steps to implement full JES3 support.

Using JES3 2.2 and Higher

JES3 releases 2.2 and higher use the standard OS/390 interface. However in most cases, JES3 will be the first subsystem to process messages and commands. For AutoOPERATOR to correctly process JES3 messages and commands, AutoOPERATOR must process the messages and commands before JES3 does.

To accomplish this, you must use the BBPARM member BBISSP00 parameter JESFLTR=. When this parameter is used, AutoOPERATOR installs a JES filter that allows AutoOPERATOR to see message or command traffic before JES does.

Possible values for the JESFLTR= parameter are:

```
JESFLTR=WT0|CMD
```

or

```
JESFLTR=(CMD, WT0)
```

The following describes under what circumstances you can use these specifications.

Intercepting Messages

Use JESFLTR=WT0 to suppress messages from a JES3 console. This specification replaces the use of the IATUX31 exit, as described in “Intercepting Messages” on page 133.

JES3 messages are input to the Rules Processor with an event type of MSG before they are seen by JES.

Intercepting Commands

Use `JESFLTR=CMD` to suppress messages from both JES2 and JES3 consoles. This specification affects:

- All JES2 commands
- Only JES3 commands that are entered at an OS/390 console

The JES3 command exit IATUX18 is required to suppress JES3 commands entered from a JES3 console.

JES3 commands are input to the Rules Processor with an event type of `CMD`.

Implementing the JES Filter

There are two ways to implement the JES filter:

- Through the `JESFLTR` parameter in `BBPARM` member `BBISSP00`; for example, specify:

```
JESFLTR=WTO
```

to suppress messages from a JES3 console.

To suppress all JES2 commands and JES3 commands entered from an OS/390 console, specify:

```
JESFLTR=CMD
```

To suppress both commands and messages, specify:

```
JESFLTR=(CMD, WTO)
```

The filter is installed after you have made these specifications and the BBI-SS PAS is cold started.

- Through the `BBI` command `.SET`

Use this command to establish the filter if the BBI-SS PAS was started before JES or to reestablish the filter (if it was removed during the life of a BBI-SS PAS) by specifying:

```
. SET JESFLTR=WTO
```

or

```
. SET JESFLTR=CMD
```

or

```
. SET JESFLTR=(CMD, WTO)
```

Issuing these commands will install the filter.

Removing the filter: To remove the filter at any time, regardless of whether the BBI-SS PAS is active or not, use the command `UNHOOK`. Enter the `UNHOOK` command as an OS/390 command and prefix it with the BBI-SS PAS command character prefix as defined by the `CMDCHAR` parameter in `BBPARM` member `BBISSP00`. The default command character is a period (`.`).

Other implementation considerations: You should not restart JES while the filter is installed. If JES is restarted while the filter is installed, one of two things might happen:

- The JES restart will fail
- AutoOPERATOR might not be able to issue any JES messages or commands

If either of these things happen:

1. Issue the UNHOOK command from an OS/390 console.
2. Restart JES.
3. Reestablish the filter by either method described in “Implementing the JES Filter”.

Issuing JES3 Commands from AutoOPERATOR

To correctly issue JES3 commands, the JES3CMD= parameter in BBPARM member BBISSP00 should be set to JES3CMD=N0.

Full command response support is available.

Using JES3 Releases Prior to 2.2

Releases of JES3 prior to 2.2 do not broadcast any information to the subsystem interface which AutoOPERATOR uses to drive the Rule Processor.

AutoOPERATOR provides a JES3 message exit (IATUX31) and a JES3 command exit (IATUX18) which are used to interface with the Rule Processor. Refer to BBSAMP members IATUX31 and IATUX18 for instructions on how to implement these required JES3 exits. By installing these exits, JES3 messages and commands can be seen by AutoOPERATOR and automation (triggered by these messages and commands) with Rules will be possible.

Intercepting Messages

When the IATUX31 exit is installed, JES3 messages are input to the Rule Processor with an event type of JES3. OS/390 messages will be input twice to the Rule Processor: once from the Master Broadcast function (subsystem interface) with an event type of MSG and again from the IATUX31 exit with an event type of JES3.

Any Rule Processor actions against event type JES3 messages affects JES3 devices only. Any Rule Processor actions against event type MSG affects MCS consoles or the SYSLOG.

Intercepting Commands

When the IATUX18 exit is installed, JES3 commands are input to the Rules Processor with an event type of CMD.

Issuing JES3 Commands from AutoOPERATOR

Prior to JES3 release 2.2, SVC 34 could not be used to issue JES3 commands. The commands must be issued using the subsystem interface. To do this:

1. Specify `JES3CMD=YES` in BBPARM member `BBISSP00` to indicate that the subsystem interface should be used instead of SVC 34.
2. Specify the correct JES3 command character for your system using the `JESCHAR=` parameter in `BBISSP00`.

When either `IMFEXEC CMD` or a TS user issues an OS/390 command with the first character equal to the command character specified by the `JESCHAR=` parameter, the command is issued using the subsystem interface instead of SVC 34. If either the `JES3CMD=` or `JESCHAR=` parameters is set incorrectly, JES3 commands can not be issued using `IMFEXEC CMD` or from the TS.

Command response is not available for these JES3 releases.

Chapter 17. Implementing COMMON STORAGE MONITOR

This chapter discusses implementing COMMON STORAGE MONITOR with AutoOPERATOR.

Overview

Common storage is storage that can be addressed by all address spaces. It comprises both a common storage area (CSA) and a system queue area (SQA). CSA is multiple-key storage that may or may not be “fixed” (backed by real memory). SQA is key-zero storage that is always fixed.

Common storage is a limited resource, especially storage that is fixed and storage below the 16Mb line. When you allocate common storage you are responsible for freeing it when it is no longer needed or when abnormal termination occurs. Neglecting this step can cause shortages of available space. Such shortages often result in unscheduled system outages.

The system does not keep track of what address spaces have allocated which common storage locations. COMMON STORAGE MONITOR provides the link between common storage allocation and the address spaces allocating it. COMMON STORAGE MONITOR tracks requests for CSA and SQA storage and maintains a table of allocations by jobname and address space ID. Requests are time-stamped and accumulated by subpool, storage key, and location (above or below the 16Mb line).

COMMON STORAGE MONITOR can perform long-term or short-term monitoring of common storage. Use long-term monitoring to diagnose problems with slowly increasing allocations of common storage, or to ascertain which address spaces are allocating excessive amounts of common storage. Use short-term monitoring to diagnose problems with sudden increases in common storage allocations, or to monitor a single address space or a few address spaces that you suspect are overusing common storage.

OS/390 does not keep CSA storage allocation information in its own control blocks. COMMON STORAGE MONITOR builds and maintains the control blocks required to store this information. Since this generates both CPU and storage overhead, COMMON STORAGE MONITOR is not permanently installed into OS/390. Instead, COMMON STORAGE MONITOR can be dynamically installed and removed in accordance with the need to monitor allocation of common storage.

Because of this dynamic installation, and because the data COMMON STORAGE MONITOR collects cannot be obtained through other means, information will not be available about common storage allocated before COMMON STORAGE MONITOR is started. To obtain complete information, therefore, start COMMON STORAGE MONITOR as soon as possible after IPL. (See “Starting and Stopping COMMON STORAGE MONITOR” on page 136 for more information.)

COMMON STORAGE MONITOR can track all requests for common storage or only a subset (such as for CSA or SQA). Monitoring can be further limited to requests for common storage below the 16Mb line. To monitor all common storage and all address spaces, COMMON STORAGE MONITOR consumes about 1-2% of CPU and 300-1024Kb of extended fixed CSA. If this is too much overhead, you can reduce it substantially by tracking only selected storage and address spaces.

Starting and Stopping COMMON STORAGE MONITOR

COMMON STORAGE MONITOR is a component of SYSPROG and is distributed with the OS/390 option as part of AutoOPERATOR. You can use either SYSPROG primary commands from a terminal session command line or an EXEC command to start and stop COMMON STORAGE MONITOR and to summarize the data it collects. Refer to SYSPROG definitions in the *MAINVIEW AutoOPERATOR Options User Guide* or to the IMFEXEC statements in the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* and the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs* for more information.

If COMMON STORAGE MONITOR is already active, it must be stopped before a new COMMON STORAGE MONITOR can be started.

Activate COMMON STORAGE MONITOR by using one of the following methods:

- As a procedure (see “Starting and Stopping COMMON STORAGE MONITOR as a Procedure” on page 136)
- Through PERFORMANCE MANAGER (see “Starting and Stopping COMMON STORAGE MONITOR with ISPF” on page 138)

In an ISPF environment, a confirmation panel is displayed to ensure that COMMON STORAGE MONITOR is not stopped inadvertently.

Note: This method is available only if you have PERFORMANCE MANAGER.

- Using the SYSPROG CSMON service if RESOLVE PLUS is installed
- Using SYSPROG CSMON service (see the SYSPROG definitions in the *MAINVIEW AutoOPERATOR Options User Guide*)
- From an EXEC (see the IMFEXEC statements in the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*)

Note: If you already have the full RESOLVE PLUS product or the PERFORMANCE MANAGER product, these steps may already have been completed.

If COMMON STORAGE MONITOR is stopped and started again, all previous accounting of common storage is lost.

Starting and Stopping COMMON STORAGE MONITOR as a Procedure

For long-term monitoring, start COMMON STORAGE MONITOR during the IPL procedure by adding the following statement to member COMMND00 in SYS1.PARMLIB:

```
com=' start cmfcsmon[, sub=mstr]'
```

This causes COMMON STORAGE MONITOR to start automatically at each IPL. Use the OPT= keyword in the COMMON STORAGE MONITOR JCL to specify monitoring of all or selected address spaces. The default is all address spaces.

The SUB=MSTR parameter in the START command is optional. SUB=MSTR allows COMMON STORAGE MONITOR to start before JES2 or JES3 so that COMMON STORAGE MONITOR can track common storage requests from JES. If SUB=MSTR is included:

- The CMFCSMON procedure must be in SYS1.PROCLIB. This procedure is distributed in BBPARM as \$\$\$CSMON, shown in Figure 24.
- If a STEPLIB is used in the procedure, it must be cataloged in the master catalog.
- The SYSIN DD statement must point to a file cataloged in the master catalog.
- The procedure must not specify any JES data sets.

Common storage allocated during IPL, before initialization of COMMON STORAGE MONITOR, is not accounted for.

```

$$$$CSMON PROC OPT=' START' , MEM=' BBXJOBS' , DSN=' SYS1. PARMLIB'
/******
/*
/** THIS PROC ACTIVATES THE BBXS COMMON STORAGE MONITOR. *
/** IT CAN BE PLACED IN SYS1. PROCLIB AND ACTIVATED AT *
/** IPL WITH THE ENTRY, COM=' S $$$CSMON, SUB=MSTR' IN *
/** COMMDOO OF SYS1. PARMLIB. A STEPLIB IS REQUIRED IF *
/** THE BBLINK LOAD LIBRARY IS NOT IN THE LINK LIST. *
/**
/** SYNTAX: *
/**
/** OPT=' STOP' *
/**
/** OPT=' START, +- -+ +- -+ +- -+ *
/** OPT=' START, |ALL |, |BOTH|, |ANY |', #CSSTE' *
/** |TSU | |CSA | |BELOW| *
/** |STC | |SQA | +- -+ *
/** |BATCH | +- -+ *
/** |JOBNAMES| *
/** +- -+ *
/**
/** IF JOBNAMES IS SPECIFIED MEMBER &MEM IN DATA SET *
/** &DSN MUST CONTAIN THE LIST OF SELECTED JOB NAMES *
/** FOR MONITORING. *
/**
/** EXAMPLES: *
/**
/** S $$$CSMON, OPT=' START, ALL, BOTH, ANY' <- DEFAULTS *
/** S $$$CSMON, OPT=' START, STC, CSA, BELOW' *
/** S $$$CSMON, OPT=' STOP' *
/**
/******
//S1 EXEC PGM=BBXCSMON, TIME=1440, PARM=' &OPT'
/*STEPLIB DD DISP=SHR, DSN=BOOL. BBLINK <- AUTHORIZED LIBRARY
//SYSIN DD DISP=SHR, DSN=&DSN(&MEM) <- JOBNAMES DATA SET

```

Figure 24. \$\$\$CSMON JCL in BBPARM

Starting and Stopping COMMON STORAGE MONITOR with ISPF

If you choose not to start COMMON STORAGE MONITOR during your IPL procedure, start and stop COMMON STORAGE MONITOR using OS/390 PERFORMANCE MANAGER if you have this product installed. You can also start CSMON with a RESOLVE primary command. Refer to the SYSPROG definitions in the *MAINVIEW AutoOPERATOR Options User Guide* and the IMFEXEC statements section in the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* and the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.

Chapter 18. Implementing Continuous State Manager

This chapter provides information that will help you to implement Continuous State Manager (CSM) and attain the best possible performance.

Performance Considerations

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 book *TSO Customization Guide* for more information about VLF.

Define CSM EXECs (CSMUP, CSMDOWN, CSMBUILD, and CSMINIT) as high-priority AutoOPERATOR EXECs in BBPARM member AAOEXP00. See Chapter 3, “Continuous Operation: Using the Dynamic Parameter Manager” on page 31, for more information about AAOEXP00 and high-priority EXECs.

Additional Implementation Considerations

The following sections describe installing and accessing 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 which group 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 is running. You can override the group by specifying a different group name on the CSMALTDB parameter in the OS/390 start command for the BBI-PAS. Refer to “Subsystem (SS) JCL Parameter CSMALTDB” 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 which is used to specify the name of a CSM group that is used by CSM when it initializes. When you make a change to the CSMALTDB parameter, a cold start of the subsystem with VPOOL=RESET specified is required for the change to take effect.

An example of specifying the group name on the CSMALTDDB parameter follows:

```
CSMALTDDB=SYSB
```

where SYSB is the name of a SSID not local to the BBI-SS PAS that is starting CSM.

End-of-Memory Initiated CSM EXEC (CSMEOM)

Normally, object termination can be monitored using standard OS/390 and JES messages. However, there are situations when monitoring based on these messages is not sufficient because an object may 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 it. CSM detects objects that terminate at End-of-Memory by using the CSMEOM EXEC.

See the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* for more information about EOM EXECs.

Accessing CSM for the First Time

Take the following steps before you start using CSM from the terminal session:

1. Obtain and enter a valid key for the AutoOPERATOR for OS/390 option in BBPARM member BBKEYS.

The AutoOPERATOR for OS/390 option is required for CSM processing, which includes issuing both OS/390 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 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 each group has 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 RULESCAN== parameter to I for INDIVIDUAL.

The RULESCAN== parameter controls the global automation strategy for all AutoOPERATOR Rules. For more information about Rules and automation strategy, refer to the *MAINVIEW AutoOPERATOR Basic Automation Guide*.

4. In BBPARM member AAOEXP00, enter the names of all CSM EXECs and ensure these are defined as High Priority EXECs. The names of the CSM EXECs are listed in the AAOEXP00 member.
5. Create a Shared Repository data set that will contain all defined CSM objects and schedules.

If the IBM library for SAA REXX/370 (program number 5695-014) is not available, customize the BBREXXALT JCL found in BBSAMP to run on your system. Run BBREXXALT to install the REXX/370 alternate library. If you do not run this job, the compiled REXX object in INITCSM will fail.

You can create a repository data set by one of two methods:

- Complete the step in AutoCustomization called “Allocate Shared Repository Data Set for CSM”.
- Run the batch job contained in BBSAMP member named INITCSM. For more information, refer to “Creating a Shared Repository Data Set for CSM” on page 143.

6. In BBPARM member BBIVARxx member, 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

CSMREPDS = hilevel.datasetname

where

CSMREPDS Is the name of a shared variable that must be present for CSM to function correctly

hilevel.datasetname Is the name of the Shared Repository data set created in Step 5 above.

Note: If you are implementing CSM in an environment where the AutoOPERATOR subsystem(s) have JESFLTR parameters specified in the BBISSPxx member, and are running more than one CSM system on an LPAR, you will need to add a line in the BBIVARxx member:

CSMMSPL = ' Y'

This creates a shared variable that the CSM Rule Sets use to determine whether or not certain communications messages are suppressed or not.

7. Start the BBI-SS PAS and specify START=COLD, VPOOL=RESET

When the cold start is complete, the BBI Journal should contain the following message:

ACM003I CSM initialization complete

By completing these steps, CSM automatically initializes at BBI-SS PAS startup when message PM0010I is issued.

8. If you are migrating from a previous version of AutoOPERATOR (either AutoOPERATOR 4.1 or 5.1) and you want to migrate your database repository, you must complete this step to convert your database.

- a. Start the BBI-SS PAS and specify `START=COLD, VPOOL=RESET`.
 - b. Follow the instructions in “Migrating a CSM Database from a Previous Release” on page 146.
9. 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 Basic Automation Guide* for more information about adding groups and objects.
 10. Once you have either created a new group or converted all your AutoOPERATOR 4.1 or 5.1 database repositories, enable the newly created groups with the Enable line command on the CSM Object Groups panel.

Note: When a converted group becomes enabled, the statistics about the objects are offloaded to the data set specified in the `Dat aset Name` field, as shown in Step 4 on page 147.

After the data is offloaded, it is available in comma-separated format. Figure 25 shows an example of what the contents in the data set might look like:

```

BROWSE      BAOKMZ. KMZ1. OFFLOAD                               Line 00001813 Col 001 080
Command ==>>>                                               Scroll ==>> PAGE
P, AAOC SM22, 11: 46: 47. 764226, 20000725, SJSD, KMZ1, KMZ1
P, AAOC SM58, 11: 46: 48. 320489, 20000725, SJSD, KMZ1, KMZ1
S, AAOC SM27, 11: 46: 51. 367131, 20000725, SJSD, KMZ1, KMZ1
S, AAOC SM55, 11: 46: 53. 437561, 20000725, SJSD, KMZ1, KMZ1
S, AAOC SM59, 11: 46: 53. 626765, 20000725, SJSD, KMZ1, KMZ1
P, AAOC SM17, 11: 47: 43. 247170, 20000725, SJSD, KMZ1, KMZ1
S, AAOC SM01, 11: 47: 52. 415915, 20000725, SJSD, KMZ1, KMZ1
***** Bottom of Data *****

```

Figure 25. Example of Offloaded Data

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

Important Note

The information contained in this section is part of Step 5 as described in “Accessing CSM for the First Time” on page 140. 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 which will contain all defined CSM objects and schedules. Complete the procedure in this section only if you did not create a Shared Repository by completing the steps in AutoCustomization called “Allocate Shared Repository Data Set for CSM”.

To customize this jobstream:

1. Modify 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 jobstream.

After it 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 current space allocations below 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 145.

```

//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 HI LEVEL=' *HI LEVEL*',
//          SSID=' *SSID*',
//          SYSOUT=*
//*
//DEFINE EXEC PGM=IDCAMS, REGION=3M
//SYSPRINT DD SYSOUT=&SYSOUT
//*
//INIT EXEC PGM=IRXJCL, REGION=3M,
//          PARM=' INITREP'
//STEPLIB DD DISP=SHR, DSN=&HI LEVEL. . BBLINK
//SYSEXEC DD DISP=SHR, DSN=&HI LEVEL. . BBPROC
//CSMREPS DD DISP=SHR, DSN=&HI LEVEL. . &SSID. . BBI REPS
//SYSTSPRT DD SYSOUT=&SYSOUT
//          PEND
//*
//INITIT EXEC INITCSM
//DEFINE. SYSIN DD *
DELETE (*HI LEVEL*. *SSID*. BBI REPS)
SET MAXCC=0
DEFINE CLUSTER( NAME(*HI LEVEL*. *SSID*. BBI REPS) -
                KEYS(128 0) -
                RECORDSIZE(14237 14237) -
                SHAREOPTIONS(4 )) -
DATA( NAME(*HI LEVEL*. *SSID*. BBI REPS. DATA) -
      SPEED -
      CISCZ(14336) -
      CYL(50, 5) -
      VOL(*VOL*)) -
INDEX( NAME(*HI LEVEL*. *SSID*. BBI REPS. INDEX) -
       TRK(10, 10) -
       VOL(*VOL*))
LISTC ENT(*HI LEVEL*. *SSID*. BBI REPS) ALL
/*

```

Figure 26. BBSAMP member INITCSM Sample JCL

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 can hold 7 physical blocks of data. Each record in the repository has a length of 14,237 bytes. Each cylinder on a 3390 has 15 tracks of data.

Therefore, for example, if you have thirty groups, one master and eighteen objects you would have 137 to store. The calculation is shown in Figure 27:

$$\begin{array}{l} ((30 \text{ groups} \times 4 \text{ records}) + (18 \text{ objects} \times 1 \text{ record}) + (\text{one master record})) = 137 \text{ records} \\ \quad \quad \quad (1) \qquad \qquad \qquad (2) \qquad \qquad \qquad (3) \end{array}$$

Figure 27. Calculating Repository Size

where:

- (1) Each group takes four repository records.
- (2) Each object takes one repository record.
- (3) 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 that means that you need 39.14 tracks to store one hundred and thirty seven records. 39.14 tracks fits on 2.60 cylinders.

39.14 tracks / 15 tracks per cylinder \approx 2.60 cylinders of data

- a. For the first two fields, enter the name of your CSM data set that you are replacing, and the member name (for example: CSMssid).
- b. For the third field (Data Set Name), specify the name of a data set where the conversion process logs information related to the conversion.

The data set may be a pre-allocated RECFM=FB, LRECL=80 data set or the data set may be dynamically allocated with these DCB attributes: RECFM=FB, LRECL=80, BLKSIZE=23680.

For the fourth field (Disposition), specifying MOD for disposition causes the Routine to append data to end of the sequential data set.

Specifying SHR will cause 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, Group Detail Control panel is displayed.

4. Enter information related to the CSM group being converted on the Group Detail Control panel.

For the Ruleset Id field, you must add the Rule Set ID you specify on this panel to the RULESET= parameter in BBPARM member AAOPRM00 (refer to Step 2 on page 140).

Figure 30 shows an example of a completed Group Detail Control panel.

```

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

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

Rule information:
Ruleset Dataset ==> BAOKMZ. AA061. KMZ1. UBBPARM
Ruleset Id      ==> G2          Rule Prefix ==> ACMI      Suffix ==> 0001

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

```

Figure 30. Example of the Group Detail Control Panel

The default name of the CSM group is taken from the last four character of the CSMssid member being converted. If the Group Name you choose already exists, CSM issues a short message in the upper right corner.

5. Enter a new Group Name in the pop-up window (Figure 31) or enter CANCEL to terminate the conversion.

```

BMC Software ---- CSM Conversion - Group name ----- AutoOPERATOR
COMMAND ==>

Group name NS61 already exists. Specify new name for CSM Group.

  Group Name ==> _____

Enter END command to process or CANCEL to leave

```

Figure 31. Pop-Up Window for New Group Name

6. As the conversion progresses, a pop-up window (Figure 32) is displayed and as every 5 objects are converted, the counter is updated.

```

BMC Software ----- CSM Conversion Status ----- AutoOPERATOR

                                     CSM Conversion in Progress

                                     15 of 164 objects converted

```

Figure 32. CSM Conversion Status Pop-Up Panel

7. When the conversion is complete, the CSM Conversion Log is displayed (Figure 33).

```

***** Top of Data *****
BMC Software                CSM Conversion Log

Input database ... BAONIS.NS61.BBPARAM(CSMNEW)
Converted by ..... BAONIS2 at: 16:54 on: 10/05/2000
New Group ..... NEW
Objects processed . 50
Conversion status . Successful for group NEW. No exceptions.

  Objects      Excepti on Notes
  -----      -
AAOCSMN1
AAOCSMN2
AAOCSMN3
AAOCSMN4
AAOCSMN5
AAOCSMN6

```

Figure 33. CSM Conversion Log

During conversion, you might encounter conflicts or mismatches with object definitions. For more information about dealing with object-definition mismatches, refer to “Dealing with Object-Definition Mismatches after Converting” on page 150.

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 34).

```

BMC Software ----- Global Overview ----- Conversion Successful
COMMAND ==>                                     Scroll ==> PAGE
                                                TGT --- NS61

Primary Commands: EXCEption  Group      SORT   SAVE   ADD
                  CALendar   CANCEL  UNsort  CONVERT
                  Locate     CMDSHOW Xref    SAVERULE

Line Command - Enter "/" to Select Action
LC  Object   Group-Id System  Actual      Desi red    Ind  Type
-----
___ AA0QAS01 NS61          UNKNOWN     UNKNOWN
___ AA0NS61  NS61          UNKNOWN     UNKNOWN
___ CAS      NS61          UNKNOWN     UNKNOWN
___ CICS11   NS61          UNKNOWN     UNKNOWN
___ IMS71X   NS61          UNKNOWN     UNKNOWN
___ TSO      NS61          UNKNOWN     UNKNOWN
___ MVS
***** Bottom of data *****

```

Figure 34. CSM Data Set Conversion Completed

9. Repeat Step 2 through Step 8 for each version 4.1 or 5.1 CSM data set and member name to be converted.

| Dealing with Object-Definition Mismatches after Converting

The conversion process converts every object into one record in the Shared Repository data set. Because each object may have only one record within the Shared Repository, but it may be a member of more than one group, you may encounter object definition mismatches. If a mismatch is encountered, it will be identified on the CSM Conversion Log data set (Figure 33 on page 148).

An object definition mismatch may occur when an object already has a record in the Shared Repository and it is being converted as a member of a different group and has non-matching definitions (see Table 5).

Table 5 shows all possible definitions for an object. The first column contains the object definitions that may be different per group. The second column contains the object definitions that must be identical for all groups containing this object.

Table 5. Object Definitions

Object Definitions that Do Not Have to Match in the Repository	Object Definitions that Must Match within the Repository
Parents Client Server Schedules	Object Type Restart Only Verify Force Down Start Command/EXEC User Post Start EXEC Stop Command/EXEC User Post Stop EXEC Cancel Command/EXEC Recovery Command/EXEC Start Command Time Out Stop Command Time Out Start Command Limit Start Event Text ID and Type Stop Event Text ID and Type Abnormal Termination Events TSO Send User ID Pager Contact Pager Info Alert Queue Rules Generation

You can resolve a mismatch by starting CSM and use the CSM object definition panel to edit the object and ensure that the object definitions (with the exception of the object's Parents, Clients, Servers and Schedules) for *each object in every group match exactly*.

If you do not want each object in every group to have identical definitions, you can also resolve a mismatch by using variables in CSM object definitions which can resolve to different values for the different groups on different BBI-SS PASs. The definition can contain AutoOPERATOR SHARED variables. For more information, refer to the section "Using Variables in the Object Definition Fields" in Chapter 11 of the *MAINVIEW AutoOPERATOR Basic Automation Guide*.

Another way to resolve a mismatch is to place the new group into a different Shared Repository data set.

Sample JCL for Copying the Shared Repository

Figure 35 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 name BAONI S. CSM. DATABASE and BAONI S. CSM. ALTDB with your data set names.

```
//JOB CARD
//CPY EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//F1 DD DSN=BAONI S. CSM. DATABASE, DISP=SHR <== Current Repository data set
//F2 DD DSN=BAONI S. CSM. ALTDB, DISP=SHR <== New Repository data set
//SYSIN DD *
        REPRO I FILE(F1) O FILE(F2)
/*
```

Figure 35. JCL Example for Copying Shared Repository

Chapter 19. Specific AutoOPERATOR Parameters and JCL

This chapter contains information about the following:

- Enabling REXX EXEC support
- Detecting EXEC loops
- Adding JES2 data to the MESSAGE ACTIVITY STATISTICS and ADDRESS SPACE application displays (OS/390 option only)

REXX Support for EXECs

The REXX EXEC environment is disabled when AutoOPERATOR is initially installed and activated. To enable REXX support, perform the following two steps and then restart your BBI-SS PAS:

1. 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.

2. Add the REXX=YES parameter to the AAOEXP00 member in the BBPARM data set.

The default is REXX=NO.

EXEC Loop Detection

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 above.

Parameter	Description
-----------	-------------

TIMEXLIM=nnnn	
----------------------	--

Optional.

Specifies the CPU time limit for an EXEC. If this time is exceeded, the EXEC is terminated with a USER 3001 abend code.

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 includes multiprocessor machines.

Refer to Chapter 3, “Continuous Operation: Using the Dynamic Parameter Manager” on page 31 for information about using the Dynamic Parameter Manager application to change these parameters. Also, refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs* for more information about performance issues and using EXECs.

Priority 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

The following describes the BBPARM member AAOEXP00 parameters you can modify:

Keyword	Description
----------------	--------------------

EXEC=xxxxxxx	
---------------------	--

Enter the names of EXECs that you want to execute a high priority.

MAXNORM=nn	
-------------------	--

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.

Note: This is limited more by BBI-SS PAS memory requirements than by the 99 MAX range.

MAXHIGH=n	
------------------	--

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.

Refer to Chapter 3, “Continuous Operation: Using the Dynamic Parameter Manager” on page 31 for information about using the Dynamic Parameter Manager application to change these parameters. Also, refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs* for more information about performance issues and using EXECs.

Defining the Message Counter Table

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:

MCTSI ZE=xxxx

in BBPARM member BBISSP00.

This 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:

```
$HASP309    INIT 4 ACTIVE
$HASP309    INIT 2 INACTIVE
$HASP309    INIT 4 INACTIVE
```

This results in the total number of the activities associated with the three events appearing on the Event Activity Statistics panel as:

```
Total messages:    60
```

See the *MAINVIEW AutoOPERATOR Basic Automation Guide* for more information about the Rule Processor application and the Event Activity Statistics panel.

Displaying MVS JES2 Job Numbers in Address Space Application

If you have the MVS option for 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 class.

Starting BBI-SS PAS before JES

If you want AutoOPERATOR to automate functions that are normally performed before JES is operational, the 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, VOL=SER=... , UNIT=...  
//BBDUMP02 DD DISP=SHR, DSN=dumpdsn2, VOL=SER=... , UNIT=...  
//BBDUMP03 DD DISP=SHR, DSN=dumpdsn3, VOL=SER=... , 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 OS/390 console, or an EXEC in the standard BBI command format.

These dump datasets are used when a BBI FORMATTED dump is taken which 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, an SVC dump is taken and these SYSUDUMP datasets 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. Start the BBI-SS PAS with the command:

```
S SSJCL. SSID, SUB=MSTR
```

6. Change the following two DD statements in the BBI-SS PAS JCL:

```
//SYSTSIN DDdummy  
//SYSTSPRT DDdummy
```

Note: The terminal session (TS) can be started only after JES has initialized.

If the AutoOPERATOR SS is started prior to JES, the IMFEXEC SUBMIT command cannot be used and you cannot use any command or program that attempts to allocate to a SYSOUT data set.

If you use 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 v1.2 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 `$ACTIVATE` and `$TJOBDEF,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=5. IMFJNUM=5 is the default to ensure compatibility for existing AutoOPERATOR customers. The new setting will be effective after the next warm start of the BBI-PAS. BMC Software recommends that new customers specify the AAOPRMxx option IMFJNUM=V immediately.

When the IMFJNUM option is not specified or is set to five, 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 will be 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 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 AutoOPERATOR.
- A highlighted error WTO (AU6021E) message is issued the first time 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.

Chapter 20. Accessing Advanced SYSPROG Services from AutoOPERATOR

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 AutoOPERATOR and to execute the advanced early warning system (AEWS) in the AutoOPERATOR address space.

The full set of Advanced SYSPROG services are automatically available to AutoOPERATOR if you have MAINVIEW for OS/390 installed into the same SMP zones as 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, then you must make the following JCL modifications:

- Add MAINVIEW for OS/390's BBLINK data set as the first data set on the STEPLIB DD statement in the SS JCL.
- Add MAINVIEW for OS/390's BBLOAD data set as the first data set on the BBILOAD DD statement in the SS JCL.

If you have combined MAINVIEW for OS/390's BBLINK and BBLOAD data sets, make a reference to the combined data set on both STEPLIB and BBILOAD.

AutoOPERATOR requires all data sets in the BBILOAD 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 MAINVIEW for OS/390's BBLOAD data set before adding it to the BBILOAD concatenation.

Not all the Advanced RESOLVE commands function under AutoOPERATOR. For example, the CONSOLE service does not function because it uses full screen support and the 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 AutoOPERATOR Address Space

To run the advanced early warning system (AEWS) in the AutoOPERATOR address space, add the LIB DD statement (as described in the *RESOLVE PLUS Reference Manual*) to the AutoOPERATOR subsystem JCL.

Part 3. Implementing AutoOPERATOR Security

Part 3 of this guide describes the implementation of product security.

Chapter 21. Authorization for Special AutoOPERATOR Functions	163
Command Authorization	163
Chapter 22. Authorizing EXECs Invoked by an Operator or the Rule Processor	165
Chapter 23. Implementing AutoOPERATOR for CICS Function Security	
(CMRSECU)	167
CMRSECU TYPE=INITIAL Statement (Required)	168
CMRSECU TYPE=ENTRY Statement by Operator ID	172
CMRSECU TYPE=FINAL Statement (Required).	173

Chapter 21. Authorization for Special AutoOPERATOR Functions

Depending on how an AutoOPERATOR EXEC is invoked, it is assigned an 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:

Target ID of the remote system	For an EXEC scheduled through a message from a target
Subsystem ID of the remote BBI-SS PAS	For an EXEC scheduled from a BBI-SS PAS message or one that is time-initiated
Jobname or RACF USERID	For an externally initiated EXEC

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. Following is a summary of authorization commands for AutoOPERATOR.

For all AutoOPERATOR options:

Command	Description
EXEC	Authority to invoke EXECs
MVSCMD	Authority to issue MVS commands
RULEAUTH	Rules processor authorization which can be READ, UPDATE, or none

For the AutoOPERATOR for IMS option only:

Command	Description
IMSCMD	Authority to issue IMS commands
IMSMMSG	Authority to send IMS messages to LTERMs
IMSTRAN	Authority to issue IMS transactions

For the AutoOPERATOR for OS/390 option only:

Command	Description
RESAUTH	Allowed to issue authorized SYSPROG commands For more information about issuing Advanced SYSPROG commands and the security exit, refer to the <i>MAINVIEW AutoOPERATOR Options User Guide Options</i> .

A description of how to use these commands is in the *Implementing Security for MAINVIEW Products*. The comment field of the USERID member of the BBPARM data set also describes the commands and their valid parameters.

Chapter 22. Authorizing EXECs Invoked by an Operator or the Rule Processor

A terminal operator on one BBI-SS PAS can invoke an EXEC on another BBI-SS PAS by using the MVS MODIFY command. Also, a message filtered by the rules processor application can invoke an EXEC that invokes another EXEC on a different BBI-SS PAS. In both cases a BBPARM member must authorize the BBI-SS PAS' reception of the EXEC. 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 %execname starts an EXEC on SS02, then SS02 must have a BBPARM member SS01. And if %execname then issues an EXEC from SS02 to be invoked on SS01, then SS01 must have a BBPARM member SS02. 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, AutoOPERATOR issues error message AM3119W and does not invoke the EXEC.

Chapter 23. Implementing AutoOPERATOR for CICS Function Security (CMRSECU)

This chapter discusses function security for MAINVIEW for CICS and/or AutoOPERATOR for CICS and how it is specified by assembling and linking the CMRSECU module using the CMRSECU macro.

This table can be reset dynamically with the command:

```
. RESET PARM CMRSECU
```

See the *MAINVIEW Implementation Guide* for a description of the RESET control command and supported parameters.

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.
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
- AutoOPERATOR for CICS action service

TYPE=ENTRY 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.)

TYPE=FINAL 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 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.

CMRSECU TYPE=INITIAL Statement (Required)

The CMRSECU TYPE=INITIAL statement defines security for each:

- MAINVIEW for CICS function or action service
- 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 non-specified targets. If an * is not specified and a non-specified target is referenced online, the first CMRSECU TYPE=INITIAL statement is used to define 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 172).
- NO indicates that the function is not secured. All operator IDs allowed access to MAINVIEW for CICS and/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 for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.

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.

Action parameters for the CMRSECU TYPE=INITIAL statement are listed in Table 6 on page 169.

Display parameters for the CMRSECU TYPE=INITIAL statement are listed in Table 7 on page 170.

The default for each parameter is NO.

Table 6. Action Parameters for the CMRSECU TYPE=INITIAL and TYPE=ENTRY Statements

Parameter	MAINVIEW for CICS Service Description	Applicable Rule Command or IMFEEXEC CICS Command
ALTAID=NO YES	Specifies whether to secure the change capability for the Automatic Initiate Descriptor statistics display.	N/A
ALTCLAS=NO YES	Specifies whether to secure the change capability for the classes display.	ALTER
ALTCONN=NO YES	Specifies whether to secure the status change capability in the CONNECT service. The expanded displays are included.	N/A
ALTDEST=NO YES	Specifies the status change capability for the DEST service.	DISABLE DEST ENABLE DEST PURGE DEST
ALTER=NO YES	Specifies whether to secure the change capability for the virtual storage address display.	ALTERVS
ALTFILE=NO YES	Specifies whether to secure the status change capability in the FILE, DDIR, DSNAMES, and DATATABL services.	ALLOC CLOSE FILE DISABLE FILE ENABLE FILE FREE OPEN FILE RECOVERDB STARTDB STOPDB
ALTICES=NO YES	Specifies whether to secure the change capability in the ICE display.	PURGE ICE
ALTPROG=NO YES	Specifies whether to secure the status change capability in the PROGRAM and REMOTES services.	DISABLE PROGRAM DROP ENABLE PROGRAM LOAD PROGRAM NEWCOPY PROGRAM
ALTTASK=NO YES	Specifies whether to secure the task kill capability in the TASK, ENQUEUE, TERMINAL, CONSOLES, and CONNXPND services.	KILL
ALTTERM=NO YES	Specifies whether to secure the status change capability in the TERMINAL and CONSOLES services.	ACQUIRE INSERVE OUTSERVE RELEASE SPURGE
ALTTRAN=NO YES	Specifies whether to secure the status change capability in the TRAN display.	ENABLE TRAN DISABLE TRAN
ALTTSUT=NO YES	Specifies whether to secure the status change capability in the TSUT display.	PURGE TSUT

Table 6. Action Parameters for the CMRSECU TYPE=INITIAL and TYPE=ENTRY Statements (Continued)

Parameter	MAINVIEW for CICS Service Description	Applicable Rule Command or IMFEXEC CICS Command
SETTCLA=NO YES	Specifies whether to secure the alteration of the attributes of a transaction class.	ALTER TCLASS
SETDMPD=NO YES	Specifies whether to secure the alteration of the attributes of a dump data set.	ALTER DUMPDS
SETSYS=NO YES	Specifies whether to secure the alteration of the CICS system attributes.	ALTER SYSTEM
SETTCPS=NO YES	Specifies whether to secure the alteration of a TCP/IP service.	ALTER TCPIP SERVICE
SETJVMP=NO YES	Specifies whether to secure the alteration of the status of the JVM pool.	ALTER JVMPOOL

Table 7. Display Parameters for CMRSECU TYPE=INITIAL and TYPE=ENTRY Statements

Parameter	MAINVIEW for CICS Service Description
ABEND=NO YES	Specifies whether to secure the Abend status display.
AID=NO YES	Specifies whether to secure the Automatic Initiate Descriptor (AID) statistics display.
CLASS=NO YES	Specifies whether to secure the class maximum statistics display.
CONNECT=NO YES	Specifies whether to secure the connection statistics display. The expanded displays are included.
DB2SYSP=NO YES	Specifies whether to secure the DB2 system parameters and DBCTL displays.
DB2TASK=NO YES	Specifies whether to secure the DB2 task statistics and DBCTASK displays.
DEST=NO YES	Specifies whether to secure the destination ID information display.
DLI=NO YES	Specifies whether to secure the DL/I statistics display.
DUMP=NO YES	Specifies whether to secure the virtual storage address display.
ENQUEUE=NO YES	Specifies whether to secure the enqueued resources display.
EXITS=NO YES	Specifies whether to secure the task-related and global user exits display.
FILE=NO YES	Specifies whether to secure the files statistics display in the DL/I data base display, the file dsnames displays, and the CICS data tables displays. The expanded displays are included.
HISTORY=NO YES	Specifies whether to secure the transaction history display.
ICES=NO YES	Specifies whether to secure the Interval Control Element (ICE) statistics display.
JOURNAL=NO YES	Specifies whether to secure the journal file statistics display.
MONITOR=NO YES	Specifies whether to secure the tuning opportunities display.
NUCLEUS=NO YES	Specifies whether to secure the nucleus module, LPA, and TCB displays.
PLAN=NO YES	Specifies whether to secure the DB2 plan display.

Table 7. Display Parameters for CMRSECU TYPE=INITIAL and TYPE=ENTRY Statements (Continued)

Parameter	MAINVIEW for CICS Service Description
PPST=NO YES	Specifies whether to secure the DL/1 PPST/TASK activity display.
PROBLEM=NO YES	Specifies whether to secure the realtime problem analysis display. Note: If PROBLEM=YES, you must specify OPID=xxxxxxx in the TYPE=ENTRY statement. The OPID= specification must equal the AUTOID or USRID specified in BBPARM (see page 233) 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
- AutoOPERATOR for CICS action service

which 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 which 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=xxxxxxx|*

OPID Specifies an operator ID, where xxxxxxx is:

- TSO user ID for access through a terminal session
- EXCP user ID for access through an EXCP session
- VTAM user ID for access through a VTAM session

* Is a generic qualifier for an operator ID; for example:

AB*

The display and action parameters are the same as the CMRSECU TYPE=INITIAL parameters.

Action parameters for the CMRSECU TYPE=INITIAL statement are listed in Table 6 on page 169.

Display parameters for the CMRSECU TYPE=INITIAL statement are listed in Table 7 on page 170.

CMRSECU TYPE=FINAL Statement (Required)

The CMRSECU TYPE=FINAL statement has one parameter and ends this option generation; for example:

```
CMRSECU TYPE=FINAL
```

Part 4. Appendixes

Part 4 of this guide contains the following appendixes:

Appendix A. Issuing Commands from Terminal Sessions	177
Issuing OS/390 Commands	177
Using Consoles to Issue OS/390 Commands	177
Issuing BBI Commands	177
Issuing IMS Commands	178
Appendix B. Changing the Use of CICS Recoverable Temporary Storage	179
Appendix C. CICS Transaction and Program Names	181
Transaction Entries for MAINVIEW for CICS and AutoOPERATOR for CICS	181
Program Entries for MAINVIEW for CICS and AutoOPERATOR for CICS	183
Release Independent Programs	183
Release Dependent Programs	183
Appendix D. Manual Activation of BBI-SS to CICS Communication	187
Appendix E. BBPARM Data Set Members for AutoOPERATOR	189
BBPARM Member AAOALS00 Parameters (AutoOPERATOR Only)	190
BBPARM Member AAOALT00 Parameters (AutoOPERATOR Only)	192
BBPARM Member AAOARP00 Parameters (AutoOPERATOR Only)	193
BBPARM Member AAOEWD00 Parameters (AutoOPERATOR Only)	194
BBPARM Member AAOEXP00 Parameters (AutoOPERATOR Only)	195
BBPARM Member AAOGME00 Parameters (AutoOPERATOR Only)	199
BBPARM Member AAOMQL00 (AutoOPERATOR Only)	202
BBPARM Member AAONCD00 Parameters (AutoOPERATOR Only)	204
BBPARM Member AAOPRM00 Parameters (AutoOPERATOR Only)	205
BBPARM Members AAORUL00 Parameters (AutoOPERATOR Only)	208
BBPARM Member AAOTRN00 Parameters (AutoOPERATOR Only)	209
BBPARM Member AAOTSP00 Parameters (AutoOPERATOR Only)	210
BBPARM Member BBIVAR00 Parameters (AutoOPERATOR Only)	217
Appendix F. Sample Data Set Members	221
Sample Members for AutoOPERATOR Only	221
Sample Members for AutoOPERATOR for CICS and MAINVIEW for CICS	223
Sample Members for AutoOPERATOR for IMS, MAINVIEW for IMS, and MAINVIEW for DBCTL	224
Appendix G. Expanding the PROFILE Pool Data Set	225
Using the XTLOAD Utility	225
Appendix H. AutoRULE Worksheet	227
Section 1: MPF List Conversion Parameters	227
Section 2: AutoMate Conversion Parameters	228
Section 3: Generating Rules	228
Section 4: Creating Rules for Production Job Abends	229
Section 5: Creating Rules for Production Job Abends	229
Describing AutoRULE Flags	230
Appendix I. How Product Libraries Should Be Used	233

Parameter Library	235
BBPARM	235
UBBPARM	235
IMS-Specific Parameter Library (IMF and AutoOPERATOR for IMS Only)	236
Sample Library	236
BBSAMP	236
UBBSAMP	236
Profile Library	237
BBPROF	237
SBBPROF	237
User BBPROF	238
Procedure Library (AutoOPERATOR Only)	238
BBPROC	238
UBBPROC	238
BBI-2 Product Library Use Summary	239
Appendix J. OSPI Terminal Definitions for CICS and IMS	241
IMS Terminal Definitions	241
CICS Terminal Definitions	241
Autoinstall Terminals	241
Explicit Terminal Definitions	241
RDO Macro	242
Appendix K. Establishing a VTAM Connection for AutoOPERATOR to PATROL EM Communication	245
Defining VTAM ACBs	245
Defining 3270 Printers to VTAM	246
Resetting BBPARM Member AAOALT00	246
Controlling the AutoOPERATOR to PATROL EM Interface	247
Modifying the VTAM Connection with BBPARM Member AAOALT00	248
3287 Data Stream Format for AutoOPERATOR to PATROL EM	249
Diagnostics	251

Appendix A. Issuing Commands from Terminal Sessions

AutoOPERATOR allows OS/390, IMS, CICS, and DBCTL commands to be issued from a terminal session (TS) command line.

Issuing OS/390 Commands

OS/390 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 AutoOPERATOR for OS/390 (MAO) key is active in the subsystem, 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 OS/390 Commands

AutoOPERATOR consoles are either subsystem consoles or Extended MCS (X-MCS) consoles. Refer to Chapter 4, "MVS Console Considerations" on page 65 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 OS/390 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.

Appendix B. Changing the Use of CICS Recoverable Temporary Storage

BMC Software occasionally starts a CICS transaction 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 used by BMC Software (CMRI) must be changed by applying the following ZAP:

```
NAME CMRI NT2 CMRI NT2
VER 00BC C3D4D9C9
REP 00BC xxxxxxxx (any nonrecoverable prefix)
```

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” in the *MAINVIEW AutoOPERATOR Customization Guide* when implementing AutoOPERATOR for CICS **only**
- “Implementation Considerations” in the chapter “Standard Implementation Procedures ” in the *MAINVIEW for CICS Customization Guide*.

Appendix C. CICS Transaction and Program Names

This section describes the transaction and program names generated by MAINVIEW for CICS and MAINVIEW AutoOPERATOR for CICS.

Transaction Entries for MAINVIEW for CICS and AutoOPERATOR for CICS

Table 8 describes the transaction IDs used by MAINVIEW for CICS and MAINVIEW AutoOPERATOR for CICS. Table 9 on page 182 lists ZAPs that may be applied to change these IDs.

Table 8. Transaction IDs

Tran ID	Description
FCD2	Performs an action request that uses CICS services for completion. It is scheduled once for each request to ensure proper serialization.
FCM1	Routes messages to terminals through BMS for AutoOPERATOR for CICS message broadcast.
FIC2	Captures and records MAINVIEW for CICS statistical data. It runs once each minute.
FST2	Activates (FST2 QON), deactivates (FST2 QOFF), and controls the communications and data collection components under CICS and switches dual journals (FST2 SWITCH). It normally runs asynchronously (without a terminal), but it can be entered from an OS/390 console defined to CICS.
JNL2	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.
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 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 9. ZAPs for Transaction ID Changes

Tran ID	ZAP Required if Transaction ID Is Changed
FCD2	NAME CMRI NT2 CMRI NT2 VER 00C0 C6C3C4F2 REP 00C0 xxxxxxxx New Transaction ID
FCM1	NAME CMRI NT2 CMRI NT2 VER 00A8 C6C3D4F1 REP 00A8 xxxxxxxx New Transaction ID
FIC2	NAME CMRI NT2 CMRI NT2 VER 00B8 C6C9C3F2 REP 00B8 xxxxxxxx New Transaction ID
FST2	NAME CMRI NT2 CMRI NT2 VER 00B0 C6E2E3F2 REP 00B0 xxxxxxxx New Transaction ID
JNL2	NAME CMRI NT2 CMRI NT2 VER 00B4 D1D5D3F2 REP 00B4 xxxxxxxx New Transaction ID
SMN2	(No ZAP required)

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 AutoOPERATOR for CICS

This section provides a descriptive list of Program entries generated for MAINVIEW for CICS and AutoOPERATOR for CICS. The lists are grouped by release independent and release dependent programs.

Release Independent Programs

The programs listed in Table 10 can operate under any CICS release.

Table 10. Release Independent Programs

Program	Description
CMRINT2	CICS products internal table
CMRLOGM	CICS products message logging program
CSLOAD	CICS products common services program
CMRTMON	CICS products interactive startup program
CMRTMAP	CICS products interactive startup BMS mapset
CBQCMD	AutoOPERATOR for CICS broadcast program
CMRXTEG	MAINVIEW for CICS verification program
OLTSCAO	Start and stop AutoOPERATOR for CICS
OLTVCAO	Test software compatibility

Release Dependent Programs

These programs operate under a specific CICS release. The release dependency is indicated by the following suffixes used with the program name. Table 11 on page 184 is a descriptive list of the programs.

Suffix	CICS Release
6	CICS/ESA 4.1
9	CICS Transaction Server 1.2
0	CICS Transaction Server 1.3
1	CICS Transaction Server 2.1
2	CICS Transaction Server 2.2

Table 11. Release Dependent Programs

Program	Description
CMRFSET6 CMRFSET9 CMRFSET0 CMRFSET1 CMRFSET2	CICS products initialization program
CMRJRNL6 CMRJRNL9 CMRJRNL0 CMRJRNL1 CMRJRNL2	CICS products BBI-SS communications program
CMRFCMD6 CMRFCMD9 CMRFCMD0 CMRFCMD1 CMRFCMD2	CICS products action service program
CMRTDPX6 CMRTDPX9 CMRTDPX0 CMRTDPX1 CMRTDPX2	AutoOPERATOR for CICS transient data exit program
CMRCMPX6 CMRCMPX9 CMRCMPX0 CMRCMPX1 CMRCMPX2	MAINVIEW for CICS monitoring task-related user exit program
CMRKCPX6 CMRKCPX9 CMRKCPX0 CMRKCPX1 CMRKCPX2	MAINVIEW for CICS data collection global exit program
CMRKCPX6 CMRKCPX9 CMRKCPX0 CMRKCPX1 CMRKCPX2	MAINVIEW for CICS data collection global exit program
CMRTRUE6 CMRTRUE9 CMRTRUE0 CMRTRUE1 CMRTRUE2	MAINVIEW for CICS task-related user exit program

Table 11. Release Dependent Programs (Continued)

Program	Description
CMRROLL6 CMRROLL9 CMRROLL0 CMRROLL1 CMRROLL2	MAINVIEW for CICS statistics output program driver
CMRSLOG6 CMRSLOG9 CMRSLOG0 CMRSLOG1 CMRSLOG2	MAINVIEW for CICS statistics gathering program
CMRXEIO6 CMRXEIO9 CMRXEIO0 CMRXEIO1 CMRXEIO2	MAINVIEW for CICS task kill global user exit program
OLTSCMR6 OLTSCMR9 OLTSCMR0 OLTSCMR1 OLTSCMR2	Start and stop MAINVIEW for CICS data collection
OLTVCMR6 OLTVCMR9 OLTVCMR0 OLTVCMR1 OLTVCMR2	Test OLTP release compatibility (MAINVIEW for CICS)

Appendix D. 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 OS/390 console which 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 AutoOPERATOR Customization Guide* when implementing MAINVIEW AutoOPERATOR for CICS **only**
- “Standard Implementation Procedures” in the MAINVIEW for CICS Customization Guide (when implementing MAINVIEW for CICS or AutoOPERATOR for CICS or both)

To start BBI-SS to CICS communication from a CICS terminal, invoke the SMN2 transaction, then type YES in the ESTABLISH COMMUNICATIONS field and press ENTER.

To start BBI-SS to CICS communication from an OS/390 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
```

invokes the FST2 transaction in the CICS region with the OS/390 job name of CICSProd.

Appendix E. BBPARM Data Set Members for AutoOPERATOR

This section describes BBPARM data set members that you can use as samples to set up and customize AutoOPERATOR.

Table 12 describes all the AutoOPERATOR-specific BBPARM members. Documentation for other BBPARM members (such as BBISSP00) or that are shared among several products can be found in the *MAINVIEW Common Customization Guide*.

Table 12. BBPARM Data Set Members: AutoOPERATOR Only

BBPARM Member Name	Description
AAOALS00	This member defines limits for extended storage for AutoOPERATOR ALERTs and ALERT parameters on your system.
AAOALT00	This member is used to define the VTAM connections for sending ALERTs to COMMAND/POST workstations
AAOARP00	This member is used by the AutoOPERATOR Automation Reporter application to activate and control collection intervals for offline data collection.
AAOEWD00	This is an Elan Workstation parameter member.
AAOEXP00	This member defines thread values and high-priority EXEC names and limits the CPU time that EXECs can use.
AAOGME00	This member is used by AutoOPERATOR to create General Messages Exchange (GME) connections to allow applications to communicate.
AAOMQL00	This member is used to specify which MQSeries queues will be made eligible for automation.
AAONCD00	This member is used for the AutoOPERATOR ACCESS NV option.
AAOPRM00	This member defines Operator Request parameters for the Operator Request application.
AAORUL00	This member contains Rule Set definitions.
AAOTRN00	This member contains the identification of IMS transactions that can be submitted from an AutoOPERATOR EXEC or from a terminal session.
AAOTSP00	This member is used for the TapeSHARE for AutoOPERATOR option.
BBIVAR00	This member defines any number of pre-set 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.

BBPARM Member AAOALS00 Parameters (AutoOPERATOR Only)

BBPARM member AAOALS00 is used by all AutoOPERATOR products to define limits for extended storage for ALERTs on your system in BBPARM member AAOALS00.

For documentation about how to use the ALERTs application, refer to the *MAINVIEW AutoOPERATOR Basic Automation Guide*; for documentation about how to use the AutoOPERATOR application Dynamic Parameter Manager to update the parameters in AAOALS00, refer to the *AutoOPERATOR Customization Guide*.

The following table lists the parameters for AAOALS00.

Table 13. BBPARM Member AAOALS00 - AutoOPERATOR Only

Parameter	Description
MAXSTOR= <u>0</u> nnnnn	<p>Is the maximum amount of BBI-SS PAS private virtual storage to be allocated to ALERTs.</p> <p>Storage is allocated in kilobytes. The default is 0; this means that 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.</p> <p>For example, a value of 2000 K means you can store about 7800 ALERTs. Use the AutoOPERATOR ALERT Statistics display to estimate the average size of an ALERT on your system.</p> <p>Possible values range from 0 to 99999. A value of 0 means that no checking is done.</p>
WARNLVL1= <u>60</u> nn	<p>Is the first warning threshold level for queued ALERTs when the MAXSTOR parameter is set to a value greater than 0 K.</p> <p>The default is 60%. This means that if MAXSTOR=1000, the warning message (AU6100W) is issued when the amount of virtual storage allocated to queued ALERTs reaches 600K.</p> <p>Possible values range from 0 to 99.</p>
WARNLVL2= <u>75</u> nn	<p>Is the second warning threshold level for queued ALERTs when the MAXSTOR parameter is set to a value greater than 0 K.</p> <p>The default is 75%. This means that if MAXSTOR=1000, the warning message (AU6100W) is issued when the amount of virtual storage allocated to queued ALERTs reaches 750K.</p> <p>Possible values range from 0 to 99.</p> <p>Note: Important ALERTs are not generated when ALERT storage usage reaches 100%.</p>

Table 13. BBPARM Member AAOALS00 - AutoOPERATOR Only (Continued)

Parameter	Description
ALERTNV=YES NO	<p>Allows you to specify that all created ALERTs will have a default value where either all ALERTs are set to be retained across BBI-SS PAS restarts and OS/390 IPLs or not. ALERTs that are retained (or non-volatile) are written and saved to disk.</p> <p>This parameter is equivalent to the IMFEXEC ALERT parameter RETAIN(YES NO). Therefore, setting the ALERTNV parameter to ALERTNV=NO would mean that all ALERTs would have a default RETAIN setting of RETAIN(NO).</p> <p>Setting the ALERTNV parameter to ALERTNV=YES would mean that all ALERTs would have a default RETAIN setting of RETAIN(YES).</p> <p>The default is NO.</p> <p>Note: Regardless of which setting you specify for the ALERTNV parameter, the setting can be overridden when you create a specific ALERT, where RETAIN=YES or RETAIN=NO is explicitly set.</p> <p>Changing the ALERTNV setting takes effect only after a BBI-SS PAS warm or cold start.</p>
PUBLISH=ADD REPLACE NO	<p>Sets the default for the PUBLISH parameter when creating/deleting AutoOPERATOR ALERTs.</p> <p>Specifies whether AutoOPERATOR should (by default) send ALERTs to PATROL Enterprise Manager (EM) consoles through the Generalized Message Exchange (GME).</p> <p>Possible values are:</p> <p>ADD</p> <ul style="list-style-type: none"> • Send ALERTs but do not override previously published ALERTs <p>REPLACE</p> <ul style="list-style-type: none"> • Send ALERTs but delete previously published ALERTs that had the same Key and Queue before new ALERT is published <p>NO</p> <ul style="list-style-type: none"> • Do not send ALERTs <p>The default is ADD.</p>

BBPARM Member AAOALT00 Parameters (AutoOPERATOR Only)

BBPARM member AAOALT00 is used to define the VTAM connections for sending ALERTs to COMMAND/POST workstations. The following table lists the parameters that can be used to define the destination and level of diagnostics to be issued.

Table 14. BBPARM Member AAOALT00 - AutoOPERATOR Only

Parameter	Description
APPLID=name	Defines the value of the APPLID keyword on the ACB used by OPEN in establishing the session with VTAM. This must be the same name as the one entered in the VTAMLIST entry as the name of the application.
DEST=device	Defines a VTAM destination for the ALERTs. This VTAM destination (a COMMAND/POST workstation) must be defined to the network as a 3287 printer.
STATUS= <u>ON</u> OFF	Optional. Defines whether the destination is to be enabled (ON) during initialization, or left disconnected (OFF). The default is ON.
SYNCINTV= <u>30</u> nnn	Optional. Defines the synch-up interval between AutoOPERATOR and the workstation. At this defined interval, all current ALERTs are retransmitted to the workstation to compensate for any earlier lost or misprocessed transmissions. The default is 30 minutes.
MSGLVL= <u>WARN</u> msgtype	<p>Optional. Defines which messages are to be written to the journal; where WARN is the default and msgtype can be:</p> <ul style="list-style-type: none"> • NONE No messages to Journal • SEVERE No VTAM connection, internal errors • ERROR Lost or dropped connection, error limit reached • WARN No BBPARM member, termination request • INFORMATIONAL Normal start and stop messages • DEBUG Echo all messages in the Journal

BBPARM Member AAOARP00 Parameters (AutoOPERATOR Only)

BBPARM member AAOARP00 is used by the AutoOPERATOR Automation Reporter application to activate and control the collection intervals for the offline data collection portion of the Automation Reporter.

For documentation about how to use the Automation Reporter, refer to the *MAINVIEW AutoOPERATOR Basic Automation Guide*; for documentation about using the AutoOPERATOR application Dynamic Parameter Manager to update the AAOARP00 member, refer to the *AutoOPERATOR Customization Guide*.

The following table lists the parameters for AAOARP00.

Table 15. BBPARM Member AAOARP00 - AutoOPERATOR Only

Parameter	Description
DSN=(data set name)	<p>Is the name of a user-allocated data set where collected data will be offloaded. This data set must be allocated or no data collection will occur.</p> <p>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.</p>
DISP=(<u>SHR</u> MOD)	<p>Is the disposition (shared or modify) of the user-allocated data set. The default is SHR.</p> <p>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.</p>
OFFINTVL=(60 through <u>1440</u>)	<p>Is the data offload interval in minutes. The default is 1440 minutes (once a day).</p> <p>You can specify any value from 60 minutes (once an hour) to 1440 minutes.</p> <p>Note that the greater the frequency of offloading collected data, the faster the user-allocated data set will be filled.</p>
INTVL=(0 <u>10</u> through 1440)	<p>Is the time interval (in minutes) at which data is collected.</p> <p>The default is 10 minutes; specifying 0 means no data collection is performed.</p> <p>The minimum interval is 10 minutes and the maximum is 1440 (once a day).</p>
QUEUE=(AutoOPERATOR ALERT queue name)	<p>Is the 1- to 8-character AutoOPERATOR ALERT queue name to which any ALERTs generated by the Automation Reporter are routed.</p> <p>The default queue name is ARALERT.</p> <p>The queue specified here will be created, if it does not already exist, when the first Automation Reporter ALERT is created.</p>

BBPARM Member AAOEWD00 Parameters (AutoOPERATOR Only)

The following table lists the parameters for initializing BBPARM member AAOEWD00 for AutoOPERATOR Elan.

Table 16. BBPARM Member AAOEWD00 - AutoOPERATOR Only

Parameter	Description
DEBUG=YES NO	Generates debugging messages.
HBINT=30 nnn	<p>Defines the default heartbeat interval.</p> <p>For example:</p> <p>HBINT=xx</p> <p>where xx is the number of seconds between heartbeat messages that are sent from the host SS to the Elan workstation.</p> <p>This parameter is supported for the Elan Workstation only.</p> <p>For more information, see the IMFEXEC command statements in the <i>MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs</i> or the <i>MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs</i></p>
NTINT=70 nnn	<p>Defines the number of seconds for the Notify time out interval.</p> <p>For example:</p> <p>NTINT=xxx</p> <p>where xxx is the number of seconds for the Notify timeout interval. The default is 70 seconds.</p> <p>This value is set once when the connection between AutoOPERATOR and MAINVIEW AutoOPERATOR/Links is made.</p> <p>This parameter is supported for the MAINVIEW AutoOPERATOR/Links only.</p> <p>For more information, see the IMFEXEC command statements in the <i>MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs</i> or the <i>MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs</i>.</p>

BBPARM Member AAOEXP00 Parameters (AutoOPERATOR Only)

BBPARM member AAOEXP00 is used by all AutoOPERATOR products to define thread values, priority EXEC names, EXEC loop parameters, data set allocation, and abend count parameters.

For documentation about AutoOPERATOR EXECs, refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*; for documentation about using the AutoOPERATOR application Dynamic Parameter Manager to update the parameters in AAOEXP00, refer to the *MAINVIEW AutoOPERATOR Customization Guide*.

The following table lists the parameters for AAOEXP00.

Table 17. BBPARM Member AAOEXP00 - AutoOPERATOR Only

Parameter	Description
AUDITMSG= <u>YES</u> NO	Specifies if 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.
EMABCNT= <u>5</u> nn	<p>Is the maximum number of abends that the EXEC Manager can sustain before it shuts down. To try to restart the EXEC Manager, issue the command . S COMP=EM. If this does not work, you must restart the BBI-SS PAS.</p> <p>When the EXEC Manager stops, no EXECs can run in the system. Valid values range from 1 to 99. The default is 5.</p> <p>Use this threshold to avoid recursive or persistent abends that demand attention.</p> <p>Abends in individual EXECs do not count against this threshold. The count is cumulative from the time the target BBI-SS PAS is started.</p>
EXABCNT= <u>5</u> nn	<p>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.</p> <p>For example, if you specify EXABCNT=10 and an EXEC abends 10 times, the EXEC is disabled after the tenth abend.</p>
EXEC=name	Specifies the names of EXECs that are to be executed as high priority EXECs. A high priority EXEC is any EXEC listed with the EXEC= keyword. This parameter can be repeated.
MAXTPUT= <u>0</u> nnnnn	<p>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. A value of 0 means that no checking is done.</p> <p>TPUTs for EXECs scheduled with the WAIT=YES parameter are counted separately from the scheduling EXEC.</p>
MAXHIGH= <u>5</u> nn	Maximum high priority EXECs to execute concurrently. A high priority EXEC is any EXEC listed with the EXEC= keyword.

Table 17. BBPARM Member AAOEXP00 - AutoOPERATOR Only (Continued)

Parameter	Description
MAXHIGHQ= <u>0</u> 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. A value of 0 means that no checking is done .
MAXNORM= <u>1</u> nn	Maximum normal priority EXECs to execute concurrently. A normal priority EXEC is any EXEC not in the high priority list.
MAXNORMQ= <u>0</u> 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. A value of 0 means that no checking is done .
OSPINUM= <u>0</u> nnn	Maximum number of OSPI ACBs defined. If 0, the request for a session will be aborted upon encountering the first ACB which cannot be opened due to a reason other than that it is in use. Specifying the correct number of ACB definitions here helps bypass problems where ACBs have become unusable due to some other reason.
OSPIPRFX= <u>OSPI</u> name	A four-character prefix to use when selecting an OSPI ACB from the pool of available ACBs.
PEREXLIM= <u>00</u> nn	<p>Specifies an optional CPU percentage usage for an EXEC. If an EXEC's CPU usage exceeds this percentage during any interval of 15 seconds after the EXEC begins, the EXEC is terminated with a U3001 abend code. With the default value of 0, no percentage limit is assigned.</p> <p>The CPU percentage is calculated based on the total CPU time available on 1 CPU within that 15 second interval. For example, 20% means 20% of 15 seconds. If the CPU time exceeds 3 seconds with any given 15 second interval, the EXEC will be terminated.</p>
PREFIX= <u>SSID</u> name	Specifies a prefix to be used when allocating a dataset in an EXEC if the data set name is not enclosed within quotes. This prefix will also serve as the user ID associated with jobs submitted by EXECs. If not specified, the default will be the same as the BBI-SS PAS ID.

Table 17. BBPARM Member AAOEXP00 - AutoOPERATOR Only (Continued)

Parameter	Description
REXX= <u>YES</u> NO	<p>NO Only CLIST is supported for EXECs.</p> <p>YES Both CLIST and REXX are supported for EXECs.</p> <p>The default is REXX=YES.</p> <p>Restrictions: When REXX=YES is specified, the number of EXECs that can be processed concurrently is limited by the MAXHIGH and MAXNORM parameter settings and limited by the number of environment blocks assembled in the module IRXANCHR.</p> <p>You can reset this number (and then reassemble and relink the module). The default number of environment blocks for IRXANCHR is 40. The number of EXECs is arrived at by subtracting 3 from the number of environment blocks and dividing by 2; for example:</p> $(x - 3) / 2$ <p>where x is the number of environment blocks.</p> <p>During AutoOPERATOR start-up, AutoOPERATOR checks to see if REXX=YES and checks the settings of MAXNORM and MAXHIGH. If the sum of MAXNORM and MAXHIGH exceeds the number of environment blocks in module IRXANCHR, a warning message is issued.</p> <p>Note: When changing from REXX=NO to REXX=YES, you must restart the subsystem.</p>
SELLIM= <u>0</u> nn	<p>Is the maximum number of active EXECs on a single thread. Valid values range from 0 to 99. The default is 0. A value of 0 means that no checking is done.</p> <p>This parameter is designed to control recursive calls (where EXECA schedules EXECB, that schedules EXECC, and so on) that run out of control, potentially filling up the private storage in the BBI-SS PAS and causing automation to stop.</p>
SUBXAUTH= <u>DEFER</u> JOBNAME RACF USER	<p>DEFER The decision for using JOBNAME or user ID for the security check is deferred to the batch job (IMFSUBEX) execution time.</p> <p>JOBNAME The JOBNAME is used for the IMFSUBEX security checks. This parameter overrides the origin specified on the batch job.</p> <p>RACF The user ID is used for the IMFSUBEX security checks. This parameter overrides the origin specified on the batch job.</p> <p>USER Same as RACF.</p>
TIMEXLIM= <u>00</u> nnnn	<p>Specifies an optional CPU time limit for an EXEC. If the time limit is exceeded, the EXEC is terminated with a U3001 abend code. With the default value of 0, no time limit is assigned.</p>

Table 17. BBPARM Member AAOEXP00 - AutoOPERATOR Only (Continued)

Parameter	Description
TSOTIME= <u>1200</u> nnnn	Specifies the time out value for attached TSO commands in seconds. A value of 0 disables timing.
UCPARMS= <u>YES</u> NO	Specifies whether EXEC parameters are translated to uppercase. This parameter is important for rule-initiated EXECs and JRNLEXEC=YES.
UNITNAME= <u>SYSALLDA</u> name	Specifies an esoteric unit name to be used when you allocate a data set in an EXEC and the unit parameter is omitted. If not specified, the default shown is used.
WARNLVL1= <u>60</u> nn	<p>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%.</p> <p>Use this threshold to set the first percentage that triggers sending a warning message.</p>
WARNLVL2= <u>75</u> nn	<p>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%.</p> <p>Use this threshold to set a second percentage that triggers sending a warning message.</p>

BBPARM Member AAOGME00 Parameters (AutoOPERATOR Only)

BBPARM member AAOGME00 is used by the AutoOPERATOR General Messages Exchange (GME) to identify which GME nodes receive messages from and send messages to AutoOPERATOR, define the local GME node, and control which instrumentation events are traced.

The following table lists the parameters for AAOGME00

Table 18. BBPARM Member AAOGME00 - AutoOPERATOR Only

Parameter	Description
DEBUGMSG= <u>YES</u> <u>NO</u>	<p>Is an optional parameter that specifies whether to display GME debug messages in the BBI journal and in WTO format.</p> <p>Valid values are YES and NO. The default value is NO.</p>
GMEACT= <u>YES</u> NO	<p>Used in conjunction with the RESET PARM AAOGMEn, RESYNC RECYCLE command to activate or deactivate the GME facility.</p> <p>GMEACT must be specified as YES to use the AutoOPERATOR PATROL EM extension or Command MQ On Ramp.</p> <p>Valid values are YES and NO. The default value is YES.</p>
HOSTNAME=(<u>subsystem id of the BBI-SS PAS</u>)	<p>Is the name of the local GME node and is used to identify this node to other nodes.</p> <p>The default value is the SSID of the BBI-SS PAS. The value can be comprised of numeric and alphabetic characters, underscores (_), and periods (.), but the name <i>must</i> begin with an alphabetic character. The value cannot contain blanks. The maximum length is 48 characters.</p>
LSTNPORT= <u>7051</u> nnnnn	<p>Identifies the listen port number for Web server functions.</p> <p>Client applications, such as Command MQ On Ramp, use this port number to connect to this GME node.</p> <p>Valid values are 0-65535. The default value is 7051.</p>
TCPJOB= <u>*</u>	<p>Not applicable for AutoOPERATOR version 6.1.</p> <p>For AutoOPERATOR version 6.1, the TCP/IP job name is set from the TCPNAME parameter of BBPARM member BBTTCP00. If the parameter is omitted, the BBI-SS PAS attempts to determine the TCP/IP address space to use. If more than one address space is found, the first one found with the latest release is used.</p> <p>For more information about BBPARM member BBTTCP00, refer to the <i>MAINVIEW Common Customization Guide</i>.</p>
TGTHB= <u>0</u> nnn	<p>Is the remote GME node's heartbeat interval in minutes. If this amount of time passes without any messages being received, the local node closes the connection.</p> <p>Valid values are 0-999. The default value is 0, which disables the connection checking.</p>

Table 18. BBPARM Member AAOGME00 - AutoOPERATOR Only (Continued)

Parameter	Description
TGTIP=(network address)	<p>Is the IP address or hostname where the applications to be contacted reside. It marks the start of a group of remote node parameters in the form “TGT*”.</p> <p>The address may be in IPv4, IPv6, or domain name format.</p>
TGTMXMSL= <u>32768</u> nnnnn	<p>Is the maximum message length accepted from the remote GME node.</p> <p>Valid values are 0-32768. The default value is 32768.</p>
TGTNAME=name	<p>Is the remote GME node’s name. It is used to verify that a connection is made to the correct GME node. Typically, this will be the workstation ID of the system where the remote GME node is running.</p> <p>The value can be comprised of numeric and alphabetic characters, underscores (_), and periods (.), and contains no blanks. The maximum length is 48 characters.</p>
TGTPORT= <u>7051</u> nnnnn	<p>Is the remote GME node’s listen port number to connect to.</p> <p>Valid values are 0-65535. The default value is 7051.</p>
TGTRTC= <u>9999</u> nnnn	<p>Is the number of connection retries allowed if a failure occurs.</p> <p>Valid values are 0-9999. A value of 0 disables the reconnection processing. A value of 9999 instructs GME to attempt to reconnect until it is successful. The default value is 9999.</p> <p>Certain platforms have a TCP/IP parameter that delays the closing of TCP/IP connections. In Solaris, this parameter is named <code>tcp_close_wait_interval</code>. GME/MVS cannot reconnect while the connection is in this <code>close_wait</code> state, because <code>connectAAO</code> sees it as a duplicate connection. For information regarding this parameter, refer to your platform-specific documentation.</p>
TGTRTI= <u>3</u> nnn	<p>Is the interval in minutes between connection retry attempts.</p> <p>Valid values are 0-999. A value of 0 disables the reconnection attempt. The default value is 3.</p> <p>Certain platforms have a TCP/IP parameter that delays the closing of TCP/IP connections. In Solaris, this parameter is named <code>tcp_close_wait_interval</code>. GME/MVS cannot reconnect while the connection is in this <code>close_wait</code> state, because <code>connectAAO</code> sees it as a duplicate connection. For information regarding this parameter, refer to your platform-specific documentation.</p>
TRACEAPP= <u>ERROR</u>	<p>Is the minimum severity of application trace records to create.</p> <p>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.</p>
TRACEBUF= <u>50</u> nnnnn	<p>Determines the amount of storage, in thousands of bytes, to allocate for the in-storage wrap-around trace buffer.</p> <p>The maximum value is 99999. The default value is 50.</p>

Table 18. BBPARM Member AAOGME00 - AutoOPERATOR Only (Continued)

Parameter	Description
TRACEGME= <u>ERROR</u>	<p>Is the minimum severity of GME trace records to create.</p> <p>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.</p>
TRACELK=YES <u>NO</u>	<p>Specifies whether to perform a DNS lookup when logging IP addresses in the Trace Table.</p> <p>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.</p>
TRACESEC= <u>WARN</u>	<p>Is the minimum severity of security trace records to create.</p> <p>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.</p>
TRACEZN= <u>LOCAL</u>	<p>Is the timezone to use for timestamps in the Trace Table.</p> <p>Valid values are GMT and LOCAL. The default value is LOCAL.</p>

BBPARM Member AAOMQL00 (AutoOPERATOR Only)

Use the BBPARM member AAOMQL00 parameters TYPE, QMGR, and QUEUE to specify MQSeries queues that will be made 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)

Table 19. BBPARM Member AAOMQL00 - AutoOPERATOR Only

Parameter	Description
TYPE T (<u>INCL</u> EXCL)	<p>Specifies whether <i>MAINVIEW AutoOPERATOR for MQSeries Installation and User Guide</i> should include or exclude this queue in the set of queues that are eligible for automation. Possible values are:</p> <p>INCL I Specifies that AutoOPERATOR should make this queue eligible for possible automation. This is the default setting.</p> <p> Its abbreviation is I.</p> <p>EXCL E Specifies that AutoOPERATOR should not make this queue eligible for automation.</p> <p> Its abbreviation is E.</p> <p>Abbreviations can be used; for example you can specify any of the following: T(I) or TYPE(I) or TYPE(INCL)</p> <p>The default is INCL I.</p>
QMGR M(queue manager name)	<p>Specifies the four-character ID for a local OS/390 queue manager that AutoOPERATOR will monitor. The wildcard characters asterisk (*) and plus (+) are supported.</p> <p>This is a required parameter and there is no default value. <i>You must specify a queue manager name</i>, or a partial queue manager name with wildcards (a plus (+) represents one character and an asterisk (*) represents one or more characters).</p> <p>QMGR can be abbreviated to M; for example you can sepcify: QMGR(CSQ1) or M(CSQ1)</p>
QUEUE U(queue name)	<p>Specifies the 48-character name of the queue that AutoOPERATOR will make eligible for automation. The wildcard characters asterisk (*) and plus (+) are supported.</p> <p>This is a required parameter and there is no default value. <i>You must specify a queue name</i>, or a partial queue name with wildcards (a plus (+) represents one character and an asterisk (*) represents one or more characters).</p> <p>QUEUE can be abbreviated to U; for example you can specify: QUEUE(SYSTEM.ADMIN.QMGR.EVENT) or U(SYSTEM.ADMIN.QMGR.EVENT)</p>

Table 19. BBPARM Member AAOMQL00 - AutoOPERATOR Only

Parameter	Description
<p>OPEN O(EXCL <u>SHARED</u> PROCESSOLD IGNOREOLD)</p>	<p>Specifies how a queue is opened and what happens to the messages in the queue after it is opened. Possible values are as follows:</p> <p>EXCLUSIVE E Specifies the queue should be opened with the MQOO_INPUT_EXCLUSIVE open option. This means other applications will be unable to open the queue while AutoOPERATOR has it open. It also means that if another application has the queue opened already, AutoOPERATOR will be unable to open it.</p> <p>The abbreviation is E.</p> <p>SHARED S Specifies the queue should be opened with the MQOO_INPUT_SHARED option. This is the default option.</p> <p>The abbreviation is S.</p> <p>PROCESSOLD 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 AutoOPERATOR.</p> <p>The abbreviation is P.</p> <p>IGNOREOLD I Specifies that AutoOPERATOR will not process the existing messages on the queue and no automation will take place for those messages.</p> <p>The abbreviation is I. This option is the default.</p> <p>Abbreviations can be used; for example you can specify O(E,P) which is equivalent to specifying OPEN(EXCLUSIVE,PROCESSOLD).</p> <p>Example 1: OPEN(EXCLUSIVE,PROCESSOLD) or O(E,P)</p> <p>Indicates the queue is to be opened with the MQOO_INPUT_EXCLUSIVE option and all existing messages found on the queue at open time should be routed through the rule processor for automation.</p> <p>Example 2: OPEN(SHARED,IGNOREOLD) OR O(S,I)</p> <p>Indicates the queue is to be opened with the MQOO_INPUT_SHARED option and all existing messages found on the queue at open time should be ignored.</p>

BBPARM Member AAONCD00 Parameters (AutoOPERATOR Only)

BBPARM member AAONCD00 is the Access NV BBPARM member. The following table lists the parameter supported by AAONCD00.

Table 20. BBPARM Member AAONCD00 - AutoOPERATOR Only

Parameter	Description
DEBUG=YES <u>NO</u>	The only parameter supported. For normal operations, leave this parameter to the default setting of NO.

BBPARM Member AAOPRM00 Parameters (AutoOPERATOR Only)

BBPARM member AAOPRM00 is used by all AutoOPERATOR products as a general purpose member for defining Operator Request parameters in BBPARM member AAOPRM00.

The following table lists the parameters for AAOPRM00.

Table 21. BBPARM Member AAOPRM00 - AutoOPERATOR Only

Parameter	Description
HONORMPF=[YES NO]	Specifies whether MPF suppression is turned on (YES) or off (NO). HONORMPF determines whether or not messages will be handled by the Rule Processor when the MPF suppression flag is set.
MPFFILTR=(AUTO= YES NO token,SUP= YES NO)	Specifies whether or not AutoOPERATOR will check WTORs for the MPF settings for the AUTO and SUP keywords. 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 75.
RULESCAN=F Q A I	Equals an automation strategy of F (FIRST), Q (QUALIFIED), A (ALL), or I (INDIVIDUAL) to determine how Rule Sets and Rules are searched to match events. F(irst) Indicates only the first Rule with matching selection criteria is fired. Q(ualified) Indicates that only most qualified Rules with matching selection criteria are fired. A(All) Indicates Rules with matching selection criteria are fired. I(ndividual) Indicates that each Rule Set can have its own (individual) automation strategy. Each Rule Set can be set with First, Qualified, or All. The automation strategy specified here is used when the BBI-SS PAS is either warm or cold started.
RULESET=xx or RULESET=RULxxxx	RULESET can specify one or more RULESETs to become active upon initialization of the BBI-SS PAS. Format of the statement is RULESET=xx for one, or RULESET=(xx, yy, . . . , zz) for multiple sets. The two-character code is appended to AAORUL. You also can use the 5-character suffix (where the prefix is RUL); for example, RULESET=RULxxxxx or RULESET=(RULxxxxx, RULyyyyy, RULzzzzz) . You can specify a long list of Rule Set names by continuing to the next line. Separate each Rule Set name with a comma.

Table 21. BBPARM Member AAOPRM00 - AutoOPERATOR Only (Continued)

Parameter	Description
ORDESC=(1,2,11)	<p>WTO descriptor codes for messages eligible for the Operator Request application.</p> <p>To not check descriptor codes (or, to accept all messages), specify ORDESC=0. The default is 1,2,11.</p> <p>Note: Messages must meet both ORROUT and ORDESC criteria to be accepted by Operator Request.</p> <p>Possible values are 0 through 16.</p>
ORROUT=0	<p>WTO route codes for messages eligible for the Operator Request application. The default is 0 and specifies the OR application to accept all messages regardless of the route code. A setting of 2,5 specifies that the OR application accepts messages of a 2 route code or a 5 route code.</p> <p>Note: Messages must meet both ORROUT and ORDESC criteria to be accepted by Operator Request.</p> <p>Possible values are 0 through 128.</p>
SUPREPLY= <u>NO</u> YES	<p>Suppresses commands issued as replies to WTORs (write-to-operator with reply).</p> <p>Specifying YES changes the text of the command issued by a terminal session (TS) user or EXEC to For example, if an EXEC issues the command:</p> <pre>R 33, XYZ</pre> <p>the BBI-SS PAS Journal shows the command as:</p> <pre>R 33, *SUPPRESSED*</pre> <p>Therefore, specify YES when you want to suppress the appearance of some text in the BBI-SS PAS Journal (such as the use of a password). The local Journal will show the original message.</p> <p>The default is NO.</p>
MQEV=YES <u>NO</u>	<p>Specifies that <i>MAINVIEW AutoOPERATOR for MQSeries Installation and User Guide</i> should automatically enable instrumentation events for a queue manager during connection if it is not already enabled.</p> <p>The default value is NO.</p>

Table 21. BBPARM Member AAOPRM00 - AutoOPERATOR Only (Continued)

Parameter	Description
MQGINHIB=xxxxxxx <u>JRNL</u>	<p>Specifies the action AutoOPERATOR should take when AutoOPERATOR attempts to listen to a queue which is defined as GET(DISABLED). Possible settings are</p> <p>JRNL Issue a message to the BBI Journal stating that AutoOPERATOR cannot listen to the queue.</p> <p>WTO Issue a write-to-operator (WTO) message stating that AutoOPERATOR cannot listen to the queue.</p> <p>IGNORE Take no action.</p> <p>ALTER Alter the queue to GET(ENABLED).</p> <p>The default value is JRNL.</p>
MQNSHARE=xxxxxxx <u>JRNL</u>	<p>Specifies the action AutoOPERATOR should take when AutoOPERATOR attempts to listen to a queue which is defined as NOSHARE. Possible settings are</p> <p>JRNL Issue a message to the BBI Journal stating that AutoOPERATOR cannot listen to the queue.</p> <p>WTO Issue a write-to-operator (WTO) message stating that AutoOPERATOR cannot listen to the queue.</p> <p>IGNORE Take no action.</p> <p>ALTER Alter the queue to SHARE.</p> <p>The default value is JRNL.</p>
MQEVLPRC=xxxxxxx	<p>Specifies the name of the MQSeries Event Listener PROC.</p> <p>Do not use this parameter if the MQSeries Event Listener is not required. Refer to the <i>MAINVIEW AutoOPERATOR for MQSeries Installation and User Guide</i> for more information about the MQSeries Event Listener and co-existence of AutoOPERATOR and other BMC Software products that require the MQSeries event queues.</p>
ALRMSFX=xx	<p>Specifies the name of the MVALARM member to use if the MVALARM product was initialized in the BBI-SS. For the MVALARM initialization to happen, the PRODUCT=MVA line need to be present in the BBISSPXX member of BBIPARM. Multiple members can be specified in the format (XX,YY,ZZ ...). This parameter is ignored when the MVALARM product is not activated.</p>

BBPARM Members AAORUL00 Parameters (AutoOPERATOR Only)

BBPARM members AAORUL00 contain sets of distributed AutoOPERATOR Rule Sets for use with AutoOPERATOR Rules. These Rule Set names are therefore reserved for use to distribute AutoOPERATOR Rules.

The distributed members include:

Member name	Description
AAORULBA	Contains a set of Rules that can be used to automate the startup and shutdown of IMS, DB2, and CICS.
AAORULBB	Contains a set of Rules that can be used to initialize multiple variables or functions in response to the PM0010I message.
AAORULBC	Contains a starter set of Rules that is used for and by implementation of the AutoOPERATOR application CSM.
AAORULBx	Contains sample Rules for the AutoOPERATOR product where x is a pre-defined variable.
AAORULCS	Contains a starter set of Rules that can be used for the implementation of AutoOPERATOR CSM.
AAORULC1	Contains sample Rules that intercept CICS and MAINVIEW for CICS messages.
AAORULD1	Contains sample Rules for DB2 solutions.
AAORULD2	Contains sample Rules for DB2 solutions.
AAORULD3	Contains sample Rules for DB2 solutions.
AAORULD5	Contains sample Rules for DB2 solutions.
AAORULD6	Contains sample Rules for DB2 solutions.
AAORULD8	Contains sample Rules for DB2 solutions.
AAORULI1	Contains sample Rules that generate ALERTs for IMS messages.
AAORULJV	Contains sample Rules to intercept messages and commands.
AAORULM1	Contains sample Rules for MAINVIEW Sample Solutions.
AAORUL00	Contains sample Rules for AutoOPERATOR Rule Processor application.

For more information about these Rule Set members, refer to the section about Rules in the *MAINVIEW AutoOPERATOR Basic Automation Guide*.

BBPARM Member AAOTRN00 Parameters (AutoOPERATOR Only)

The following table lists the IMS transaction parameters for BBPARM member AAOTRN00.

Table 22. BBPARM Member AAOTRN00 - AutoOPERATOR Only

Parameter	Description
RLTERM= <u>MASTER</u> LTERM name	IMS/AO uses the RLTERM as the input LTERM when inserting a transaction to the IMS queue for processing. The status of this LTERM should be active when an IMS transaction is submitted through the IMS/AO from a terminal session or from an EXEC. If you do not expect a response, set RLTERM to DFSMTCNT, which is always active.
TRAN=transaction name	The transaction names define a list of IMS transactions that can be submitted through the IMS/AO from a terminal session or from an EXEC. A plus sign (+) can be used as a positional qualifier and an asterisk (*) can be used as a generic qualifier. When the plus sign and asterisk are used together, the asterisk must be the last character.

BBPARM Member AAOTSP00 Parameters (AutoOPERATOR Only)

BBPARM member AAOTSP00 is used by the TapeSHARE for AutoOPERATOR component.

For documentation about how to use the TapeSHARE for AutoOPERATOR component, refer to the *MAINVIEW AutoOPERATOR Options User Guide*; for documentation about how to use the AutoOPERATOR application Dynamic Parameter Manager to update the parameters in AAOTSP00, refer to the *MAINVIEW AutoOPERATOR Customization Guide*.

The following table lists the parameters for the AutoOPERATOR component TapeSHARE.

Table 23. BBPARM Member AAOTSP00 - AutoOPERATOR Only

Parameter	Definition
ACTION=[<u>DEFAULT</u> CANCEL WTOR NOHOLD HOLD]	<p>The action TapeSHARE should take when Allocation Failed Event occurs because a successful GIVE cannot be performed.</p> <p>Possible values and their definitions are:</p> <p>DEFAULT Allows the installation's default action to occur</p> <p>CANCEL Cancels the job</p> <p>WTOR Issues a WTOR to the operator</p> <p>If the site does not have a defined default action, this is the default.</p> <p>NOHOLD Allows the job to wait without holding resources</p> <p>HOLD Allows the job to wait while holding resources</p> <p>Example:</p> <p>ACTI ON=WTOR</p>
FREE=[<u>DEALLOC</u> Deallocation procedure name]	<p>For use with MVS SP4.x only.</p> <p>Allows you to specify the 1- to 8-character name of a deallocation procedure which starts after TapeSHARE issues the VARY OFFLINE command to GIVE a device to another image. Scheduling the procedure triggers (de)allocation processing in the system GIVING the device and places the device in an OFFLINE state.</p> <p>Example:</p> <p>FREE=RELEASE</p>

Table 23. BBPARM Member AAOTSP00 - AutoOPERATOR Only (Continued)

Parameter	Definition
<p>NOGIVE=device address</p> <p>Default value is none.</p>	<p>A list of device addresses (or a range of addresses) that should not be GIVEN to other partners within the TapeSHARE PLEX when they request devices to TAKE.</p> <p>Note: This does not prevent you from manually GIVING a NOGIVE device.</p> <p>In other words, a list of device addresses that are dedicated to this image. If a device address is not associated with this parameter, TapeSHARE assumes there are no devices dedicated to this image.</p> <p>Example:</p> <p>NOGIVE=0123</p> <p>or</p> <p>NOGIVE=0120-012F</p> <p>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.</p> <p>For example:</p> <p>NOGIVE=0123 NOTAKE=0123</p> <p>If the device addresses 0123 is specified on both the NOGIVE and NOTAKE parameters, the 0123 is completely excluded from TapeSHARE control. TapeSHARE will not attempt to GIVE or TAKE this device.</p>

Table 23. BBPARM Member AAOTSP00 - AutoOPERATOR Only (Continued)

Parameter	Definition
<p>NOTAKE=device address</p> <p>Default value is none.</p>	<p>A list of device addresses (or a range of addresses) that this image should not TAKE when a request for resources is made.</p> <p>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.</p> <p>Example:</p> <p>NOTAKE=0123</p> <p>or</p> <p>NOTAKE=0120- 012F</p> <p>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.</p> <p>For example:</p> <p>NOGIVE=0123 NOTAKE=0123</p> <p>If the device address 0123 is specified on both the NOGIVE and NOTAKE parameters, the 0123 device is completely excluded from TapeSHARE control. TapeSHARE will not attempt to GIVE or TAKE this device.</p>

Table 23. BBPARM Member AAOTSP00 - AutoOPERATOR Only (Continued)

Parameter	Definition
PARTNER=subsystem name	<p>A BBI-SS PAS name (that can be associated with either one or a range of tape devices) that is a TapeSHARE PLEX partner.</p> <p>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.)</p> <p>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.</p> <p>Example 1:</p> <p>PARTNER=SYSA</p> <p>In this example, SYSA is a partner to this image.</p> <p>Example 2:</p> <p>PARTNER=SYSC, 0123</p> <p>In this example, SYSC is a partner to this image and the device whose address is 0123 cannot be GIVEN to partner SYSC.</p> <p>Example 3:</p> <p>PARTNER=SYSB, 0120- 0127</p> <p>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.</p>

Table 23. BBPARM Member AAOTSP00 - AutoOPERATOR Only (Continued)

Parameter	Definition
<p>PREF=[YES NO FORCE]</p>	<p>Allows you to specify whether or not Preferencing is activated.</p> <p>If activated, TapeSHARE uses the devices identified on the PREFDEV parameter for only the operation (input or output) identified for that device.</p> <p>This parameter must be used in conjunction with the PREFDEV parameter.</p> <p>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 will be GIVEN to SYSA.</p> <p>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.</p> <p>Example:</p> <p>PREF=YES</p> <p>or</p> <p>PREF=NO</p> <p>or</p> <p>PREF=FORCE</p>
<p>PREFDEV=device address</p> <p>A tape device address (or a range of tape device addresses) followed by either an I for input-only or O for output-only; see example</p> <p>Default value is none.</p>	<p>A list of device addresses identified to TapeSHARE on this image that are allocated to perform input-only or output-only functions.</p> <p>This parameter should be used in conjunction with the PREF parameter.</p> <p>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.</p> <p>If a value is not associated with this parameter, TapeSHARE will not invoke device preferencing.</p> <p>Example:</p> <p>PREFDEV=01A0- 01A3, I</p> <p>or</p> <p>PREFDEV=0130, O</p> <p>or</p> <p>PREFDEV=0150- 015F, O</p>

Table 23. BBPARM Member AAOTSP00 - AutoOPERATOR Only (Continued)

Parameter	Definition
RETRYCNT=(0 2 through 10)	<p>The number of times TapeSHARE attempts to satisfy a request for devices after an initial attempt has failed.</p> <p>For example, suppose this system's request for devices is not satisfied and 3 retries is specified. TapeSHARE will attempt 3 more times to satisfy this request for devices.</p> <p>Example:</p> <p>RETRYCNT=3</p> <p>This parameter is used in conjunction with the RETRYINT parameter which specifies how long TapeSHARE waits between retry attempts.</p> <p>Possible values are 0 - 10.</p>
RETRYINT=(0 30 through 300)	<p>The number of seconds TapeSHARE waits for devices between attempts after an initial attempt has failed.</p> <p>For example, suppose:</p> <ul style="list-style-type: none"> • The RETRYCNT= parameter is set to 3 retries • The RETRYINT= parameter is set to 45 seconds <p>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).</p> <p>Should all three attempts fail, an Allocation Failed Event occurs and the specified action is taken.</p> <p>Example:</p> <p>RETRYCNT=3</p> <p>Possible values are 0 - 300.</p>

Table 23. BBPARM Member AAOTSP00 - AutoOPERATOR Only (Continued)

Parameter	Definition
TIMEOUT=(0 <u>120</u> through 999)	<p>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 completed, an Allocation Failed Event occurs unless you have specified a number of retries on the RETRYCNT parameter.</p> <p>Example:</p> <p>TIMEOUT=20</p> <p>Possible values are 0 - 999.</p>
TRACE=[YES <u>NO</u>]	<p>Writes diagnostic information to BBITST DD statement.</p> <p>Example:</p> <p>TRACE=NO</p> <p>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).</p>

BBPARM Member BBIVAR00 Parameters (AutoOPERATOR Only)

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.

Table 24. BBPARM Member BBIVAR00 - AutoOPERATOR Only

Parameter	Description
MYVAR999=ACTIVE MYVAR999 = active (Example settings)	This member contains any number of variables that are to be set at BBI-SS PAS startup when VPOOL=RESET is specified with the start up parameters. (VPOOL=RESET requires a COLD start.) These SHARED variables are set before the 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 type Rules.

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 BBI VAR and a suffix. The default suffix is 00. An alternate 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 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


```
BBI VAR=(99)
```

 where 99 is the suffix for BBPARM member BBIVAR99.

To specify multiple BBIVARxx members:

```
BBI VAR=(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 will start up with.

For example, if the BBCFG member contains:

```
BBI VAR=(99, 12)
BBI VAR=(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 may cause the default BBIVAR00 to be processed.

- The BBIVARxx member contains syntax errors.

The following syntax rules apply to processing within the BBIVARxx member:

Any number of entries may be specified.

- Each line is restricted to 72 characters and entries cannot be continued over multiple lines.
- Variable 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 = val ue
```

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.
- 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:

```
A = 0
I. A = 5
```

creates a variable called A and a variable called I.A, not I.O.

- 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 above 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 (Ending quote missing)
```

QSMID = SYSB (Q-variables are reserved and read-only)

Appendix F. Sample Data Set Members

The BBSAMP data set contains sample members you can edit for your site's use.

Sample Members for AutoOPERATOR Only

The following table describes BBSAMP members used only for AutoOPERATOR.

Table 25. BBSAMP Data Set Members for AutoOPERATOR

BBSAMP Member Name	Description
\$EXECJOB	EXEC testing batch job.
\$EXECTST	EXEC testing CLIST.
\$TESTDAT	EXEC testing sample test case.
AOALRTDF	Allows you to define resources to RACF for 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.
AOALRTPE	Allows you to permit users to 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.
AOAPPLDF	Allows you to define all AutoOPERATOR application resources to RACF.
AOAPPLPE	Allows you to permit users to AutoOPERATOR application resources.
AOCMDDF	Allows you to define all AutoOPERATOR command-level resources to RACF.
AOCMDPE	Allows you to permit users to AutoOPERATOR command-level resources.
AOEXECDF	Allows you to define all AutoOPERATOR EXEC name resources to RACF.
AOEXECPE	Allows you to permit users to AutoOPERATOR EXEC name resources.
AOANYDF	Allows you to define all AutoOPERATOR AOAnywhere functions to RACF.
AOANYPE	Allows you to permit users to AutoOPERATOR AOAnywhere functions.
AOPARMDF	Allows you to define all 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.
AOAPPLPE	Allows you to permit users to all 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.
CAOTTAB	Macro used to generate the list of terminal types supported by the AutoOPERATOR for CICS Broadcast application.
CICSTART	Sample EXEC used to automate the start up of the CICS environment.
CICSTERM	Sample EXEC used to automate the shut down of the CICS environment.
DB2START	Sample EXEC used to automate the start up of the DB2 environment.

Table 25. BBSAMP Data Set Members for AutoOPERATOR (Continued)

BBSAMP Member Name	Description
DB2TERM	Sample EXEC used to automate the shut down of the DB2 environment.
DIVDEF	Creates the linear data set used by AutoOPERATOR to store a variety of data, including: non-volatile ALERTs and data recorded by the Automation Reporter. The subsystem must be down prior to submitting this jobstream.
DIVUTIL	This JCL copies the BBIDIV data set to a newly allocated data set or produces a report of space utilization in the BBIDIV.
IMSSTART	Sample EXEC used to automate the start up of the IMS environment.
IMSTERM	Sample EXEC used to automate the shut down of the IMS environment.
DTABJCL	Sample JCL to assemble and link CAODTAB.
JES2MAO	Sample JCL to assemble with user's SYS1.HASPSRC.
NAICMDS	Defines NAIEXEC and NAISTUB to NetView.
NAIINIT	Initializes an Access NV controlled OST.
NAIOPT00	Defines the BBI-SS PAS that communicates with NetView and OSTs used by that BBI-SS PAS.
NAIPROF	Sample OST initialization profile.
NAISVAR	Sets global variables for Access NV tasks.
NAITASK	Defines NATASK as a NetView optional subtask (OPT).
RVARTST	EXEC that shows settings of variables.
TTABJCL	Sample JCL to assemble and link CAOTTAB.

Sample Members for 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 26. BBSAMP Data Set Members for MAINVIEW AutoOPERATOR for CICS and MAINVIEW for CICS

BBSAMP Member Name	Description										
CMRASM	<p>Sample JCL to assemble and link-edit any of the following modules:</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Product</th> </tr> </thead> <tbody> <tr> <td>CMRPRBT</td> <td>MAINVIEW for CICS</td> </tr> <tr> <td>CMRRAPM</td> <td>MAINVIEW for CICS</td> </tr> <tr> <td>CMRSECU</td> <td>MAINVIEW for CICS and AutoOPERATOR for CICS</td> </tr> <tr> <td>CMRSOPT</td> <td>MAINVIEW for CICS</td> </tr> </tbody> </table> <p>Note: Descriptions for each of these modules can be found in this chapter</p>	Name	Product	CMRPRBT	MAINVIEW for CICS	CMRRAPM	MAINVIEW for CICS	CMRSECU	MAINVIEW for CICS and AutoOPERATOR for CICS	CMRSOPT	MAINVIEW for CICS
Name	Product										
CMRPRBT	MAINVIEW for CICS										
CMRRAPM	MAINVIEW for CICS										
CMRSECU	MAINVIEW for CICS and AutoOPERATOR for CICS										
CMRSOPT	MAINVIEW for CICS										
CMRCSDES	Sample JCL containing RDO definitions for CICS/ESA and CICS Transaction Server regions.										
CMRPLT	Macro used to generate valid PLT entries for CICS/ESA and CICS Transaction Server regions.										
CMRSECU	Security for CICS option of AutoOPERATOR and MAINVIEW for CICS.										
CMRUSECU	Sample for defining function security.										

Sample Members for AutoOPERATOR for IMS, MAINVIEW for IMS, and MAINVIEW for DBCTL

Table 27 describes BBSAMP members used only for AutoOPERATOR for IMS, MAINVIEW for IMS, and MAINVIEW for DBCTL.

Table 27. BBSAMP Data Set Members for AutoOPERATOR for IMS, MAINVIEW for IMS, and MAINVIEW for DBCTL

BBSAMP Member Name	Description
ICOPY7	Sample install JCL to copy BMC Software modules to IMS.RESLIB data set available to the IMS 4.1 control region.
ICOPY8	Sample install JCL to copy BMC Software modules to IMS.RESLIB data set available to the IMS 5.1 control region.
ICOPY9	Sample install JCL to copy BMC Software modules to IMS.RESLIB data set available to the IMS 6.1 control region.
IMRUVTF	Provides an interface from MAINVIEW for IMS/AO to BMC Software's 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).
LINKNUC	Sample JCL to link BMC Software AOI exit IBAOUE3 into the IMS nucleus (IMS Release 4.1 and below).

Appendix G. Expanding the PROFILE Pool Data Set

This appendix discusses using the XTLOAD utility which is located in BBSAMP member XTJCL. Use this utility for the data set where PROFILE variables are stored. Currently, when this data set gets full, you must allocate a new, larger data set and then re-initialize the new data set. All PROFILE variables in the old data set are lost.

If you have a large number of variables and do not want to lose them and you also need a 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.

Do not use IEBGENER to copy the smaller dataset into the new, larger dataset. You must use the XTLOAD utility for AutoOPERATOR to recognize the additional space.

One way you will know when the data set for the PROFILE pool is too full is when IMFEXEC VPUT requests to the PROFILE data set (referred to by the BBIVARS DD statement in the BBI-SS PAS JCL) fail with a return code of 20 or when IMFEXEC VPUTL requests fail with a return code of 8.

Using the XTLOAD Utility

To expand the PROFILE pool data set, use the XTLOAD utility. You must bring down AutoOPERATOR before you can use this utility to expand it.

XTLOAD copies the content of the existing BBIVARS data set into a larger data set. After successful execution of XTLOAD, the new, larger data set can be specified on the BBIVARS DD statement. Alternatively, the new data set can be renamed to match the data set name on the BBIVARS DD statement.

Use the following sample JCL to execute XTLOAD.

```
//STEP1 EXEC PGM=XTLOAD
//STEPLIB DD DSN=hl level . ???? . BBLINK, DISP=SHR
//BBIVARS DD DSN=old. bbi vars. dataset, DISP=OLD
//NEWVARS DD DSN=new. bbi vars. dataset,
//          DISP=(NEW, CATLG), UNIT=SYSDA,
//          SPACE=(TRK, (nn))
//SYSPRINT DD SYSOUT=R
```

Figure 36. Sample JCL to Execute XTLOAD

Modify the sample JCL as follows:

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

- 4 The NEWVARS, BBIVARS, or SYSPRINT DD statement is missing.
- 8 The NEWVARS DD data set is not larger than the BBIVARS DD data set.

If the BBIVARS data set is full, the IMFEXEC VPUTL command receives a return code of

- 8 BBIVARS data set is full.

Appendix H. AutoRULE Worksheet

This worksheet has five sections that you must fill in before you attempt to complete the AutoCustomization step, “Run AutoRULE for AutoOPERATOR, Event Management Utility”. Each section corresponds to a panel in AutoCustomization. “Implementing AutoRULE” on page 109 provides documentation that describes the AutoCustomization steps.

Section 1: MPF List Conversion Parameters

Fill in the information for converting an MPFLSTxx member to AutoOPERATOR Rules.

MPF Data Set Name	_____
MPF Suffix	_____
This is the 2-character suffix of the SYS1.PARMLIB member MPFLSTxx.	

Fill in the information in the following table so that AutoRULE can convert all USEREXIT references to AutoRULE records (which are later converted to AutoOPERATOR Rules). For each of the USEREXITs that you want to convert, you can specify one of the following flags:

- SAVE (to indicate that AutoRULE should not convert the MPF record)
- IGNORE (to indicate that AutoRULE should ignore MPF records with that USEREXIT)
- A flag from “Section 2: AutoMate Conversion Parameters” on page 228.

USEREXIT Name	Action Flag

Specify a default action for USEREXITs not explicitly listed in the above table.

Section 2: AutoMate Conversion Parameters

Fill in the information in the following table when you want to convert AutoMate automation to AutoOPERATOR Rules.

AutoMate data set name	_____
AutoMate member names to convert	You can convert up to 72 members at a time. _____

Section 3: Generating Rules

Use this section of the worksheet to help you determine how you want AutoOPERATOR Rules generated. Fill in the Rule Set name 3-character suffix in the following table.

Types of Events	Three-Character Rule Set Name Suffix
Messages starting with A through G	_____
Messages starting with H through L	_____
Messages starting with M through S	_____
Messages starting with T through Z	_____
Journal Events	_____
CICS TDQ Events	_____
IMS Events	_____
Command Events	_____

List the started tasks to exclude from abend notification:

List the user IDs to exclude from abend notification:

Section 4: Creating Rules for Production Job Abends

Use this section of the worksheet to help create a Rule that generates an ALERT when a Production job abends.

Fields	User-Specified Value
Job Name	_____
User ID	_____
RACF Group	_____
Job Class	_____
Accounting Information	_____

Section 5: Creating Rules for Production Job Abends

Use this section of the worksheet to tell AutoRule what to do with ALERTs and other notification. AutoRULE sends notifications to the user IDs you specify. If you specify more than one TSO ID, separate them with spaces.

If There are Problems in this Area....	Send a Notification to this User ID
OS/390	_____
CICS	_____
IMS	_____
DB2	_____
Networking	_____
DASD	_____
Unauthorized Commands	_____
Automation	_____

Use this table to list the names of ALERT queues that you want AutoRULE to send ALERTs to. To specify more than one queue name, separate the queue names with two colons (: :).

If there are Problems in this Area...	Send ALERTs to this ALERT Queue Name
OS/390	_____
CICS	_____
IMS	_____
DB2	_____
Networking	_____
Tape	_____
Batch	_____
All other ALERTs	_____

List any additional parameters here:

- ALERT Target name

- Default Beeper Name

- Default Beeper Information

Describing AutoRULE Flags

This section describes the flag characters that AutoRULE uses while creating AutoOPERATOR Rules. These flags are used during conversion of MPF list automation or AutoMate Rules to AutoOPERATOR Rules.

You can manually edit the KnowledgeBase to include or exclude a flag. The flag must be in lowercase characters and start in column 1. There are three kinds of flags:

- Selection: Tells AutoRULE what to use as the Rule's selection criteria
- Action: Tells AutoRULE what to use as the Rule's action specifications
- Special: Used by AutoRULE for comments and additional data

Selection Flags: The following list indicates the flags for a Rule's selection criteria.

- p** Production jobs only
- t** Started tasks only
- j** Batch jobs only
- u** TSO users only
- g** Message text (including message ID)
- h** WHEN clause on the #h record

Action Flags: The following list indicates the flags for a Rule's actions.

a	Alert
b	Beep a person
c	Command is on #c record
e	EXEC name is on #e record
l	Log message
n	Notify a person
r	Reply is on #r record
s	Suppress
v	Set variable; specified on the #r record
w	ALERT/REWORD; text is on the #w record
x	Reject a command
z	Delete an ALERT; ALERT message ID in on the #z record

Special Flags: The following list indicates special flags and their descriptions.

.	Comment
*	Comment
>	Comment
d	Disabled Rule
%	Event type from here on in current member
#x	For g,h,b,c,e,r,v,w, and y: additional information required by the flag
r	Reply is on #r record
s	Suppress
v	Set variable; specified on the #r record
w	ALERT/REWORD; text is on the #w record
x	Reject a command
z	Delete an ALERT; ALERT message ID in on the #z record

Appendix I. How Product Libraries Should Be Used

Objective

The purpose of this section is to ensure that site changes to customized product libraries are not lost when your site migrates to a new release or applies product maintenance.
--

This section provides information about which product libraries to use when you make changes to a product. It explains how the product libraries are created, what their intended use is, and which libraries to use to make your site's changes.

The types of product libraries are:

- SMP-maintained distributed target libraries

These are created during product installation. The installation procedures are described in the *OS/390 and z/OS Installer Guide*.

- Site-customized product libraries

These are created for you by AutoCustomization, or you can create them manually. The AutoCustomization procedures are described in the *OS/390 and z/OS Installer Guide*. Manual procedures are described in Implementation Guide for BBI-2 products.

- Product-user libraries

A user profile (userid.BBPROF) is created during terminal session initialization if one does not exist already.

Figure 37 shows how the product libraries should be configured.

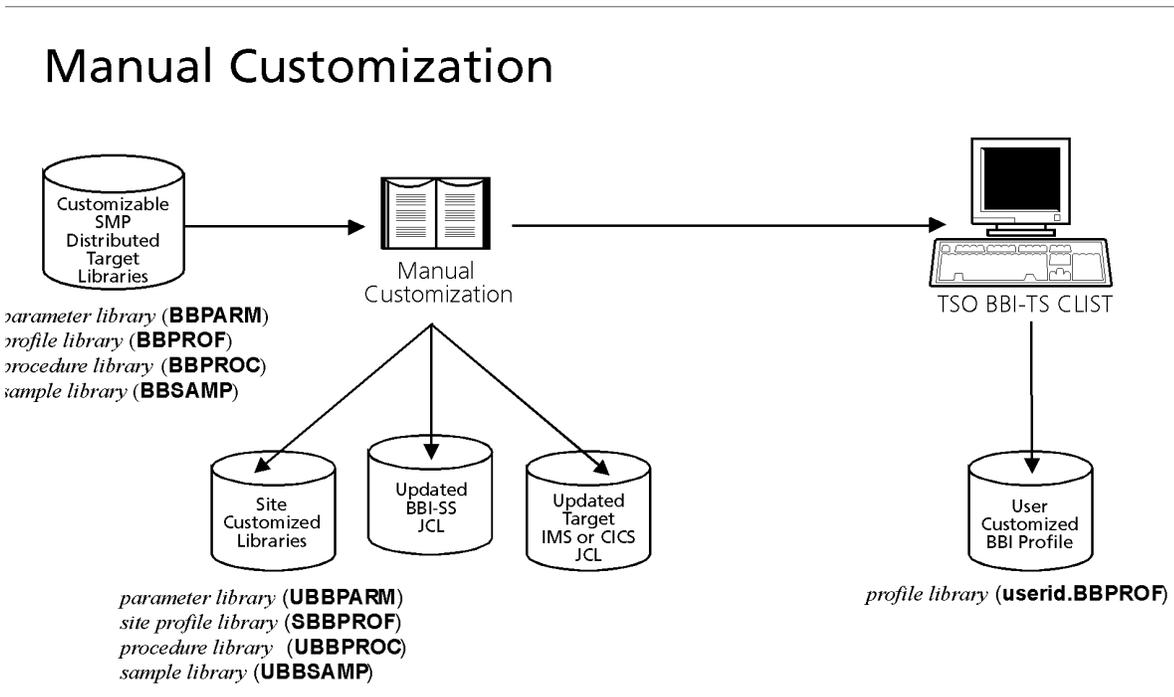
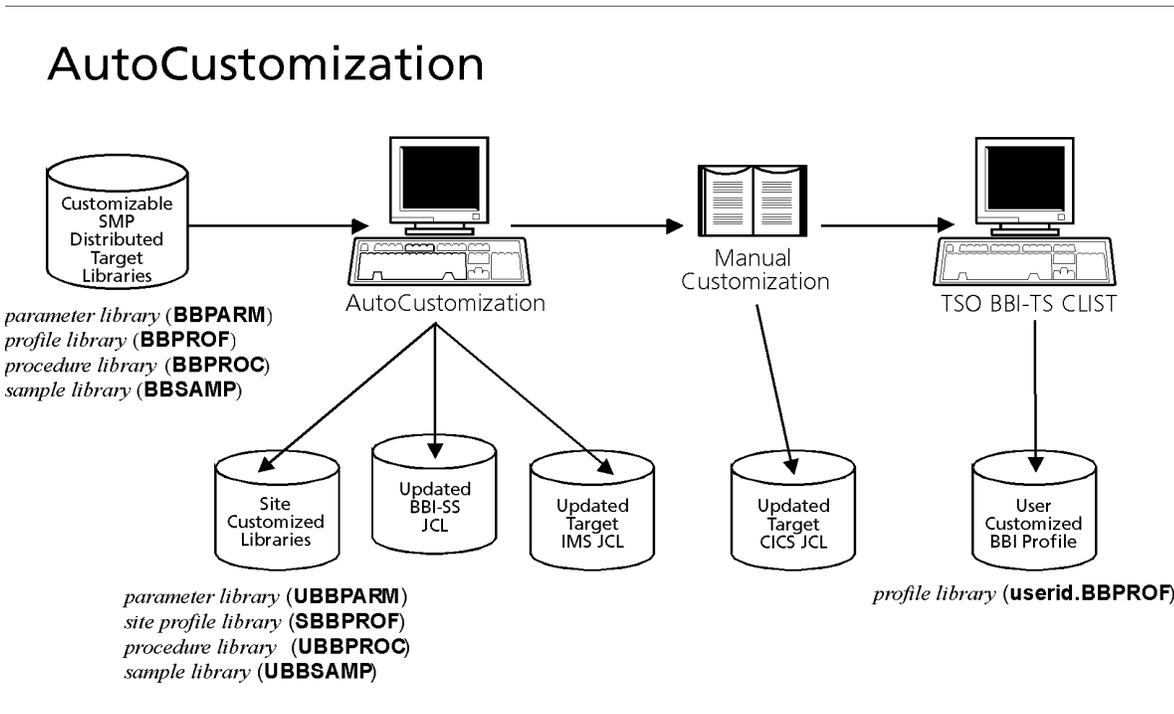


Figure 37. BBI-2 Data Set Customization

The distributed libraries comprise a:

Parameter library	Product parameters Members in this library contain BBI-2 parameter definitions and product-specific parameter definitions.
Sample library	Product samples Members in this library contain macros, sample JCL, and sample user exit routines.
Profile library	Profile information, PF key, and cycle refresh definitions for a terminal session
Procedure library	AutoOPERATOR execute command lists (EXECS)

Use the contents of the distributed libraries as models to create your own site-customized product libraries. The distributed libraries should never be modified. All changes described in this manual should be made only to site-customized product libraries. If you change the distributed libraries, subsequent SMP maintenance will overwrite your changes.

Parameter Library

Each member in this data set contains parameters for a separate product or BBI-2 support component. Your site can have several parameter libraries, a distributed version and one or more site-customized versions. A BBIPARM DD statement can concatenate multiple parameter library data sets. With this technique, a site parameter library set can be created and a data set individualized for each BBI-SS can be concatenated before the site library.

BBPARM

BBPARM is a distributed target library that is created during product installation. This version should never be modified. Subsequent SMP maintenance will overwrite any changes you make. The distributed name is used in this manual for reference only.

UBBPARM

UBBPARM is a copy of the distributed parameter library. It can be created automatically by AutoCustomization, or you can create it manually. Use this copy to make any parameter library changes described in this manual.

AutoCustomization: If you used AutoCustomization successfully when you installed your product libraries as described in the *OS/390 and z/OS Installer Guide*, it created a UBBPARM data set for you and customized its members.

Manual Customization: If you are customizing the product manually:

1. Create your own UBBPARM data set.
2. Copy the members you need from the distributed BBPARM library into the UBBPARM data set you created.
3. Change the copied members for your needs.

IMS-Specific Parameter Library (IMF and AutoOPERATOR for IMS Only)

A parameter library, `i bbparm`, where `i bbparm` represents a user-defined parameter library, may be created for each IMS so that each has its own `IMFSYS00` member plus any other members that you want to make unique to an IMS target. `IMFSYS00` contains `BBI-SS` to IMS communication parameters. The `SUBSYS` communication parameter establishes communication between IMS and the `BBI-SS` for the IMF and AutoOPERATOR for IMS products.

Manual Customization: A parameter library for a specific IMS must be created manually and copied from the distributed parameter library as described in Customization Guide for MVIMS products. Use this copy of the parameter library to make product changes that are to be unique to a specific IMS target. For example, different Event Collector initialization parameters can be defined for a specific IMS by copying the `IMFECPO0` member from the distributed parameter library to your user-defined parameter library.

Sample Library

The members in this data set contain:

- Sample JCL that can be edited and submitted to perform specified functions
- Macros that are referenced when assembling user-written services
- Sample user exit routines

Your site can have several sample libraries, a distributed version and one or more site-customized versions. Some members are for all the `BBI-2` products and some are product-specific.

BBSAMP

`BBSAMP` is a distributed target library that is created during product installation. This version should never be modified. Subsequent SMP maintenance will overwrite any changes you make. The distributed name is used in this manual for reference only.

UBBSAMP

`UBBSAMP` contains copies of members from the distributed sample library. It can be created automatically by AutoCustomization, or you can create it manually. You can use `UBBSAMP` to make any changes to members described in this manual.

AutoCustomization: If you used AutoCustomization successfully when you installed your product libraries as described in the *OS/390 and z/OS Installer Guide*, it created a `UBBSAMP` data set for you if one did not exist already. `UBBSAMP` contains copies of members from the distributed sample library. Use these members to customize a product to your site's needs.

You can use `UBBSAMP` to modify other `BBI-2` product samples. If you need to modify a product-specific sample as described in this manual:

1. Verify that the member was not copied by AutoCustomization to `UBBSAMP`.

2. Copy the sample member you need from the distributed BBSAMP to UBBSAMP.
3. Make the change in the copied member.

Manual Customization: If you are customizing the product manually:

1. Create your own UBBSAMP data set.
2. To change a sample member described in this manual, copy the applicable member(s) you need from the distributed BBSAMP library into the UBBSAMP data set you created.
3. Change the member(s) you copied in UBBSAMP.

Profile Library

Your site can have several profile libraries, a distributed version and one or more site-customized versions. Members in this data set contain profile information and cycle refresh definitions for a BBI-TS. Other members are dynamically created by BBI applications. Do not change any members in this library unless instructed to.

You can have a site profile library and a user profile library. The site library can be created automatically by AutoCustomization, or you can create it manually. The site library is a common profile shared by all site users. The terminal session (BBI-TS) CLIST creates a user profile automatically if one does not exist already. Users should have their own profile library so that each user can specify:

- Unique PF keys
- CYCLE commands
- Target system defaults
- Primary Option Menu
- A unique set of application profiles

The user profile and the site profile should be concatenated before the distributed profile. When a profile is saved, it is stored in the first profile library defined in the concatenation.

BBPROF

BBPROF is a distributed target library that is created during product installation. This version should never be modified. Subsequent SMP maintenance will overwrite any changes you make. The distributed name is used in this manual for reference only.

SBBPROF

SBBPROF is an optional data set. It can be created automatically by AutoCustomization, or you can create it manually. Use SBBPROF to make any changes described in this manual that you want to be shared by all users at your site.

AutoCustomization: If you used AutoCustomization successfully when you installed your product libraries as described in the *OS/390 and z/OS Installer Guide*, it created an SBBPROF data set for you and customized its members.

Manual Customization: If you are customizing the product manually:

1. Create a common SBBPROF data set for your site.
2. Copy the applicable member(s) you need from the distributed BBPROF library into the SBBPROF data set you created.
3. Change the member(s) you copied in SBBPROF.

User BBPROF

There should be a profile data set for each user so that each user has an individual application profile. The terminal session CLIST created a user profile automatically if one did not exist. It is called `userid.BBPROF`, where `userid` is the user's logon ID. This data set contains profile members customized by a user.

Procedure Library (AutoOPERATOR Only)

Your site can have several procedure libraries available, a distributed version and one or more site-customized versions. Members in this data set contain executable procedures used by AutoOPERATOR. These procedures are execute command lists (EXECs) that automate site functions. For more information about EXECs, see the manuals shipped with your AutoOPERATOR product.

BBPROC

BBPROC is a distributed target library that is available when AutoOPERATOR is installed successfully. This version should never be modified. Subsequent SMP maintenance will overwrite any changes you make. The distributed name is used in this manual for reference only.

UBBPROC

UBBPROC is used to contain new user-written EXECs or customized AutoOPERATOR-supplied EXECs from the distributed BBPROC library. It can be created automatically by AutoCustomization for AutoOPERATOR, or you can create it manually.

AutoCustomization: If you used AutoCustomization successfully when you installed AutoOPERATOR as described in the *OS/390 and z/OS Installer Guide*, it created a UBBPROC data set for AutoOPERATOR.

If you need to use a specific EXEC sample:

1. Verify that the member was not copied by AutoCustomization to UBBPROC.
2. Copy the sample member you need to modify from the distributed BBPROC to UBBPROC.
3. Make the change in the copied member.

Manual Customization: If you are customizing AutoOPERATOR manually:

1. Create your own UBBPROC data set.

2. Copy the member(s) you need to modify from the distributed procedure library into the UBBPROC data set you created.
3. Change the member(s) you copied in UBBPROC.

BBI-2 Product Library Use Summary

Table 28 summarizes how the BBI-2 product libraries should be configured.

Table 28. BBI-2 Product Library Configuration

Distributed Library	Customized Library	Created by AutoCustomization?	Allocated to which Address Space?	Allocated to What DD Name?
BBSAMP	UBBSAMP	Yes, AutoCustomization creates UBBSAMP.	None	None
BBSAMP	UBBPARM	Yes, AutoCustomization creates UBBPARM.	BBI-SS CICS target IMS target	BBIPARM
	ibbparm	No, ibbparm is a user-defined library that must be created manually.	IMS target	IMFPARM
BBPROC	UBBPROC	Yes, AutoCustomization creates UBBPROC but only for AutoOPERATOR.	BBI-SS	SYSPROC
BBPROF	user i d.BBP ROF and/or SBBPROF	user i d.BBPROF: No, AutoCustomization does not create a user profile. It is created automatically by the TSO BBI-TS CLIST. A VTAM or EXCP BBI-TS user i d.BBPROF must be created manually. If you are using MAINVIEW Alternate Access for EXCP or VTAM communication, you do not need to create a user i d.BBPROF library. SBBPROF: Yes, AutoCustomization creates a site profile.	BBI-TS	BBIPROF
Concatenation: User and site-customized libraries should be concatenated before the distributed libraries.				

Appendix J. OSPI Terminal Definitions for CICS and IMS

The following sections provide information about defining OSPI virtual terminals to CICS and IMS.

IMS Terminal Definitions

IMS requires all terminals to be defined in the IMS SYSGEN. If an attempt is made to logon with an OSPI ACB that has not been defined to IMS, the session is not accepted. Following is an example of the IMS terminal definitions required for OSPI ACBs:

```
TERMI NAL  MODEL=2,
            FEAT=( , NOCD, NOPEN) ,
            FPBUF=256, OPTI ONS=(TRANRESP) ,
            NAME=OSPI 0000                ** OSPI ACB name **

            NAME  OSPI 0000                ** Any LTERM name **
                                           ** (subject to IMS security) **
```

CICS Terminal Definitions

The terminal definitions required by CICS depend upon 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” on page 241 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” 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 38 on page 242 provides an example of how to use Resource Definition Online to define an OSPI virtual terminal.

RDO Macro

```

OVERTYPE TO MODIFY
CEDA Alter
  TYpeterm      : OSPI
  Group        : TEST
RESOURCE TYPE
DEVi ce       ==> LUTYPE2
TERmmodel     ==> 2
SESSi ontype  ==>
LDcli st      :
SHi ppable    ==> No           No | Yes
MAPPING PROPERTIES
PAGesi ze     ==> 000 , 000   0- 999
ALTPage       ==> 000 , 000   0- 999
ALTSUffi x    ==>
FMhparm       ==> No           No | Yes
OBOperid      ==> No           No | Yes
PAGING PROPERTIES
AUTOPage      ==> No           No | Yes
DEVICE PROPERTIES
DEFscreen     ==> 024 , 080   0- 999
ALTScreen     ==> 000 , 000   0- 999
APLKybd       ==> No           No | Yes
APLText       ==> No           No | Yes
AUDi bl eal arm ==> No           No | Yes
COLor         ==> No           No | Yes
COPy          : No           No | Yes
DUal casekybd ==> No           No | Yes
EXTendedds    ==> No           No | Yes
HI l i ght    ==> No           No | Yes
Katakana      ==> No           No | Yes
LI ghtpen     ==> No           No | Yes
Msrcontrol    ==> No           No | Yes
OBFormat      ==> No           No | Yes
PARTi ti ons  ==> No           No | Yes
PRI ntadapter ==> No           No | Yes
PROgsymbols   ==> No           No | Yes
VALi dati on  ==> No           No | Yes
FOrmfeed      ==> No           No | Yes
HOr i zform   ==> No           No | Yes
VERti cal form ==> No           No | Yes
TEXTKybd      ==> No           No | Yes
TEXTPri nt    ==> No           No | Yes
Query         ==> No           No | Col d | All
OUtl i ne     ==> No           No | Yes
SOSi          ==> No           No | Yes
BACktrans     ==> No           No | Yes
CGcsgi d      ==> 00000 , 00000 0- 65535
SESSI ON PROPERTIES
ASci i        ==> No           No | 7 | 8
SENdsi ze     ==> 00000       0- 30720
RECEi vesi ze ==> 00000       0- 30720
BRacket       : Yes           Yes | No
LOGMode       ==> D6327802

```

Figure 38. Example of Using Resource Definition Online (Part 1 of 2)

```

DIAGNOSTIC DISPLAY
ERRLastline ==> No          No | Yes
ERRIntensify ==> No        No | Yes
ERRColor     ==> NO        NO|Blue|Red|Pink|Green
                                   |Turquoise|Yellow|Neutral
                                   No|Blink|Reverse|Underline

ERRHighlight ==> No

OPERATIONAL PROPERTIES
AUTOConnect ==> No          No | Yes | All
ATi          ==> Yes        No | Yes
TTi          ==> Yes        Yes | No
CCreatesess ==> No          No | Yes
RELreq       ==> No          No | Yes
DIScreq      ==> Yes        Yes | No
Nepclass     ==> 000        0-255
SIGNoff      ==> Yes        Yes | No | Logoff

MESSAGE RECEIVING PROPERTIES
Routedmsgs  ==> All         All | None | Specific
LOGOnmsg    ==> Yes        No | Yes

APPLICATION FEATURES
BUildchain  : Yes          No | Yes
USerarealen ==> 000        0-255
Ioarealen   ==> 02000 , 02000 0-32767
UCtran      ==> Yes        No | Yes

RECOVERY
RECOVOption ==> Sysdefault  Sysdefault | Clearconv | Releases
                                   | Uncondrel | None
RECOVNotify ==> None        None | Message | Transaction

```

Figure 39. Example of Using Resource Definition Online (Part 2 of 2)

Appendix K. Establishing a VTAM Connection for AutoOPERATOR to PATROL EM Communication

AutoOPERATOR establishes a VTAM connection to PATROL EM using a unique ACB. The ACB and destination names must be defined to VTAM and to AutoOPERATOR. For more information, see “Defining VTAM ACBs” and “Modifying the VTAM Connection with BBPARM Member AAOALT00” on page 248.

To enable AutoOPERATOR ALERTs to be sent to PATROL EM through a VTAM connection, you must:

1. Define a VTAM ACB for AutoOPERATOR to use to communicate with PATROL EM.
2. Define a 3270 printer to VTAM.
3. Reset BBPARM member AAOALT00.

Defining VTAM ACBs

You need to define a VTAM ACB that AutoOPERATOR will use to communicate with PATROL EM; for example:

```
SAONETC APPL AUTH=(ACQ,NOTSO,VPACE)
```

where SAONETC is a unique user-specified name representing the APPLID keyword specified in the BBPARM member AAOALT00 (see “Modifying the VTAM Connection with BBPARM Member AAOALT00” on page 248).

You can add this statement to any member in SYS1.VTAMLST. However, if you have other VTAM definitions for the AutoOPERATOR subsystem, you might want to include this statement in the BBIAPPLx member (refer to the *MAINVIEW Common Customization Guide* for more information about the BBIAPPLx member).

Note: Each BBI-SS PAS must have a unique APPLID specified for each subsystem that is connected to PATROL EM. The APPLID name must be different than the one used for BBI-SS PAS to BBI-SS PAS communication.

Defining 3270 Printers to VTAM

Each PATROL EM outboard processor appears as a 3270 printer (for example, a 3284, 3287, etc.) to VTAM and AutoOPERATOR. If the connection is SNA, you must define the processor as a LU3 printer and you must not define it as an SCS printer. An example of a VTAM definition for a local non-SNA printer is:

```
FB4048  LOCAL  CUADDR=048,  
              DLOGMOD=S3270,  
              FEATUR2=MODEL2.  
              TERM=3286
```

An example of a VTAM definition for a local SNA printer is:

```
A070A08    LU    LOCADDR=8,  
            DLOGMOD=DSC2K
```

Resetting BBPARM Member AAOALT00

After a VTAM session is established, AutoOPERATOR communicates ALERT activity to PATROL EM at BBI-SS PAS start up or when you issue the following BBI control commands:

- . RESET PARM AAOALT00
- . RESET NETCALRT
- . START COMP NETCALRT

Controlling the AutoOPERATOR to PATROL EM Interface

Following is a list of BBI control commands that you can use to control the AutoOPERATOR to PATROL EM interface:

Table 29. BBI Control Commands Available to AutoOPERATOR to PATROL EM Interface for VTAM Connections

BBI Control Command	Parameters	Description
.RESET AAOALT00	PARM P	Drops all connections to PATROL EM and terminates the VTAM session. After processing of AAOALTxx where xx can be any suffix, the VTAM connection and all remote connections are established.
.RESET NETCALRT	[ALL <i>destid</i>] [ON OFF RELOAD]	<p>Defines any destination that can be stopped or restarted.</p> <p>The format of the command is:</p> <pre>. RESET NETC [ALL <i>destid</i>] [ON OFF RELOAD]</pre> <p>ALL (Default) Indicates all destinations are to be processed as specified by ON, OFF, or RELOAD.</p> <p><i>destid</i> Indicates a specific destination (VTAM ID) to be processed as specified by ON, OFF, or RELOAD.</p> <p>ON (Default) Attempts to establish a connection.</p> <p>OFF Is a request to disconnect a connection.</p> <p>RELOAD Is a request to send all ALERTs to the specified destination without dropping or reestablishing the connection.</p>
.START COMP NETCALRT	N/A	Requests the starting of the AutoOPERATOR to PATROL EM interface task.
.STOP COMP NETCALRT	N/A	Requests the stopping of an AutoOPERATOR to PATROL EM interface task.
.DISPLAY NETCALRT	ALL <i>destid</i>	<p>ALL Requests the status of all destinations and their current AutoOPERATOR to PATROL EM interface tasks. It is the default.</p> <p><i>destid</i> Specifies a VTAM ID to request the status of a specific destination.</p>

For a complete description of all the available BBI control commands, refer to the *MAINVIEW Administration Guide* where the complete list of BBI control commands is documented.

Modifying the VTAM Connection with BBPARM Member AAOALT00

Use BBPARM member AAOALT00 to define the VTAM ACB name for sending ALERTs to PATROL EM destinations, the names of the destination devices, and the level of diagnostics issued. The parameters in this member are:

Table 30. BBPARM Member AAOALT00

Parameter	Description
APPLID=name	Defines the value of the APPLID keyword on the ACB used by OPEN in establishing the session with VTAM. This must be the same name as the one entered in the VTAMLIST entry as the name of the application.
DEST=device	Defines a VTAM destination for the ALERTs. This VTAM destination (a PATROL EM workstation) must be defined to the network as a 3287 printer.
STATUS= <u>ON</u> OFF	Optional. Defines whether the destination is to be enabled (ON) during initialization, or left disconnected (OFF). The default is ON.
SYNCINTV= <u>30</u> nnn	Optional. Defines the synch-up interval between AutoOPERATOR and the workstation. At this defined interval, all current ALERTs are retransmitted to the workstation to compensate for any earlier lost or misprocessed transmissions. The default is 30 minutes.
MSGLVL= <u>WARN</u> msgtype	<p>Optional. Defines which messages are to be written to the journal; where WARN is the default and msgtype can be:</p> <ul style="list-style-type: none"> • NONE No messages to Journal • SEVERE No VTAM connection, internal errors • ERROR Lost or dropped connection, error limit reached • WARN No BBPARM member, termination request • INFORMATIONAL Normal start and stop messages • DEBUG Echo all messages in the Journal

3287 Data Stream Format for AutoOPERATOR to PATROL EM

AutoOPERATOR sends a message to PATROL EM for each ALERT that is added or deleted. The message consists of entries delimited by one blank space. Refer to Table 31.

Certain fields (such as Origin and Queue Name) might contain imbedded blanks because they are user-defined. Any imbedded blanks are replaced with _ (underscore). The only exception is the message text where imbedded blank spaces are not replaced in this field.

Certain fields (such as Origin and User) can be left completely blank. Since the fields are position-sensitive, any all-blank field is replaced with a single _ (underscore).

Table 31 has a description of each of the fields in the output message.

Table 31. Output Messages Description

Field Name	Maximum Length	Description
Anchor field	7	<p>Literal to identify a message as an ALERT; its value can be:</p> <p>ALRT2 ALERT-add request</p> <p>ALRTCX ALERT-cancel request</p> <p>ALERT text is not included in the data stream for ALRTCX.</p> <p>ALRTHD2 A heading message requesting that all current ALERTs be cancelled, since all ALERTs currently active in the AutoOPERATOR application will be transmitted</p> <p>The ALRTHD2 value is used in the following cases:</p> <ul style="list-style-type: none"> • When AutoOPERATOR initializes • When any of the following BBI commands are issued: <ul style="list-style-type: none"> – .START COMP NETCALRT – .RESET DEST – .RESET AAOALT00 • After each interval specified in BBPARM member AAOALT00 (the SYNCINTV parameter) <p>Note: These values are valid for AutoOPERATOR version 4.1 only, and they are not valid for earlier versions of AutoOPERATOR. For earlier versions, refer to the corresponding <i>MAINVIEW AutoOPERATOR Customization Guide</i> edition for that version.</p>
Origin	8	Origin of the ALERT
Subsys ID	8	BBI-SS PAS ID
Target	8	Target system associated with the ALERT
User	8	UserID of issuer of the ALERT

Table 31. Output Messages Description (Continued)

Field Name	Maximum Length	Description
Date	5	Julian date of the ALERT in the form YYDDD
Time	6	Time of the ALERT in HHMMSS
Alarm Indicator	1	Y/N - should alarm be sounded
Priority	1	User-assigned priority of the ALERT: <ul style="list-style-type: none"> • 1 - Critical • 2 - Major • 3 - Minor • 4 - Warning • 5 - Informational • 6 - Clearing
Color code	1	User-assigned color of the message: <ul style="list-style-type: none"> • 1 - Red • 2 - Pink • 3 - Yellow • 4 - Dark Blue • 5 - Light Blue • 6 - Green • 7 - White
Queue name	8	Name of the AutoOPERATOR queue to which this ALERT is assigned
Help panel name	8	For AutoOPERATOR version 4.1 and above, the datastream includes the help panel name
Key of the ALERT	64	User-specified key of the ALERT
Message text	255	Text of the message Note: The text string /N is to be interpreted as carriage-return line-feed (CR, LF).

Diagnostics

Following are some steps you can take when attempting to diagnose whether or not the VTAM connection has failed.

1. Verify that PATROL EM is connected to the correct 3x74 port
2. If this is done, verify VTAM connectivity and the LUName by connecting a real 3270 terminal

Once the 3270 terminal is connected, issue the VARY VTAM command to inactivate and reactivate the LUName. At the bottom left-hand corner of the display, you should see the *Status* line. This confirms you are working with the correct device.

3. Try issuing the BBI command `. S COMP NETC`

AutoOPERATOR should write some characters (such as ALRTHD2) to the terminal. If this occurs, AutoOPERATOR is communicating correctly with the terminal.

Once you are sure that you can communicate with a 3270 terminal, then you can connect a real 3287 (or equivalent) printer. As AutoOPERATOR ALERTs are being created, they should be written to the printer. (You can further verify that the printer is correctly defined by using CICS or another VTAM application to send output to the printer.)

Finally, if AutoOPERATOR can successfully send ALERTs to the 3287 printer but PATROL EM still cannot receive them, you must begin problem determination steps on the PATROL EM side to determine why PATROL EM is not receiving the ALERTs.

Part 5. Glossary and Index

This part contains the glossary and the index.

Glossary	255
Index	267

Glossary

This glossary defines BMC Software terminology. Other dictionaries and glossaries can be used in conjunction with this glossary.

Since this glossary pertains to BMC Software-related products, some of the terms defined might not appear in this book.

To help you find the information you need, this glossary uses the following cross-references:

Contrast with	Indicates a term that has a contrary or contradictory meaning.
See	Indicates an entry that is a synonym or contains expanded information.
See also	Indicates an entry that contains related information.

A

action. Defined operation, such as modifying a MAINVIEW window, that is performed in response to a command. *See* object.

active window. Any MAINVIEW window in which data can be refreshed. *See* alternate window, current window, window.

administrative view. Display from which a product's management tasks are performed, such as the DSLIST view for managing historical data sets. *See* view.

ALT WIN field. Input field that allows you to specify the window identifier for an alternate window where the results of a hyperlink are displayed. *See* alternate window.

Alternate Access. *See* MAINVIEW Alternate Access.

alternate form. View requested through the FORM command that changes the format of a previously displayed view to show related information. *See also* form, query.

alternate window. (1) Window that is specifically selected to display the results of a hyperlink. (2) Window whose identifier is defined to the ALT WIN field. *Contrast with* current window. *See* active window, window, ALT WIN field.

analyzer. (1) Online display that presents a snapshot of status and activity data and indicates problem areas. (2) Component of CMF MONITOR. *See* CMF MONITOR Analyzer.

application. (1) Program that performs a specific set of tasks within a MAINVIEW product. (2) In MAINVIEW VistaPoint, combination of workloads to enable display of their transaction performance data in a single view.

application trace. *See* trace.

ASCH workload. Workload comprising Advanced Program-to-Program Communication (APPC) address spaces.

AutoCustomization. Online facility for customizing the installation of products. AutoCustomization provides an ISPF panel interface that both presents customization steps in sequence and provides current status information about the progress of the installation.

automatic screen update. Usage mode wherein the currently displayed screen is refreshed automatically with new data at an interval you specify. Invoked by the ASU command.

B

batch workload. Workload consisting of address spaces running batch jobs.

BBI. Basic architecture that distributes work between workstations and multiple OS/390 targets for BMC Software MAINVIEW products.

BBI-SS PAS. *See* BBI subsystem product address space.

BBI subsystem product address space (BBI-SS PAS). OS/390 subsystem address space that manages communication between local and remote systems and that contains one or more of the following products:

- Command MQ for S/390
- MAINVIEW AutoOPERATOR
- MAINVIEW for CICS
- MAINVIEW for DB2
- MAINVIEW for DBCTL
- MAINVIEW for IMS Online
- MAINVIEW for MQSeries
- MAINVIEW SRM
- MAINVIEW VistaPoint (for CICS, DB2, DBCTL, and IMS workloads)

BBPARM. *See* parameter library.

BBPROC. *See* procedure library.

BBPROF. See profile library.

BBSAMP. See sample library.

BBV. See MAINVIEW Alternate Access.

BBXS. BMC Software Subsystem Services. Common set of service routines loaded into common storage and used by several BMC Software MAINVIEW products.

border. Visual indication of the boundaries of a window.

bottleneck analysis. Process of determining which resources have insufficient capacity to provide acceptable service levels and that therefore can cause performance problems.

C

CA-Disk. Data management system by Computer Associates that replaced the DMS product.

CAS. Coordinating address space. One of the address spaces used by the MAINVIEW windows environment architecture. The CAS supplies common services and enables communication between linked systems. Each OS/390 or z/OS image requires a separate CAS. Cross-system communication is established through the CAS using VTAM and XCF communication links.

CFMON. See coupling facility monitoring.

chart. Display format for graphical data. See also graph.

CICSplex. User-defined set of one or more CICS systems that are controlled and managed as a single functional entity.

CMF MONITOR. Comprehensive Management Facility MONITOR. Product that measures and reports on all critical system resources, such as CPU, channel, and device usage; memory, paging, and swapping activity; and workload performance.

CMF MONITOR Analyzer. Batch component of CMF MONITOR that reads the SMF user and 70 series records created by the CMF MONITOR Extractor and/or the RMF Extractor and formats them into printed system performance reports.

CMF MONITOR Extractor. Component of CMF that collects performance statistics for CMF MONITOR Analyzer, CMF MONITOR Online, MAINVIEW for OS/390, and RMF postprocessor. See CMF MONITOR Analyzer, CMF MONITOR Online, MAINVIEW for OS/390.

CMF MONITOR Online. Component of CMF that uses the MAINVIEW window interface to present data on all address spaces, their use of various system resources, and the delays that each address space incurs while waiting for access to these resources. See CMF MONITOR, MAINVIEW for OS/390.

CMF Type 79 API. Application programming interface, provided by CMF, that provides access to MAINVIEW SMF-type 79 records.

CMFMON. Component of CMF MONITOR that simplifies online retrieval of information about system hardware and application performance and creates MAINVIEW SMF-type 79 records.

The CMFMON *online facility* can be used to view data in one or more formatted screens.

The CMFMON *write facility* can be used to write collected data as MAINVIEW SMF-type 79 records to an SMF or sequential data set.

CMRDETL. MAINVIEW for CICS data set that stores detail transaction records (type 6E) and abend records (type 6D). Detail records are logged for each successful transaction. Abend records are written when an abend occurs. Both records have the same format when stored on CMRDETL.

CMRSTATS. MAINVIEW for CICS data set that stores both CICS operational statistic records, at five-minute intervals, and other records, at intervals defined by parameters specified during customization (using CMRSOPT).

column. Vertical component of a view or display, typically containing fields of the same type of information, that varies by the objects associated in each row.

collection interval. Length of time data is collected. See also delta mode, total mode.

command delimiter. Special character, usually a ; (semicolon), used to stack commands typed concurrently on the COMMAND line for sequential execution.

COMMAND line. Line in the control area of the display screen where primary commands can be typed. Contrast with line command column.

Command MQ Automation D/S. Command MQ agents, which provide local proactive monitoring for both MQSeries and MSMQ (Microsoft message queue manager). The Command MQ agents operate at the local node level where they continue to perform functions regardless of the availability of the MQM (message queue manager) network. Functionality includes automatic monitoring and restarts of channels, queue managers, queues and command servers. In cases where automated recovery is not possible, the agents transport critical alert information to a central console.

Command MQ Automation S/390. Command MQ component, which monitors the MQM (message queue manager) networks and intercedes to perform corrective actions when problems arise. Solutions include:

- Dead-Letter Queue management
- System Queue Archival
- Service Interval Performance solutions
- Channel Availability

These solutions help ensure immediate relief to some of the most pressing MQM operations and performance problems.

Command MQ for D/S. Command MQ for D/S utilizes a true client/server architecture and employs resident agents to provide configuration, administration, performance monitoring and operations management for the MQM (message queue manager) network.

Command MQ for S/390. See MAINVIEW for MQSeries.

COMMON STORAGE MONITOR. Component of MAINVIEW for OS/390 that monitors usage and reconfigures OS/390 or z/OS common storage blocks.

composite workload. Workload made up of a WLM workload or other workloads, which are called *constituent workloads*.

constituent workload. Member of a composite workload. Constituent workloads in a composite usually belong to a single workload class, but sometimes are mixed.

contention. Occurs when there are more requests for service than there are servers available.

context. In a Plex Manager view, field that contains the name of a target or group of targets specified with the CONTEXT command. See scope, service point, SSI context, target context.

CONTEXT command. Specifies either a MAINVIEW product and a specific target for that product (see target context) or a MAINVIEW product and a name representing one or more targets (see *SSI context*) for that product.

control statement. (1) Statement that interrupts a sequence of instructions and transfers control to another part of the program. (2) Statement that names samplers and other parameters that configure the MAINVIEW components to perform specified functions. (3) In CMF MONITOR, statement in a parameter library member used to identify a sampler in the extractor or a report in the analyzer, or to describe either component's processing requirements to the operating system.

coupling facility monitoring (CFMON). Coupling facility views that monitor the activity of your system's coupling facilities.

current data. Data that reflects the system in its current state. The two types of current data are realtime data and interval data. Contrast with historical data. See also interval data and realtime data.

current window. In the MAINVIEW window environment, window where the main dialog with the application takes place. The current window is used as the default window destination for commands issued on the COMMAND line when no window number is specified. Contrast with alternate window. See active window, window.

D

DASD. Direct Access Storage Device. (1) A device with rotating recording surfaces that provides immediate access to stored data. (2) Any device that responds to a DASD program.

data collector. Program that belongs to a MAINVIEW product and that collects data from various sources and stores the data in records used by views. For example, MAINVIEW for OS/390 data collectors obtain data from OS/390 or z/OS services, OS/390 or z/OS control blocks, CMF MONITOR

Extractor control blocks, and other sources. Contrast with extractor.

delta mode. (1) In MAINVIEW for DB2 analyzer displays, difference between the value sampled at the start of the current statistics interval and the value sampled by the current analyzer request. See also *statistics interval*. (2) In CMFMON, usage mode wherein certain columns of data reflect the difference in values between one sample cycle and the next. Invoked by the DELTA ON command. See also collection interval, sample cycle, total mode.

DFSMS. Data Facility Storage Management System. Data management, backup, and HSM software from IBM for OS/390 or z/OS mainframes.

DMR. See MAINVIEW for DB2.

DMS. Data Management System. See CA-Disk.

DMS2HSM. See MAINVIEW SRM DMS2HSM.

DSO. Data Set Optimizer. CMF MONITOR Extractor component that uses CMF MONITOR Extractor data to produce reports specifying the optimal ordering of data sets on moveable head devices.

E

EasyHSM. See MAINVIEW SRM EasyHSM.

EasyPOOL. See MAINVIEW SRM EasyPOOL.

EasySMS. See MAINVIEW SRM EasySMS.

element. (1) Data component of a data collector record, shown in a view as a field. (2) Internal value of a field in a view, used in product functions.

element help. Online help for a field in a view. The preferred term is *field help*.

Enterprise Storage Automation. See MAINVIEW SRM Enterprise Storage Automation.

event. A message issued by Enterprise Storage Automation. User-defined storage occurrences generate events in the form of messages. These events provide an early warning system for storage problems and are routed to user-specified destinations for central viewing and management.

Event Collector. Component for MAINVIEW for IMS Online, MAINVIEW for IMS Offline, and MAINVIEW for DBCTL that collects data about events in the IMS environment. This data is required for Workload Monitor and optional for Workload Analyzer (except for the workload trace service). This data also is recorded as transaction records (X'FA') and program records (X'F9') on the IMS system log for later use by the MAINVIEW for IMS Offline components: Performance Reporter and Transaction Accountant.

expand. Predefined link from one display to a related display. See also hyperlink.

extractor. Program that collects data from various sources and keeps the data control blocks to be written as records. Extractors obtain data from services, control blocks, and other sources. *Contrast with* data collector.

extractor interval. *See* collection interval.

F

fast path. Predefined link between one screen and another. To use the fast path, place the cursor on a single value in a field and press Enter. The resulting screen displays more detailed information about the selected value. *See also* hyperlink.

field. Group of character positions within a screen or report used to type or display specific information.

field help. Online help describing the purpose or contents of a field on a screen. To display field help, place the cursor anywhere in a field and press PF1 (HELP). In some products, field help is accessible from the screen help that is displayed when you press PF1.

filter. Selection criteria used to limit the number of rows displayed in a view. Data that does not meet the selection criteria is not displayed. A filter is composed of an element, an operator, and an operand (a number or character string). Filters can be implemented in view customization, through the PARM/QPARM commands, or through the Where/QWhere commands. Filters are established against elements of data.

fire. The term used to indicate that an event has triggered an action. In MAINVIEW AutoOPERATOR, when a rule selection criteria matches an incoming event and *fires*, the user-specified automation actions are performed. This process is also called *handling* the event.

fixed field. Field that remains stationary at the left margin of a screen that is scrolled either right or left.

FOCAL POINT. MAINVIEW product that displays a summary of key performance indicators across systems, sites, and applications from a single terminal.

form. One of two constituent parts of a view; the other is query. A form defines how the data is presented; a query identifies the data required for the view. *See also* query, view.

full-screen mode. Display of a MAINVIEW product application or service on the entire screen. There is no window information line. *Contrast with* windows mode.

G

global command. Any MAINVIEW window interface command that can affect all windows in the window area of a MAINVIEW display.

graph. Graphical display of data that you select from a MAINVIEW window environment view. *See also* chart.

H

hilevel. For MAINVIEW products, high-level data set qualifier required by a site's naming conventions.

historical data. (1) Data that reflects the system as it existed at the end of a past recording interval or the duration of several intervals. (2) Any data stored in the historical database and retrieved using the TIME command. *Contrast with* current data, interval data and realtime data.

historical database. Collection of performance data written at the end of each installation-defined recording interval and containing up to 100 VSAM clusters. Data is extracted from the historical database with the TIME command. *See* historical data.

historical data set. In MAINVIEW products that display historical data, VSAM cluster file in which data is recorded at regular intervals.

HSM. (Hierarchical Storage Management) Automatic movement of files from hard disk to slower, less-expensive storage media. The typical hierarchy is from magnetic disk to optical disk to tape.

hyperlink. (1) Preset field in a view or an EXPAND line on a display that permits you to

- Access cursor-sensitive help
- Issue commands
- Link to another view or display

The transfer can be either within a single product or to a related display/view in a different BMC Software product. Generally, hyperlinked fields are highlighted. (2) Cursor-activated short path from a topic or term in online help to related information. *See also* fast path.

I

Image log. Collection of screen-display records. Image logs can be created for both the BBI-SS PAS and the BBI terminal session (TS).

The BBI-SS PAS Image log consists of two data sets that are used alternately: as one fills up, the other is used. Logging to the BBI-SS PAS Image log stops when both data sets are filled and the first data set is not processed by the archive program.

The TS Image log is a single data set that wraps around when full.

IMSplex System Manager (IPSM). MVIMS Online and MVDBC service that provides Single System Image views of resources and bottlenecks for applications across one or more IMS regions and systems.

interval data. Cumulative data collected during a collection interval. Intervals usually last from 15 to 30 minutes depending on how the recording interval is specified during product customization. *Contrast with* historical data.

Note: If change is made to the workloads, a new interval will be started.

See also current data and realtime data.

InTune. Product for improving application program performance. It monitors the program and provides information used to reduce bottlenecks and delays.

IRUF. IMS Resource Utilization File (IRUF). IRUFs can be either detail (one event, one record) or summarized (more than one event, one record). A detail IRUF is created by processing the IMS system log through a program called IMFLEDIT. A summarized IRUF is created by processing one or more detail IRUFs, one or more summarized IRUFs, or a combination of both, through a sort program and the TASCOSTR program.

J

job activity view. Report about address space consumption of resources. *See* view.

journal. Special-purpose data set that stores the chronological records of operator and system actions.

Journal log. Collection of messages. Journal logs are created for both the BBI-SS PAS and the BBI terminal session (TS).

The BBI-SS PAS Journal log consists of two data sets that are used alternately: as one fills up, the other is used. Logging to the BBI-SS PAS Journal log stops when both data sets are filled and the first data set is not being processed by the archive program.

The TS Journal log is a single data set that wraps around when full.

L

line command. Command that you type in the line command column in a view or display. Line commands initiate actions that apply to the data displayed in that particular row.

line command column. Command input column on the left side of a view or display. *Contrast with* COMMAND line.

Log Edit. In the MAINVIEW for IMS Offline program named IMFLEDIT, function that extracts transaction (X'FA') and program (X'F9') records from the IMS system log. IMFLEDIT also extracts certain records that were recorded on the system log by IMS. IMFLEDIT then formats the records into a file called the IMS Resource Utilization File (IRUF).

M

MAINVIEW. BMC Software integrated systems management architecture.

MAINVIEW Alarm Manager. In conjunction with other MAINVIEW products, notifies you when an exception condition occurs. MAINVIEW Alarm Manager is capable of monitoring multiple systems simultaneously, which means that MAINVIEW Alarm Manager installed on one system keeps track of your entire sysplex. You can then display a single view that show exceptions for all MAINVIEW performance monitors within your OS/390 or z/OS enterprise.

MAINVIEW Alternate Access. Enables MAINVIEW products to be used without TSO by providing access through EXCP and VTAM interfaces.

MAINVIEW Application Program Interface. REXX- or CLIST-based, callable interface that allows MAINVIEW AutoOPERATOR EXECs to access MAINVIEW monitor product view data.

MAINVIEW AutoOPERATOR. Product that uses tools, techniques, and facilities to automate routine operator tasks and provide online performance monitoring, and that achieves high availability through error minimization, improved productivity, and problem prediction and prevention.

MAINVIEW control area. In the MAINVIEW window environment, first three lines at the top of the view containing the window information line and the COMMAND, SCROLL, CURR WIN, and ALT WIN lines. The control area cannot be customized and is part of the information display. *Contrast with* MAINVIEW display area, MAINVIEW window area.

MAINVIEW display area. *See* MAINVIEW window area.

MAINVIEW Explorer. Product that provides access to MAINVIEW products from a Web browser running under Windows. MAINVIEW Explorer replaces MAINVIEW Desktop.

MAINVIEW for CICS. Product (formerly MV MANAGER for CICS) that provides realtime application performance analysis and monitoring for CICS system management.

MAINVIEW for DB2. Product (formerly MV MANAGER for DB2) that provides realtime and historical application performance analysis and monitoring for DB2 subsystem management.

MAINVIEW for DBCTL. Product (formerly MV MANAGER for DBCTL) that provides realtime application performance analysis and monitoring for DBCTL management.

MAINVIEW for IMS (MVIMS) Offline. Product with a Performance Reporter component that organizes data and prints reports used to analyze IMS performance and a Transaction Accountant component that produces cost accounting and user charge-back records and reports.

MAINVIEW for IMS (MVIMS) Online. Product that provides realtime application performance analysis and monitoring for IMS management.

MAINVIEW for IP. Product that monitors OS/390 and z/OS mission-critical application performance as it relates to TCP/IP stack usage. Collected data includes availability, connections, response times, routers, service levels, storage, traffic, Web cache, and so on.

MAINVIEW for Linux-Servers. Product that allows you to monitor the performance of your Linux systems from the MAINVIEW windows interface.

MAINVIEW for MQSeries. Delivers comprehensive capabilities for configuration, administration, performance monitoring and operations management for an entire MQM (message queue manager) network.

MAINVIEW for OS/390. System management application (known as MAINVIEW for MVS prior to version 2.5). Built upon the MAINVIEW window environment architecture, it uses the window interface to provide access to system performance data and other functions necessary in the overall management of an enterprise.

MAINVIEW for UNIX System Services. System management application that allows you to monitor the performance of the Unix System Services from a MAINVIEW window interface.

MAINVIEW for VTAM. Product that displays application performance data by application, transaction ID, and LU name. This collected data includes: connections, response time statistics, application availability, and application throughput.

MAINVIEW for WebSphere. Product that provides Web monitoring and management for applications integrated with IBM WebSphere Application Server for OS/390 or z/OS.

MAINVIEW Selection Menu. ISPF selection panel that provides access to all MAINVIEW windows-mode and full-screen mode products.

MAINVIEW SRM. *See* MAINVIEW Storage Resource Manager (SRM).

MAINVIEW SRM DMS2HSM. Product that facilitates the conversion of CA-Disk, formerly known as DMS, to HSM.

MAINVIEW SRM EasyHSM. Product that provides online monitoring and reporting to help storage managers use DFHSM efficiently.

MAINVIEW SRM EasyPOOL. Product that provides control over data set allocation and enforcement of allocation and naming standards. EasyPOOL functions operate at the operating system level to intercept normal job processing, thus providing services without any JCL changes.

MAINVIEW SRM EasySMS. Product that provides tools that aid in the conversion to DFSMS and provides enhancement to the DFSMS environment after implementation. EasySMS consists of the EasyACS functions, the SMSACSTE function, and the Monitoring and Positioning Facility.

MAINVIEW SRM Enterprise Storage Automation. Product that delivers powerful event generation and storage automation technology across the storage enterprise. Used in conjunction with MAINVIEW AutoOPERATOR, automated solutions to perform pool, volume, application, or data set-level manipulation can be created and used in response to any condition or invoked to perform ad hoc requests

MAINVIEW SRM SG-Auto. Product that provides early warning notification of storage anomalies and automated responses to those anomalies based on conditions in the storage subsystem.

MAINVIEW SRM SG-Control. Product that provides real-time monitoring, budgeting, and control of DASD space utilization.

MAINVIEW SRM StopX37/II. Product that provides enhancements to OS/390 or z/OS space management, reducing the incidence of space-related processing problems. The StopX37/II functions operate at the system level to interceptabend conditions or standards violations, thus providing services without any JCL changes.

MAINVIEW SRM StorageGUARD. Product that monitors and reports on DASD consumption and provides historical views to help control current and future DASD usage.

MAINVIEW Storage Resource Manager (SRM). Suite of products that assists in all phases of OS/390 or z/OS storage management. MAINVIEW SRM consists of products that perform automation, reporting, trend analysis, and error correction for storage management.

MAINVIEW SYSPROG Services. *See* SYSPROG Services.

MAINVIEW VistaPoint. Product that provides enterprise-wide views of performance. Application and workload views are available for CICS, DB2, DBCTL, IMS, and OS/390. Data is summarized at the level of detail needed; for example, views can be for a single target, an OS/390 or z/OS image, or an entire enterprise.

MAINVIEW window area. Portion of the information display that is not the control area and in which views are displayed and windows opened. It includes all but the first three lines of the information display. *Contrast with* MAINVIEW control area.

monitor. Online service that measures resources or workloads at user-defined intervals and issues warnings when user-defined thresholds are exceeded.

Multi-Level Automation (MLA). The user-defined, multiple step process in Enterprise Storage Automation that implements solutions in a tiered approach, where solutions are invoked one after another until the condition is resolved.

MVALARM. *See* MAINVIEW Alarm Manager.

MVAPI. *See* MAINVIEW Application Program Interface.

MVCICS. *See* MAINVIEW for CICS.

MVDB2. *See* MAINVIEW for DB2.

MVDBC. *See* MAINVIEW for DBCTL.

MVIMS. *See* MAINVIEW for IMS.

MVIP. *See* MAINVIEW for IP.

MVLNX. *See* MAINVIEW for Linux-Servers.

MVMQ. *See* MAINVIEW for MQSeries.

MVMVS. *See* MAINVIEW for OS/390.

MVScope. MAINVIEW for OS/390 application that traces both CPU usage down to the CSECT level and I/O usage down to the channel program level.

MVSRM. *See* MAINVIEW Storage Resource Manager (SRM).

MVSRMHSM. *See* MAINVIEW SRM EasyHSM.
MVSRMSGC. *See* MAINVIEW SRM SG-Control.
MVSRMSGD. *See* MAINVIEW SRM StorageGUARD.
MVSRMSGP. *See* MAINVIEW SRM StorageGUARD.
MVVP. *See* MAINVIEW VistaPoint.
MVV TAM. *See* MAINVIEW for VTAM.
MVWEB. *See* MAINVIEW for WebSphere.

N

nested help. Multiple layers of help pop-up windows. Each successive layer is accessed by clicking a hyperlink from the previous layer.

O

object. Anything you can manipulate as a single unit. MAINVIEW objects can be any of the following: product, secondary window, view, row, column, or field.

You can issue an action against an object by issuing a line command in the line command column to the left of the object. *See* action.

OMVS workload. Workload consisting of OS/390 OpenEdition address spaces.

online help. Help information that is accessible online.

OS/390 and z/OS Installer. BMC Software common installation system for mainframe products.

OS/390 product address space (PAS). Address space containing OS/390 or z/OS data collectors, including the CMF MONITOR Extractor. Used by the MAINVIEW for OS/390, MAINVIEW for Unix System Services, and CMF MONITOR products. *See* PAS.

P

parameter library. Data set consisting of members that contain parameters for specific MAINVIEW products or a support component. There can be several versions:

- The distributed parameter library, called BBPARAM
- A site-specific parameter library or libraries

These can be

- A library created by AutoCustomization, called UBBPARAM
- A library created manually, with a unique name

PAS. Product address space. Used by the MAINVIEW products. Contains data collectors and other product functions. *See* OS/390 product address space (PAS), BBI subsystem product address space (BBI-SS PAS).

performance group workload. Collection of address spaced defined to OS/390 or z/OS. If you are running OS/390 or z/OS with WLM in compatibility mode, MAINVIEW for OS/390 creates a performance group workload instead of a service class. *See* service class workload, workload definition.

PERFORMANCE MANAGER. MAINVIEW for CICS online service for monitoring and managing current performance of CICS regions.

Performance Reporter (MVIMS Offline). MVIMS Offline component that organizes data and prints reports that can be used to analyze IMS performance.

Performance Reporter. Product component that generates offline batch reports. The following products can generate these reports:

- MAINVIEW for DB2
- MAINVIEW for CICS

Plex Manager. Product through which cross-system communication, MAINVIEW security, and an SSI context are established and controlled. Plex Manager is shipped with MAINVIEW window environment products as part of the coordinating address space (CAS) and is accessible as a menu option from the MAINVIEW Selection Menu.

PRGP workload. In MVS/SP 5.0 or earlier, or in compatibility mode in MVS/SP 5.1 or later, composite of service classes. MAINVIEW for OS/390 creates a performance group workload for each performance group defined in the current IEAIPS.xx member.

procedure library. Data set consisting of members that contain executable procedures used by MAINVIEW AutoOPERATOR. These procedures are execute command lists (EXECs) that automate site functions. There can be several versions:

- The distributed parameter library, called BBPROC
- A site-specific parameter library or libraries

These can be

- A library created by AutoCustomization, called UBBPROC
- A library created manually, with a unique name

The site-created EXECs can be either user-written or customized MAINVIEW AutoOPERATOR-supplied EXECs from BBPROC.

product address space. *See* PAS.

profile library. Data set consisting of members that contain profile information and cycle refresh definitions for a terminal session connected to a BBI-SS PAS. Other members are dynamically created by MAINVIEW applications. There can be several versions:

- The distributed profile library, called BBPROF
- A site-specific profile library or libraries

These can be

- A library created by AutoCustomization, called SBBPROF
- A library created manually, with a unique name

The site library is a common profile shared by all site users. The terminal session CLIST creates a user profile automatically if one does not exist; it is called userid.BBPROF, where userid is your logon ID. User profile libraries allow each user to specify unique PF keys, CYCLE commands, target system defaults, a Primary Option Menu, and a unique set of application profiles.

Q

query. One of two constituent parts of a view; the other is form. A query defines the data for a view; a form defines the display format. *See also* form, view.

R

realtime data. Performance data as it exists at the moment of inquiry. Realtime data is recorded during the smallest unit of time for data collection. *Contrast with* historical data. *See also* current data and interval data.

Resource Analyzer. Online realtime displays used to analyze IMS resources and determine which are affected by specific workload problems.

Resource Monitor. Online data collection services used to monitor IMS resources and issue warnings when defined utilization thresholds are exceeded.

row. (1) Horizontal component of a view or display comprising all the fields pertaining to a single device, address space, user, etc. (2) Horizontal component of a DB2 table consisting of a sequence of values, one for each column of the table.

RxD2. Product that provides access to DB2 from REXX. It provides tools to query the DB2 catalog, issue dynamic SQL, test DB2 applications, analyze EXPLAIN data, generate DDL or DB2 utility JCL, edit DB2 table spaces, perform security administration, and much more.

S

sample cycle. Time between data samples.

For the CMF MONITOR Extractor, this is the time specified in the extractor control statements (usually 1 to 5 seconds).

For realtime data, the cycle is not fixed. Data is sampled each time you press Enter.

sample library. Data set consisting of members each of which contains one of the following:

- Sample JCL that can be edited to perform specific functions
- A macro that is referenced in the assembly of user-written services
- A sample user exit routine

There can be several versions:

- The distributed sample library, called BBSAMP
- A site-specific sample library or libraries

These can be

- A library created by AutoCustomization, called UBBSAMP
- A library created manually, with a unique name

sampler. Program that monitors a specific aspect of system performance. Includes utilization thresholds used by the Exception Monitor. The CMF MONITOR Extractor contains samplers.

SBBPROF. *See* profile library.

scope. Subset of an SSI context. The scope could be all the data for the context or a subset of data within the context. It is user- or site-defined. *See* SSI context, target.

screen definition. Configuration of one or more views that have been stored with the SAVEScr command and assigned a unique name. A screen includes the layout of the windows and the view, context, system, and product active in each window.

selection view. In MAINVIEW products, view displaying a list of available views.

service class workload. Collection of address spaces defined to OS/390 or z/OS. If you are running Workload Manager (WLM) in goal mode, MAINVIEW for OS/390 creates a service class workload for each service class that you define through WLM definition dialogs.

If you are running MVS 4.3 or earlier, or MVS/SP 5.1 or later with WLM in compatibility mode, MVS creates a performance group workload instead of a service class. *See* performance group workload.

service objective. Workload performance goal, specified in terms of response time for TSO workloads or turnaround time for batch workloads. Performance group workloads can be measured by either objective. Composite workload service objectives consist of user-defined weighting factors assigned to each constituent workload. For compatibility mode, neither OS/390 nor z/OS provides any way to measure service.

service point. Specification, to MAINVIEW, of the services required to enable a specific product. Services can be actions, selectors, or views. Each target (for example, CICS, DB2, or IMS) has its own service point.

The PLEX view lists all the defined service points known to the CAS to which the terminal session is connected.

service request block (SRB). Control block that represents a routine to be dispatched. SRB mode routines generally perform work for the operating system at a high priority. An SRB is similar to a task control block (TCB) in that it identifies a unit of work to the system. *See also* task control block.

service select code. Code entered to invoke analyzers, monitors, and general services. This code is also the name of the individual service.

session. Total period of time an address space has been active. A session begins when monitoring can be performed. If the product address space (PAS) starts after the job, the session starts with the PAS.

SG-Auto. *See* MAINVIEW SRM SG-Auto.

SG-Control. *See* MAINVIEW SRM SG-Control.

single system image (SSI). Feature of the MAINVIEW window environment architecture where you can view and perform actions on multiple OS/390 systems as though they were a single system. The rows of a single tabular view can contain rows from different OS/390 or z/OS images.

Skeleton Tailoring Facility. A facility in MAINVIEW AutoOPERATOR that allows skeleton JCL to be used during job submission. Skeleton JCL can contain variables within the JCL statements to be substituted with data values at job submission time. Directive statements can be used in the skeleton JCL to cause the repetition of a set of skeleton statements. This facility functions similar to the TSO skeleton tailoring facility.

SRB. *See* service request block.

SSI. *See* single system image.

SSI context. Name created to represent one or more targets for a given product. *See* context, target.

started task workload. Address spaces running jobs that were initiated programmatically.

statistics interval. For MAINVIEW for DB2, cumulative count within a predefined interval (30-minute default set by the DB2STATS parameter in the distributed BBPARM member BBIISP00) for an analyzer service DELTA or RATE display. Specifying the DELTA parameter displays the current value as the difference between the value sampled by the current analyzer request and the value sampled at the start of the current interval. Specifying the RATE parameter displays the current value by minute (DELTA divided by the number of elapsed minutes).

stem variables. A REXX facility, supported in MAINVIEW AutoOPERATOR REXX EXECs and the Skeleton Tailoring Facility, where variable names end with a period followed by a

number, such as &POOL.1. This configuration allows each variable to actually represent a table or array of data, with the zero variable containing the number of entries in the array. For example, &POOL.0 = 5 would indicate variables &POOL.1 through &POOL.5 exist.

StopX37/II. *See* MAINVIEW SRM StopX37/II.

StorageGUARD. *See* MAINVIEW SRM StorageGUARD.

summary view. View created from a tabular view using the Summarize option in view customization. A summary view compresses several rows of data into a single row based on the summarize criteria.

SYSPROG services. Component of MAINVIEW for OS/390. Over 100 services that detect, diagnose, and correct OS/390 or z/OS system problems as they occur. Accessible from the OS/390 Performance and Control Main Menu. Note that this component is also available as a stand-alone product MAINVIEW SYSPROG Services.

system resource. *See* object.

T

target. Entity monitored by one or more MAINVIEW products, such as an OS/390 or z/OS image, an IMS or DB2 subsystem, a CICS region, or related workloads across systems. *See* context, scope, SSI context.

target context. Single target/product combination. *See* context.

TASCOSTR. MAINVIEW for IMS Offline program that summarizes detail and summary IMS Resource Utilization Files (IRUFs) to be used as input to the offline components.

task control block (TCB). Address space-specific control block that represents a unit of work that is dispatched in the address space in which it was created. *See also* service request block.

TCB. *See* task control block.

terminal session (TS). Single point of control for MAINVIEW products, allowing data manipulation and data display and providing other terminal user services for MAINVIEW products. The terminal session runs in a user address space (either a TSO address space or a standalone address space for EXCP/VTAM access).

TDIR. *See* trace log directory.

threshold. Specified value used to determine whether the data in a field meets specific criteria.

TLDS. *See* trace log data set.

total mode. Usage mode in CMFMON wherein certain columns of data reflect the cumulative value between collection intervals. Invoked by the DELTA OFF command. *See also* collection interval, delta mode.

trace. (1) Record of a series of events chronologically listed as they occur. (2) Online data collection and display services that track transaction activity through DB2, IMS, or CICS.

trace log data set (TLDS). Single or multiple external VSAM data sets containing summary or detail trace data for later viewing or printing. The trace log(s) can be defined as needed or dynamically allocated by the BBI-SS PAS. Each trace request is assigned its own trace log data set(s).

trace log directory (TDIR). VSAM linear data set containing one entry for each trace log data set. Each entry indicates the date and time of data set creation, the current status of the data set, the trace target, and other related information.

transaction. Specific set of input data that initiates a predefined process or job.

Transaction Accountant. MVIMS Offline component that produces cost accounting and user charge-back records and reports.

TS. *See* terminal session.

TSO workload. Workload that consists of address spaces running TSO sessions.

U

UAS. *See* user address space.

UBBPARM. *See* parameter library.

UBBPROC. *See* procedure library.

UBBSAMP. *See* sample library.

user address space. Runs a MAINVIEW terminal session (TS) in TSO, VTAM, or EXCP mode.

User BBPROF. *See* profile library.

V

view. Formatted data within a MAINVIEW window, acquired from a product as a result of a view command or action. A view consists of two parts: query and form. *See also* form, job activity view, query.

view definition. Meaning of data that appears online, including source of data, selection criteria for data field inclusion and placement, data format, summarization, context, product, view name, hyperlink fields, and threshold conditions.

view command. Name of a view that you type on the COMMAND line to display that view.

view command stack. Internal stack of up to 10 queries. For each command, the stack contains the filter parameters, sort order, context, product, and timeframe that accompany the view.

view help. Online help describing the purpose of a view. To display view help, place the cursor on the view name on the window information line and press PF1 (HELP).

W

window. Area of the MAINVIEW screen in which views and resources are presented. A window has visible boundaries and can be smaller than or equal in size to the MAINVIEW window area. *See* active window, alternate window, current window, MAINVIEW window area.

window information line. Top border of a window. Shows the window identifier, the name of the view displayed in the window, the system, the scope, the product reflected by the window, and the timeframe for which the data in the window is relevant. *See also* window status field.

window number. Sequential number assigned by MAINVIEW to each window when it is opened. The window number is the second character in the window status field. *See also* window status field.

window status. One-character letter in the window status field that indicates when a window is ready to receive commands, is busy processing commands, is not to be updated, or contains no data. It also indicates when an error has occurred in a window. The window status is the first character in the window status field. *See also* window information line, window status field.

window status field. Field on the window information line that shows the current status and assigned number of the window. *See also* window number, window status.

windows mode. Display of one or more MAINVIEW product views on a screen that can be divided into a maximum of 20 windows. A window information line defines the top border of each window. *Contrast with* full-screen mode.

WLM workload. In goal mode in MVS/SP 5.1 and later, a composite of service classes. MAINVIEW for OS/390 creates a workload for each WLM workload defined in the active service policy.

workflow. Measure of system activity that indicates how efficiently system resources are serving the jobs in a workload.

workload. (1) Systematic grouping of units of work (e.g., address spaces, CICS transactions, IMS transactions) according to classification criteria established by a system administrator. (2) In OS/390 or z/OS, a group of service classes within a service definition.

workload activity view. Tracks workload activity as the workload accesses system resources. A workload activity view measures workload activity in terms of resource consumption and how well the workload activity meets its service objectives.

Workload Analyzer. Online data collection and display services used to analyze IMS workloads and determine problem causes.

workload definition. Workload created through the WKLIST view. Contains a unique name, a description, an initial status, a current status, and selection criteria by which address spaces are selected for inclusion in the workload. *See* Workload Definition Facility.

Workload Definition Facility. In MAINVIEW for OS/390, WKLIST view and its associated dialogs through which workloads are defined and service objectives set.

workload delay view. Tracks workload performance as the workload accesses system resources. A workload delay view measures any delay a workload experiences as it contends for those resources.

Workload Monitor. Online data collection services used to monitor IMS workloads and issue warnings when defined thresholds are exceeded.

workload objectives. Performance goals for a workload, defined in WKLIST. Objectives can include measures of performance such as response times and batch turnaround times.

Index

Symbols

\$\$\$CSMON JCL in BBPARM 137
\$EXECJOB 221
\$EXECTST 221
\$TESTDAT 221

A

AAOALS00 190
 BBPARM member 189
 modifying ALERT parameters
 Dynamic Parameter Manager 36–39
AAOALT00 192
 BBPARM member 189
AAOARP00 193
 BBPARM member 189
 modifying Automation Reporter parameters
 Dynamic Parameter Manager 39–43
AAOEWD00 194
 BBPARM member 189
AAOEXP00 153–154, 195
 modifying EXEC parameters
 Dynamic Parameter Manager 44–53
 EXEC= 154
 MAXHIGH= 154
 MAXNORM= 154
AAOGME00
 BBPARM member 189
 parameters 199
 resetting 125
AAOGME00 parameters
 DEBUGMSG 199
 GMEACT 199
 HOSTNAME 199
 LSTNPORT 199
 TCPIPJOB 199
 TGTHB 199
 TG TIP 200
 TGT MXMSL 200
 TGTNAME 200
 TGTPORT 200
 TGTRTC 200
 TGTRTI 200
 TRACEAPP 200
 TRACEBUF 200
 TRACEGME 201
 TRACELK 201
 TRACESEC 201
 TRACEZN 201
AAOMQL00
 BBPARM member 189
AAONCD00 204
 BBPARM member 189
AAOPRM00 205
 BBPARM member 189
AAORUL00 208
 BBPARM member 189
AAOTRN00 209, 217
 BBPARM member 189
 TapeSHARE 210
AAOTSP00
 BBPARM member 189
 modifying TapeSHARE parameters
 Dynamic Parameter Manager 54–61
ACF2 security 85
ACTION
 AAOTSP00 parameter
 modifying 58
address space application
 modifying JES2 job numbers 155
advanced-installation checklist 29
ALERT
 setting storage thresholds
 Dynamic Parameter Manager 37
ALERT QUEUE
 AAOARP00 parameter
 modifying 43
ALERTNV
 AAOALS00 parameter
 modifying 39
ALERTs
 queues
 creating 129
 deleting 129
 synchronization 128
ALERTs thresholds
 Dynamic Parameter Manager 36–39
alternate database
 CSM application
 selecting 139
AO exit
 execution 105
 return code 105
 routines 104
AOALRTDF 221
AOALRTPE 221
AOAPPLDF 221
AOAPPLPE 221
AOCMDDF 221
AOEEXIT
 control statements 104, 105
AOEXECDF 221
AOIEXIT
 parameter 105
AOPARMDF 221
ARM
 See Automatic Restart Manager
ARMPOLCY parameter
 valid values 73
AUDITMSG

- AAOEXP00 parameter
 - modifying 51
- authorization 163–165
- AutoCustomization
 - parameter library 235
 - procedure library 238
 - profile library 237
 - sample library 236
- Automatic Restart Manager (ARM)
 - default policy 72
 - implementing support 72–73
 - parameter values 73
- Automation Reporter parameters
 - Dynamic Parameter Manager 42–43
- AutoOPERATOR 238
 - PATROL EM connection 121
- AutoOPERATOR 6.2
 - BMC Software products 8, 21
 - support
 - IBM products 8, 21
- AutoOPERATOR Access NV
 - checklist 115
 - creating
 - DSIPARM member DSICMD 117
 - DSIPARM member DSIDMN 117
 - DSIPARM member DSIOPF 117
 - DSIPARM member NAIOPT00 116
 - operator profiles 118
 - security 118
 - implementing 115–120
 - modifying
 - NAISTUB 116
 - NetView JCL 116
 - NETVIEW parameter 116
- AutoRULE
 - implementing 109–114

B

- BBI Control Command
 - DISPLAY 126
 - RESET 126, 127
 - START 127
 - STOP 127
- BBI-SS PAS
 - starting before JES 155
- BBI-SS PAS to CICS communication
 - available functions 89
 - modifying CICS tables 86
 - overview 85
 - processing program table (PPT) 86
 - program control table (PCT) 86
 - program list table (PLT) 87
 - restarting CICS target regions 88
 - Rule Processor 89
 - specifying Resources to CICS 86
 - startup JCL 88
- BBI-SS to CICS communication
 - available functions 89

- BROADCAST application 89
- BBKEYS
 - migrating
 - version 5.1. to 6.2 23
 - version 6.1. to 6.2 10
- BBLINK 103
- BBLINK, deleting old modules 98
- BBPARAM
 - AAOALS00 189
 - AAOALT00 189
 - AAOARP00 189
 - AAOEWD00 189
 - AAOEXP00 189
 - AAOGME00 189
 - AAOMQL00 189
 - AAONCD00 189
 - AAOPRM00 189
 - AAORUL00 189
 - AAOTRN00 189
 - AAOTSP00 189
- BBPARAM member
 - AAOGME00 199
- BBPARAM members 189–219
- BBPARAM/UBBPARM considerations 233
- BBPROF/SBBPROF considerations 233
- BBSAMP
 - member
 - IATUX18 133
 - IATUX31 133
 - JES2MAO 155
 - NAICMDS 117
 - NAIINIT 117
 - NAIINITX 117
 - NAIOPT00 116
 - NAIPROF 117
 - NAIPROFX 117
 - NAISVAR 117
 - SSJCL 156
- BBSAMP/UBBSAMP considerations 233
- BMC Software products
 - AutoOPERATOR 6.2 8, 21

C

- CAOTTAB 221
 - macro 92
 - invoking 91
 - parameters 93
- Changed field
 - Dynamic Parameter Manager 34
- changing transaction IDs
 - required ZAPs 182
- CICSTART 221
- CICSTERM 221
- Cmd field
 - Dynamic Parameter Manager 34
- CMRASM 167, 223
- CMRCSDES 223
- CMRDETL (detail performance file) 181

CMRPLT 223
 CMRSECU 223
 See also security
 action parameters 168
 implementing 167
 setting 168
 TYPE=ENTRY 167
 TYPE=FINAL 167
 TYPE=INITIAL 167
 CMRSTAT (statistical file) 181
 CMRUSECU 223
 CMRUSECU member 167
 command processing 104
 common storage 135
 CSA 135
 definition 135
 migrating
 version 5.1 to 6.2 20
 version 6.1 to 6.2 7
 SQA 135
 compatibility
 migrating
 version 5.1 to 6.2 21
 version 6.1 to 6.2 8
 version 5.1 to 6.2 21
 version 6.1 to 6.2 7
 compiled REXX programs 79
 configuration
 data set 233
 product libraries 234
 CONN
 GME parameter 126
 considerations for implementing CICS 85
 consoles
 allocating 65
 CSM application 65
 overview 65
 X-MCS
 allocating 65, 66
 Continuous State Manager
 calculating repository size 145
 Continuous State Manager
 accessing for the first time 140
 alternate database
 selecting 139
 creating a shared repository 143
 End-of-Memory EXEC 140
 EXECs 139
 migrating
 version 5.1 to 6.2 15
 version 6.1 to 6.2 4
 performance considerations 139
 Rule Set member
 selecting 139
 CPO
 migrating
 version 5.1 to 6.2 23
 version 6.1 to 6.2 10
 CPU time limit
 AutoOPERATOR EXECs
 TIMEXLIM 153
 CPU usage
 migrating
 version 5.1 to 6.2 20
 version 6.1 to 6.2 7
 Created field
 Dynamic Parameter Manager 34
 CSMALTDDB
 SSJCL parameter
 setting 139
 CSMBUILD 139
 CSMDOWN 139
 CSMEOM EXEC 140
 CSMINIT 139
 CSMUP 139
 Current field
 Dynamic Parameter Manager 47
 Current Parm field
 Dynamic Parameter Manager 63
 Current SSID field
 Dynamic Parameter Manager 63
 Current Target field
 Dynamic Parameter Manager 63

D
 data set
 configuration 233
 DATAID parameter 86
 DB2START 221
 DB2TERM 222
 DDNAMEs
 migrating
 version 5.1 to 6.2 19
 version 6.1 to 6.2 6
 DEBUGMSG
 AAOGME00 parameter
 modifying 199
 Defining AutoOPERATOR Resources to CICS 10, 23
 deleting BBLINK modules 98
 Delta IMS virtual terminal 224
 dependent services 89
 DFHTST TYPE=RECOVERY 86
 DFSAOE00 104
 DISP
 AAOARP00 parameter
 modifying 43
 DISPLAY command 126
 DIVDEF 222
 DIVUTIL 222
 DSICMD
 DSIPARM member
 NAICMDS 117
 NAIEXEC 117
 NAISTUB 117
 DSIDMN
 DSIPARM member
 NATASK 117

- DSIOPF
 - DSIPARM member
 - NAIPROF 117
 - NAIPROFX 117
- DTABJCL 222
- DUMP data set
 - BBI-SS PAS 156
- Dynamic Parameter Manager
 - activating modifications 62–63
 - fields
 - ALERTNV 39
 - AUDITMSG 51
 - Changed 34
 - Created 34
 - Current 47
 - EMABCNT 47
 - EXABCNT 47
 - EXEC 52
 - ID 34
 - Lib 34
 - MAXHIGH 47
 - MAXHIGHQ 48
 - MAXNORM 47
 - MAXNORMQ 48
 - MAXSTOR 38
 - MAXTPUT 49
 - Member 34, 47
 - OSPINUM 51
 - OSPIPRFX 51
 - PEREXLIM 51
 - PREFIX 51
 - REXX 52
 - SELLIM 49
 - Size 34
 - SUBXAUTH 52
 - TIMEXLIM 51
 - TSOTIME 51
 - UCPARMS 51
 - UNITNAME 51
 - VV.MM 34
 - WARNLVL1 38, 48
 - WARNLVL2 38, 49
 - modifying ALERTs thresholds in AAOALSxx 36–39
 - modifying Automation Reporter parameters in
 - AAOARPxx 39–43
 - modifying EXEC parameters in AAOEXPxx 44–53
 - modifying TapeSHARE parameters in AAOTSPxx 54–61
 - sample panels
 - ALERTs Storage Thresholds 37
 - Automation Reporter parameters 42
 - AutoOPERATOR Primary Option Menu 32
 - Confirm Parameter Activation 62
 - EXEC ParmS 50
 - EXEC Thresholds 45
 - setting high-priority EXECs 53
 - TapeSHARE parameters 56
 - setting
 - ALERTs storage thresholds 37

- Automation Reporter parameters 42
- EXEC parameters 50
- EXEC thresholds 45
- high-priority EXECs 53
- TapeSHARE parameters 56

E

- EMABCNT
 - AAOEXP00 parameter
 - modifying 47
- End-of-Memory (EOM) EXEC
 - Continuous State Manager 140
- EOM (End-of-Memory) EXEC
 - Continuous State Manager 140
- Event Activity Statistics panel
 - modifying 154
- EXABCNT
 - AAOEXP00 parameter
 - modifying 47
- EXEC 238
 - AAOEXP00 parameter
 - modifying 52, 53, 154
 - priority EXECs 154
 - threading EXECs 154
 - priority
 - Dynamic Parameter Manager 53
- REXX
 - enabling 153
 - runaway 153
 - loop detection 153
 - samples 235
- EXEC parameters
 - setting
 - Dynamic Parameter Manager 50–53
- EXEC thresholds
 - setting
 - Dynamic Parameter Manager 45–49
- EXECs
 - Continuous State Manager 139
 - CSMEOM 140
 - migrating
 - version 5.1. to 6.2 15
 - version 6.1. to 6.2 4
- exit routines 104
- expanding PROFILE pool data set 225
- Extended MCS consoles
 - See* X-MCS
- EXTRA
 - CAODTAB macro
 - parameter 93
- extrapartition transient data queue destinations
 - selected 92

F

- FCD2 transaction 181
- FIC2 transaction 181
- fields

- Dynamic Parameter Manager 34–63
 - ALERTs Storage Thresholds panel 39
 - Automation Reporter Parms panel 42–43
 - Confirm EXEC Update panel 63
 - EXEC PARMS panel 47–50, 51–53
 - TapeSHARE parameters panel 56–61
- FREE
 - AAOTSP00 parameter
 - modifying 61
- FST2 QOFF transaction 181
- FST2 QON transaction 181

G

- GME keyword
 - displaying connections 126
 - displaying local node status 126
 - displaying published subjects 126
 - displaying trace records 126
 - RECEIVE 126
- GMEACT
 - AAOGME00 parameter
 - modifying 199

H

- high-priority EXECs
 - Dynamic Parameter Manager 53
- HONORMPF parameter 205
- HOSTNAME
 - AAOGME00 parameter
 - modifying 199
- HST2 transaction (history display) 181

I

- IBAOUE3 224
- IBM products
 - support
 - AutoOPERATOR 6.2 8, 21
- ICOPY7 224
- ICOPY8 224
- ICOPY9 224
- ID field
 - Dynamic Parameter Manager 34
- IDEL31\$5 98
- IDEL32\$5 98
- IDEL32\$6 98
- IDEL33\$5 98
- IDEL33\$6 98
- IDEL33\$7 98
- identifying transactions 182
- IELOAD 105
- IMFEXEC CICS commands 90
- IMFSYS00 104, 105
- IMRUVTF 224
- IMSID 163
- IMSSTART 222
- IMSTERM 222

- independent IMFEXEC CICS commands 90
- initialization exit 104
- INTERVAL
 - AAOARP00 parameter
 - modifying 43
- INTRA
 - CAODTAB macro
 - parameter 93
- intrapartition transient data queue destinations
 - selecting 92

J

- JCL
 - samples 224
- JES
 - start after AutoOPERATOR 155
- JES2 job numbers
 - displaying 155
- JES2MAO 222
 - BBSAMP data set 155
- JES3 support 131
 - install for JES3 support 131
- JNL2 transaction (subsystem status) 181

L

- LAST SEG LOST 106
- Last Update field
 - Dynamic Parameter Manager 63
- Lib field
 - Dynamic Parameter Manager 34
- libraries
 - configuration 234
 - distributed target 233
 - parameter 235
 - procedure 235
 - product-user 233
 - profile 235
 - sample 235
 - site-customized 233
- LINKNUC 224
- loop detection
 - AutoOPERATOR EXECs 153
- LSTNPORT
 - AAOGME00 parameter
 - modifying 199

M

- manual customization
 - parameter library 235
 - IMS-specific 236
 - procedure library 238
 - sample library 237
- manual start 181
- manual stop 181
- MAXHIGH
 - AAOEXP00 parameter

- modifying 47, 154
- MAXHIGHQ
 - AAOEXP00 parameter
 - modifying 48
- MAXNORM
 - AAOEXP00 parameter
 - modifying 47, 154
- MAXNORMQ
 - AAOEXP00 parameter
 - modifying 48
- MAXSTOR
 - AAOALS00 parameter
 - modifying 38
- MAXTPUT
 - AAOEXP00 parameter
 - modifying 49
- MCTSIZE=xxxx
 - specifying 154
- Member
 - field
 - Dynamic Parameter Manager 47
- Member field
 - Dynamic Parameter Manager 34
- message counter table
 - defining 154
- messages
 - migrating
 - version 5.1. to 6.2 14
 - version 6.1. to 6.2 3
- migrating
 - version 5.1 to 6.2
 - BBKEYS 23
 - common storage 20
 - CPO 23
 - CPU usage 20
 - DDNAMEs 19
 - downward compatibility 21
 - EXECs 15
 - messages 14, 15
 - parameter values 19
 - parameters 15
 - private storage usage 20
 - Rules 15, 17
 - security 20
 - System Maintenance Program (SMP) 24
 - upward compatibility 21
 - variables 18
 - version 6.1 to 6.2
 - BBKEYS 10
 - common storage 7
 - CPO 10
 - CPU usage 7
 - DDNAMEs 6
 - downward compatibility 7
 - EXECs 4
 - messages 3, 4
 - parameter values 6
 - parameters 4
 - private storage usage 7

- Rules 4, 5
 - security 7
- System Maintenance Program (SMP) 11
 - upward compatibility 8
 - variables 6
- modifying ALERTs thresholds
 - Dynamic Parameter Manager 36–39
- modifying Automation Reporter parameters
 - Dynamic Parameter Manager 39–43
- modifying EXEC parameters
 - Dynamic Parameter Manager 44–53
- modifying TapeSHARE parameters
 - Dynamic Parameter Manager 54–61
- MPFFLTR parameter 205
 - implementing 76–78
- MQEV
 - parameters 206
- MQGINHIB
 - parameters 207
- MQNSHARE
 - parameters 207
- MQS Rules
 - compatibility between 5.1 and 6.2 17
 - compatibility between 6.1 and 6.2 5
- MSGLVLI keyword
 - BOTH 101
 - MTO (IMS master terminal) 101
 - NONE 101
 - WTO (OS/390 system console) 101
- MTO message capture 104
- MVS consoles
 - See* consoles, subsystem, X-MCS
- MVS subsystem consoles
 - See* consoles, subsystem, X-MCS

N

- NAICMDS 222
- NAIINIT 222
- NAIOPT00 222
 - DSIPARM member
 - OPERxx statements 116
- NAIPROF 222
- NAISTUB
 - AutoOPERATOR Access NV
 - executing 118
- NAISVAR 222
- NAITASK 222
- NetView
 - DSICLD data set
 - NAIINIT 117
 - NAIINITX 117
 - NAISVAR 117
 - JCL
 - modifying 116
- NETVIEW parameter
 - AutoOPERATOR Access NV
 - BBISSP00 120
- NEW DSN

- AAOARP00 parameter
 - modifying 43
- NOGIVE
 - AAOTSP00 parameter
 - modifying 59
- NOTAKE
 - AAOTSP00 parameter
 - modifying 59

O

- OFF INT
 - AAOARP00 parameter
 - modifying 43
- Open System Procedural Interface (OSPI) 81
 - ACB pool 81, 82
 - accessing VTAM applications 81
 - defining virtual terminals 81
 - IMFEXEC LOGON 81
 - naming conventions 82
 - OSPINUM 82
 - OSPIPRFX 82
 - terminal definitions 241
 - terminal macros 241
- OPID= 172
- origin 163
- OSPI
 - See* Open System Procedural Interface (OSPI)
- OSPINUM
 - AAOEXP00 parameter
 - modifying 51
- OSPIPRFX
 - AAOEXP00 parameter
 - modifying 51

P

- parameter library
 - AutoCustomization 235
 - description 235
 - IMS-specific 236
 - manual customization 235
 - IMS-specific 236
 - use 235
- parameter values
 - migrating
 - version 5.1 to 6.2 19
 - version 6.1 to 6.2 6
- parameters
 - BBPARAM 189
 - migrating
 - version 5.1. to 6.2 15
 - version 6.1. to 6.2 4
- PARTNER
 - AAOTSP00 parameter
 - modifying 57
- PATROL EM
 - 3270 printers 246
 - ACB 245

- ALERTs ??-129
- AutoOPERATOR connection 121
- BBI control commands 247
- data stream format 249
- establishing a VTAM connection 245
- implementing TCP/IP interface 121
- implementing VTAM connection 121
 - interface
 - controlling 126
 - implementing ??-129
- PEREXLIM
 - AAOEXP00 parameter
 - modifying 51
- PEREXLIM parameter
 - AAOEXP00 parameter
 - modifying 153
- performance considerations
 - CSM 139
- PM0229W warning message
 - BBI-SS PAS startup 155
- PREF
 - AAOTSP00 parameter
 - modifying 60
- PREFDEV
 - AAOTSP00 parameter
 - modifying 61
- PREFIX
 - AAOEXP00 parameter
 - modifying 51
- private storage usage
 - migrating
 - version 5.1 to 6.2 20
 - version 6.1 to 6.2 7
- procedure library
 - AutoCustomization 238
 - description 235
 - manual customization 238
 - use 238
- processing program table (PPT) 86
 - use of 181
- product libraries
 - configuration 234
 - site-customized 233
 - types 233
- profile library
 - AutoCustomization 237
 - description 235
 - use 237
 - user 233
- PROFILE pool
 - expanding data set 225
- program control table (PCT) 86
 - use of 181
- program list table (PLT) 87
- PUBLISH
 - GME parameter 126

R

RACF security 85
RECEIVE
 GME parameter 126
recoverable temporary storage 179
Replaced by field
 Dynamic Parameter Manager 63
reset
 CMRSECU table 167
RESET command 126, 127
RESLIB 224
RETRYCNT
 AAOTSP00 parameter
 modifying 58
RETRYINT
 AAOTSP00 parameter
 modifying 58
REXX
 AAOEXP00 parameter
 modifying 52
 support for EXECs 153
REXX/370 Alternate Library
 BBREXALT 79
 implementing 79
 SMP/E messages 79
Rule Set
 CSM application
 selecting 139
Rules
 migrating
 version 5.1. to 6.2 15, 17
 version 6.1. to 6.2 4, 5
rules processor authorization 165
RULESCAN parameter 205
RULESET parameter 205
RVARTST 222

S

sample JCL 224
sample library
 AutoCustomization 236
 description 235
 manual customization 237
 use 236
sample members
 \$EXECJOB 221
 \$EXECTST 221
 \$TESTDAT 221
 AOALRTDF 221
 AOALRTPE 221
 AOAPPLDF 221
 AOAPPLPE 221
 AOCMDDF 221
 AOEXECDF 221
 AOPARMDF 221
 CAOTTAB 221
 CICSTART 221
 CICSTERM 221

CMRSECU 223
CMRUSECU 223
DB2START 221
DB2TERM 222
DIVDEF 222
DIVUTIL 222
DTABJCL 222
IMSSTART 222
IMSTERM 222
JES2MAO 222
NAICMDS 222
NAIINIT 222
NAIOPT00 222
NAIPROF 222
NAISVAR 222
NAITASK 222
RVARTST 222
TTABJCL 222
SBBPROF/BBPROF considerations 233
security
 CMRASM member 167
 CMRSECU macro 167
 CMRSECU module 167
 CMRUSECU member 167
 considerations 85
 display parameters 168
 implementing 167
 migrating
 version 5.1. to 6.2 20
 version 6.1. to 6.2 7
 setting 168
security exit 159
SYSPROG 159
security for BBI
 BBPARM member 163–165
SELLIM
 AAOEXP00 parameter
 modifying 49
setting ALERTs storage thresholds
 Dynamic Parameter Manager 37–39
setting Automation Reporter thresholds
 Dynamic Parameter Manager 39–43
setting EXEC parameters
 Dynamic Parameter Manager 50–53
setting EXEC thresholds
 Dynamic Parameter Manager 45–49
setting high-priority EXECs
 Dynamic Parameter Manager 53
setting TapeSHARE thresholds
 Dynamic Parameter Manager 54–61
Size field
 Dynamic Parameter Manager 34
skeleton tailoring
 defined 263
SMN2 transaction 181
 manual start 181
SSJCL
 CSMALTDB parameter
 setting 139

- START
 - command 127
- STOP command
 - AutoOPERATOR EXECs 127
- stopping
 - AutoOPERATOR EXECs 127
- storage thresholds
 - ALERTs
 - Dynamic Parameter Manager 37–39
- SUB=MSTR
 - SSJCL command 156
- subsystem
 - consoles
 - allocating 65
 - CSM application 65
 - subsystem status 181
- SUBXAUTH
 - AAOEXP00 parameter
 - modifying 52
- SUPREPLY parameter 206
- syntax notation xix
- SYS1.HASPSRC
 - JES2 offsets 155
- SYSPROG
 - security exit 159
- System Maintenance Program (SMP)
 - migrating
 - version 5.1 to 6.2 24
 - version 6.1 to 6.2 11

T

- TapeSHARE
 - AAOTSP00 210
- TapeSHARE parameters
 - Dynamic Parameter Manager 54–61
 - implementing 69
 - activating 69
 - password keys 69
 - prerequisite 69
 - securing 70
- TARGET 163
- target libraries 233
- TCP/IP address
 - defining 124
 - multiple 125
- TCP/IP connection
 - establishing 123
 - prerequisites 123
- TCPIPJOB
 - AAOGME00 parameter
 - modifying 199
- temporary storage prefix
 - CMRI prefix ZAP 179
 - existing definition 179
- Temporary Storage Table (TST) 179
 - compatibility 86
- terminal
 - Delta IMS virtual 224

- terminal emulation
 - VTAM sessions
 - OSPI application 81
- terminating
 - AutoOPERATOR EXECs 127
- TGTHB
 - AAOGME00 parameter
 - modifying 199
- TGTIP
 - AAOGME00 parameter
 - modifying 200
- TGTMXMSL
 - AAOGME00 parameter
 - modifying 200
- TGTNAME
 - AAOGME00 parameter
 - modifying 200
- TGTPORT
 - AAOGME00 parameter
 - modifying 200
- TGTRTC
 - AAOGME00 parameter
 - modifying 200
- TGTRTI
 - AAOGME00 parameter
 - modifying 200
- TIMEOUT
 - AAOTSP00 parameter
 - modifying 57
- TIMEXLIM
 - AAOEXP00 parameter
 - modifying 51, 153
- TRACE
 - AAOTSP00 parameter
 - modifying 61
 - GME parameter 126
- TRACEAPP
 - AAOGME00 parameter
 - modifying 200
- TRACEBUF
 - AAOGME00 parameter
 - modifying 200
- TRACEGME
 - AAOGME00 parameter
 - modifying 201
- TRACELK
 - AAOGME00 parameter
 - modifying 201
- TRACESEC
 - AAOGME00 parameter
 - modifying 201
- TRACEZN
 - AAOGME00 parameter
 - modifying 201
- transaction IDs
 - description 181
 - FCD2 181
 - FIC2 181
 - FST2 QOFF 181

- FST2 QONN 181
- HST2 181
- JNL2 181
- required ZAPs for changed IDs 182
- SMN2 181
- TRANSEC parameter 85
- TSOTIME
 - AAOEXP00 parameter
 - modifying 51
- TST 179
- TTABJCL 222
- type-1 and type-2 AO exit routines 104

U

- UBBPARM/BBPARG considerations 233
- UBBSAMP/BBSAMP considerations 233
- UCPARMS
 - AAOEXP00 parameter
 - modifying 51
- unique message total
 - modifying 154
- UNITNAME
 - AAOEXP00 parameter
 - modifying 51
- User
 - libraries 233
 - profile 233
- user AO exit
 - DFSAOUE0 104
 - DFSAOUE1 104
- User field
 - Dynamic Parameter Manager 63
- USERID 163

V

- variables
 - migrating
 - version 5.1. to 6.2 18
 - version 6.1. to 6.2 6
- VTAM sessions
 - accessing
 - OSPI application 81
- VV.MM field
 - Dynamic Parameter Manager 34

W

- WARNLVL1
 - AAOALS00 parameter
 - modifying 38
 - AAOEXP00 parameter
 - modifying 48
- WARNLVL2
 - AAOALS00 parameter
 - modifying 38
 - AAOEXP00 parameter
 - modifying 49

X

- X-MCS
 - consoles
 - allocating 65, 66
- XTLOAD utility 225

Z

- ZAP 179
 - CMRI temporary storage prefix 179
- ZAPs for changed transaction IDs 182

STOP!

IMPORTANT INFORMATION - DO NOT INSTALL THIS PRODUCT UNLESS YOU HAVE READ ALL OF THE FOLLOWING MATERIAL

By clicking the YES or ACCEPT button below (when applicable), or by installing and using this Product or by having it installed and used on your behalf, You are taking affirmative action to signify that You are entering into a legal agreement and are agreeing to be bound by its terms, EVEN WITHOUT YOUR SIGNATURE. BMC is willing to license this Product to You ONLY if You are willing to accept all of these terms. CAREFULLY READ THIS AGREEMENT. If You DO NOT AGREE with its terms, DO NOT install or use this Product; press the NO or REJECT button below (when applicable) or promptly contact BMC or your BMC reseller and your money will be refunded if by such time You have already purchased a full-use License.

SOFTWARE LICENSE AGREEMENT FOR BMC PRODUCTS

SCOPE. This is a legally binding Software License Agreement ("**License**") between You (either an individual or an entity) and BMC pertaining to the original computer files (including all computer programs and data stored in such files) contained in the enclosed Media (as defined below) or made accessible to You for electronic delivery, if as a prerequisite to such accessibility You are required to indicate your acceptance of the terms of this License, and all whole or partial copies thereof, including modified copies and portions merged into other programs (collectively, the "**Software**"). "**Documentation**" means the related hard-copy or electronically reproducible technical documents furnished in association with the Software, "**Media**" means the original BMC-supplied physical materials (if any) containing the Software and/or Documentation, "**Product**" means collectively the Media, Software, and Documentation, and all Product updates subsequently provided to You, and "**You**" means the owner or lessee of the hardware on which the Software is installed and/or used. "**BMC**" means BMC Software Distribution, Inc. unless You are located in one of the following regions, in which case "BMC" refers to the following indicated BMC Software, Inc. subsidiary: (i) Europe, Middle East or Africa --BMC Software Distribution, B.V., (ii) Asia/Pacific -- BMC Software Asia Pacific Pte Ltd., (iii) Brazil -- BMC Software do Brazil, or (iv) Japan -- BMC Software K.K. If You enter into a separate, written software license agreement signed by both You and BMC or your authorized BMC reseller granting to you the rights to install and use this Product, then the terms of that separate, signed agreement will apply and this License is void.

FULL-USE LICENSE. Subject to these terms and payment of the applicable license fees, BMC grants You this non-exclusive License to install and use one copy of the Software for your internal use on the number(s) and type(s) of servers or workstations for which You have paid or agreed to pay to BMC or your BMC reseller the appropriate license fee. If your license fee entitles You only to a License having a limited term, then the duration of this License is limited to that term; otherwise this License is perpetual, subject to the termination provisions below.

TRIAL LICENSE. If You have not paid or agreed to pay to BMC or your BMC Reseller the appropriate license fees for a full use license, then, **NOTWITHSTANDING ANYTHING TO THE CONTRARY CONTAINED IN THIS LICENSE:** (i) this License consists of a non-exclusive evaluation license ("Trial License") to use the Product for a limited time ("Trial Period") only for evaluation; (ii) during the Trial Period, You may not use the Software for development, commercial, production, database management or other purposes than those expressly permitted in clause (i) immediately above; and (iii) your use of the Product is on an **AS IS** basis, and **BMC, ITS RESELLERS AND LICENSORS GRANT NO WARRANTIES OR CONDITIONS (INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE) TO YOU AND ACCEPT NO LIABILITY WHATSOEVER RESULTING FROM THE USE OF THIS PRODUCT UNDER THIS TRIAL LICENSE.** If You use this Product for other than evaluation purposes or wish to continue using it after the Trial Period, you must purchase a full-use license. When the Trial Period ends, your right to use this Product automatically expires, though in certain cases You may be able to extend the term of the Trial Period by request. Contact BMC or your BMC reseller for details.

TERM AND TERMINATION. This License takes effect on the first to occur of the date of shipment or accessibility to You for electronic delivery, as applicable (the "**Product Effective Date**"). You may terminate this License at any time for any reason by written notice to BMC or your BMC reseller. This License and your right to use the Product will terminate automatically with or without notice by BMC if You fail to comply with any material term of this License. Upon termination, You must erase or destroy all components of the Product including all copies of the Software, and stop using or accessing the Software. Provisions concerning Title and Copyright, Restrictions (or Restricted Rights, if You are a U.S. Government entity) or limiting BMC's liability or responsibility shall survive any such termination.

TITLE AND COPYRIGHT; RESTRICTIONS. All title and copyrights in and to the Product, including but not limited to all modifications thereto, are owned by BMC and/or its affiliates and licensors, and are protected by both United States copyright law and applicable international copyright treaties. You will not claim or assert title to or ownership of the Product. To the extent expressly permitted by applicable law or treaty notwithstanding this limitation, You may copy the Software only for backup or archival purposes, or as an essential step in utilizing the Software, but for no other purpose. You will not remove or alter any copyright or proprietary notice from copies of the Product. You acknowledge that the Product contains valuable trade secrets of BMC and/or its affiliates and licensors. Except in accordance with the terms of this License, You agree (a) not to decompile, disassemble, reverse engineer or otherwise attempt to derive the Software's source code from object code except to the extent expressly permitted by applicable law or treaty despite this limitation; (b) not to sell, rent, lease, license, sublicense, display, modify, time share, outsource or otherwise transfer the Product to, or permit the use of this Product by, any third party; and (c) to use reasonable care and protection to prevent the unauthorized use, copying, publication or dissemination of the Product and BMC confidential information learned from your use of the Product. **You will not export or re-export any Product without both the written consent of BMC and the appropriate U.S. and/or foreign government license(s) or license exception(s).** Any programs, utilities, modules or other software or documentation created, developed, modified or enhanced by or for You using this Product shall likewise be subject to these restrictions. BMC has the right to obtain injunctive relief against any actual or threatened violation of these restrictions, in addition to any other available remedies. Additional restrictions may apply to certain files, programs or data supplied by third parties and embedded in the Product; consult the Product installation instructions or Release Notes for details.

LIMITED WARRANTY AND CONDITION. If You have purchased a Full-Use License, BMC warrants that (i) the Media will be, under normal use, free from physical defects, and (ii) for a period of ninety (90) days from the Product Effective Date, the Product will perform in substantial accordance with the operating specifications contained in the Documentation that is most current at the Product Effective Date. BMC's entire liability and your exclusive remedy under this provision will be for BMC to use reasonable best efforts to remedy defects covered by this warranty

and condition within a reasonable period of time or, at BMC's option, either to replace the defective Product or to refund the amount paid by You to license the use of the Product. BMC and its suppliers do not warrant that the Product will satisfy your requirements, that the operation of the Product will be uninterrupted or error free, or that all software defects can be corrected. This warranty and condition shall not apply if: (i) the Product is not used in accordance with BMC's instructions, (ii) a Product defect has been caused by any of your or a third party's malfunctioning equipment, (iii) any other cause within your control causes the Product to malfunction, or (iv) You have made modifications to the Product not expressly authorized in writing by BMC. No employee, agent or representative of BMC has authority to bind BMC to any oral representations, warranties or conditions concerning the Product. **THIS WARRANTY AND CONDITION IS IN LIEU OF ALL OTHER WARRANTIES AND CONDITIONS. THERE ARE NO OTHER EXPRESS OR IMPLIED WARRANTIES OR CONDITIONS, INCLUDING THOSE OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, REGARDING THIS LICENSE OR ANY PRODUCT LICENSED HEREUNDER. THIS PARAGRAPH SHALL NOT APPLY TO A TRIAL LICENSE.** Additional support and maintenance may be available for an additional charge; contact BMC or your BMC reseller for details.

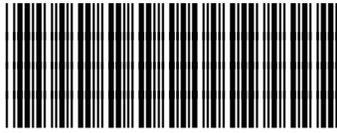
LIMITATION OF LIABILITY. Except as stated in the next succeeding paragraph, BMC's and your BMC reseller's total liability for all damages in connection with this License is limited to the price paid for the License. **IN NO EVENT SHALL BMC BE LIABLE FOR ANY CONSEQUENTIAL, SPECIAL, INCIDENTAL, PUNITIVE OR INDIRECT DAMAGES OF ANY KIND ARISING OUT OF THE USE OF THIS PRODUCT (SUCH AS LOSS OF PROFITS, GOODWILL, BUSINESS, DATA OR COMPUTER TIME, OR THE COSTS OF RECREATING LOST DATA), EVEN IF BMC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Some jurisdictions do not permit the limitation of consequential damages so the above limitation may not apply.

INDEMNIFICATION FOR INFRINGEMENT. BMC will defend or settle, at its own expense, any claim against You by a third party asserting that your use of the Product within the scope of this License violates such third party's patent, copyright, trademark, trade secret or other proprietary rights, and will indemnify You against any damages finally awarded against You arising out of such claim. However, You must promptly notify BMC in writing after first receiving notice of any such claim, and BMC will have sole control of the defense of any action and all negotiations for its settlement or compromise, with your reasonable assistance. BMC will not be liable for any costs or expenditures incurred by You without BMC's prior written consent. If an order is obtained against your use of the Product by reason of any claimed infringement, or if in BMC's opinion the Product is likely to become the subject of such a claim, BMC will at its option and expense either (i) procure for You the right to continue using the product, or (ii) modify or replace the Product with a compatible, functionally equivalent, non-infringing Product, or (iii) if neither (i) nor (ii) is practicable, issue to You a pro-rata refund of your paid license fee(s) proportionate to the number of months remaining in the 36 month period following the Product Effective Date. This paragraph sets forth your only remedies and the total liability to You of BMC, its resellers and licensors arising out of such claims.

GENERAL. This License is the entire understanding between You and BMC concerning this License and may be modified only in a mutually signed writing between You and BMC. If any part of it is invalid or unenforceable, that part will be construed, limited, modified, or severed so as to eliminate its invalidity or unenforceability. This License will be governed by and interpreted under the laws of the jurisdiction named below, without regard to conflicts of law principles, depending on which BMC Software, Inc. subsidiary is the party to this License: (i) BMC Software Distribution, Inc. - the State of Texas, U.S.A., (ii) BMC Software Distribution, B.V. - The Netherlands, (iii) BMC Software Asia Pacific Pte Ltd. -- Singapore (iv) BMC Software do Brazil -- Brazil, or (v) BMC Software K.K. -- Japan. Any person who accepts or signs changes to the terms of this License promises that they have read and understood these terms, that they have the authority to accept on your behalf and legally obligate You to this License. Under local law and treaties, the restrictions and limitations of this License may not apply to You; You may have other rights and remedies, and be subject to other restrictions and limitations.

U.S. GOVERNMENT RESTRICTED RIGHTS. UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in FAR Section 52.227-14 Alt. III (g)(3), FAR Section 52.227-19, DFARS 252.227-7014 (b) or DFARS 227.7202, as amended from time to time. Contractor/Manufacturer is BMC Software, Inc., 2101 CityWest Blvd., Houston, TX 77042-2827, USA. Any contract notices should be sent to this address.

Notes



100042414