MainView Customization Reference

Supporting

Version 6.1 of MainView Infrastructure
MainView products

March 2013
Contacting BMC Software

Several methods are available for contacting BMC Software.

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

United States and Canada

<table>
<thead>
<tr>
<th>Address</th>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC SOFTWARE INC 2101 CITYWEST BLVD HOUSTON TX 77042-2827 USA</td>
<td>1 713 918 8800  or 1 800 841 2031</td>
<td>1 713 918 8000</td>
</tr>
</tbody>
</table>

Outside United States and Canada

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

© Copyright 2010-2013 BMC Software, Inc.

BMC, BMC Software, and the BMC Software logo are the exclusive properties of BMC Software, Inc., are registered with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries. All other BMC trademarks, service marks, and logos may be registered or pending registration in the U.S. or in other countries. All other trademarks or registered trademarks are the property of their respective owners.

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

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

Linux is the registered trademark of Linus Torvalds.

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

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

Restricted rights legend

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 UNPUBLISHED—RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES.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

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

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

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

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

- Product information
  - Product name
  - Product version (release number)
  - License number and password (trial or permanent)
- Operating system and environment information
  - Machine type
  - Operating system type, version, and service pack or other maintenance level such as PUT or PTF
  - System hardware configuration
  - Serial numbers
  - Related software (database, application, and communication) including type, version, and service pack or maintenance level
- Sequence of events leading to the problem
- Commands and options that you used
- Messages received (and the time and date that you received them)
  - Product error messages
  - Messages from the operating system, such as file system full
  - Messages from related software
License key and password information

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

- Send an e-mail message to customer_support@bmc.com. (In the Subject line, enter SupID:yourSupportContractID, such as SupID:12345.)
- In the United States and Canada, call 1 800 537 1813. Outside the United States and Canada, contact your local support center for assistance.
# Contents

## About this book
- Related publications ................................................................. 13
- Conventions ............................................................................. 14

## Chapter 1   Overview of MainView Customization  
- Customization methods ............................................................. 15
- MainView architecture ............................................................... 16
  - Coordinating address spaces .................................................. 16
  - Product address spaces ......................................................... 17
  - User address spaces ............................................................ 18
  - Runtime Component System (RTCS) ...................................... 19

## Chapter 2   MainView product libraries  
- Library types ........................................................................... 21
- Distributed libraries used as models during customization .......... 22
- Customized libraries ............................................................... 24
- Link library .............................................................................. 25
- Parameter libraries ................................................................. 26
  - BBPARM .............................................................................. 27
  - UBBPARM ........................................................................... 27
  - Customized IMS parameters .................................................. 27
- Sample library .......................................................................... 28
  - BBSAMP ................................................................................ 29
  - UBBSAMP ............................................................................. 29
- Profile libraries .......................................................................... 30
  - BBPROF ............................................................................... 31
  - SBBPROF ............................................................................. 31
  - User BBPROF ....................................................................... 31
- Procedure libraries ................................................................... 32
  - BBPROC ............................................................................... 32
  - UBBPROC ............................................................................. 32
- View screen configuration and Help libraries ............................. 33
  - Site libraries .......................................................................... 34
  - User libraries ......................................................................... 35
  - Shared libraries ...................................................................... 36
  - Library concatenation ............................................................ 36
## Chapter 3 Manual customization

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of manual customization tasks</td>
<td>45</td>
</tr>
<tr>
<td>Specifying jobcards and other operational defaults</td>
<td>56</td>
</tr>
<tr>
<td>Generating JCL to run the installation verification program</td>
<td>57</td>
</tr>
<tr>
<td>Implementing Katakana terminal support</td>
<td>58</td>
</tr>
<tr>
<td>Apply passwords with the Product Authorization Utility</td>
<td>60</td>
</tr>
<tr>
<td>Creating site data sets</td>
<td>60</td>
</tr>
<tr>
<td>Creating site security data set</td>
<td>62</td>
</tr>
<tr>
<td>Adding the BMC load library to your system APF list</td>
<td>63</td>
</tr>
<tr>
<td>Adding the BMC load library to your system link list</td>
<td>64</td>
</tr>
<tr>
<td>Creating a CLIST for invoking MainView products</td>
<td>65</td>
</tr>
<tr>
<td>Adding TSF authorization for products to run under ISPF</td>
<td>68</td>
</tr>
<tr>
<td>Configure RTCS</td>
<td>69</td>
</tr>
<tr>
<td>Creating the CAS startup procedure</td>
<td>69</td>
</tr>
<tr>
<td>Allocating the CASPERM registry data set</td>
<td>71</td>
</tr>
<tr>
<td>Creating MainView host server startup procedure</td>
<td>72</td>
</tr>
<tr>
<td>Allocate MainView Infrastructure common registry</td>
<td>74</td>
</tr>
<tr>
<td>Before allocating the MainView Infrastructure common registry</td>
<td>75</td>
</tr>
<tr>
<td>Allocating the MVI common registry with BBSAMP members</td>
<td>76</td>
</tr>
<tr>
<td>Defining MainView Alarm Manager initialization parameters</td>
<td>77</td>
</tr>
<tr>
<td>Allocating MainView Alarm Manager journal logs</td>
<td>80</td>
</tr>
<tr>
<td>Allocating MainView Alarm Manager DIV data set</td>
<td>81</td>
</tr>
<tr>
<td>Creating MainView Alarm Manager startup procedure</td>
<td>82</td>
</tr>
<tr>
<td>Migrating BBMTYB00 (CAS definitions)</td>
<td>84</td>
</tr>
<tr>
<td>Verification of MainView Logger data sets</td>
<td>85</td>
</tr>
<tr>
<td>Allocating the register data set BBSAMP member LOGRREG</td>
<td>86</td>
</tr>
<tr>
<td>Allocating the DIV data set BBSAMP member LOGRDIV</td>
<td>87</td>
</tr>
<tr>
<td>Creating MainView Logger LOGGERxx PARMLIB member</td>
<td>88</td>
</tr>
<tr>
<td>Creating the MainView Logger started task procedure</td>
<td>90</td>
</tr>
<tr>
<td>Setting up a product-specific logspace</td>
<td>91</td>
</tr>
<tr>
<td>Allocating the register data set BBSAMP member LSALCREG</td>
<td>91</td>
</tr>
<tr>
<td>Allocating the DIV data set BBSAMP member LSALCDIV</td>
<td>92</td>
</tr>
<tr>
<td>Creating LOGGERxx PARMLIB member with BBPARM member</td>
<td>92</td>
</tr>
<tr>
<td>LSLOGGER</td>
<td>92</td>
</tr>
<tr>
<td>Creating a started task procedure for BBSAMP member LSLOGSTC</td>
<td>94</td>
</tr>
<tr>
<td>Historical data set determination</td>
<td>95</td>
</tr>
<tr>
<td>Before creating historical data sets</td>
<td>96</td>
</tr>
</tbody>
</table>
Allocating historical data set BBILIB member @@YZZ051 ...........................................97
Creating a z/OS PAS PARMFILE data set ........................................................................100
Creating the z/OS PAS startup procedure ......................................................................101
Creating a BBI-SS PAS PARMFILE data set ..................................................................103
Defining BBI-SS PAS suffixes and target system parameters .........................................105
Defining configuration members to BBI-SS PAS ............................................................111
Defining BBI-SS PAS interval services parameters .........................................................114
Setting the BBI-SS PAS historical recording interval ......................................................117
Allocating BBI-SS PAS journal and image logs ..............................................................118
Allocating MainView Total Object Manager data sets .....................................................120
Allocating MainView AutoOPERATOR data sets ............................................................121
Creating a BBI-SS PAS startup procedure .....................................................................121
Adding the CICS EXCI library to the BBI-SS startup procedure .......................................126
Creating the MainView Total Object Manager start procedure ......................................127
Creating automatic BBI-SS PAS log handling procedures ............................................128
Defining terminal session parameters ............................................................................131
Define cross-system communication .............................................................................136
  Planning for CAS-to-CAS communication ....................................................................140
  Planning for BBI-SS PAS-to-BBI-SS PAS communication ..........................................144
  Define BBI-SS PAS-to-BBI-SS PAS communication ....................................................147
  Define CAS-to-CAS communication ............................................................................153
Specifying product initialization within a BBI-SS PAS ....................................................164
Enabling Automatic Restart Manager (ARM) support ....................................................166
Adding the CAS ID to the BBI-SS PAS ............................................................................166
Adding extended function data sets to the BBI-SS PAS startup procedure .....................167
Update command processor table ...................................................................................169
Specifying product initialization within the z/OS PAS ....................................................171

Appendix A  CAS library member 173
CAS started task procedure statements ...........................................................................173
CAS PROC statement parameters ....................................................................................174
  procname parameter ......................................................................................................175
  SSID parameter ...........................................................................................................175
  CAPS parameter .........................................................................................................176
  COLD parameter .........................................................................................................176
  COMMORDR parameter ..............................................................................................177
  CONVXCF parameter .................................................................................................178
  DFLTGRP parameter .................................................................................................178
  DUMP parameter .......................................................................................................178
  EMM parameter ..........................................................................................................179
  XDM parameter ..........................................................................................................180
### Appendix B  
**z/OS PAS library member**

<table>
<thead>
<tr>
<th>Statement</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS PAS started task procedure statements</td>
<td>181</td>
</tr>
<tr>
<td>z/OS PAS PROC statement parameters</td>
<td>186</td>
</tr>
<tr>
<td>BAVSHLQ parameter</td>
<td>187</td>
</tr>
<tr>
<td>BBCHILV parameter</td>
<td>187</td>
</tr>
<tr>
<td>BBLINK parameter</td>
<td>187</td>
</tr>
<tr>
<td>BBSOUT parameter</td>
<td>187</td>
</tr>
<tr>
<td>CASWAIT parameter</td>
<td>188</td>
</tr>
<tr>
<td>CMDID parameter</td>
<td>188</td>
</tr>
<tr>
<td>CPM parameter</td>
<td>189</td>
</tr>
<tr>
<td>CXEN parameter</td>
<td>189</td>
</tr>
<tr>
<td>DC parameter</td>
<td>190</td>
</tr>
<tr>
<td>EM parameter</td>
<td>191</td>
</tr>
<tr>
<td>IPM parameter</td>
<td>191</td>
</tr>
<tr>
<td>procnme parameter</td>
<td>192</td>
</tr>
<tr>
<td>RGN parameter</td>
<td>192</td>
</tr>
<tr>
<td>SSID parameter</td>
<td>192</td>
</tr>
<tr>
<td>SYSID parameter</td>
<td>193</td>
</tr>
<tr>
<td>TZOVRLAP parameter</td>
<td>193</td>
</tr>
<tr>
<td>UBBParm parameter</td>
<td>194</td>
</tr>
<tr>
<td>USEZIIP parameter</td>
<td>194</td>
</tr>
<tr>
<td>XDM parameter</td>
<td>194</td>
</tr>
<tr>
<td>XDS parameter</td>
<td>195</td>
</tr>
</tbody>
</table>

### Appendix C  
**BBI-SS PAS library members**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBIBBCFG parameters</td>
<td>199</td>
</tr>
<tr>
<td>AAOALS parameter</td>
<td>200</td>
</tr>
<tr>
<td>AAOARP parameter</td>
<td>201</td>
</tr>
<tr>
<td>AAOEXP parameter</td>
<td>201</td>
</tr>
<tr>
<td>AAOGME parameter</td>
<td>201</td>
</tr>
<tr>
<td>AAOMQL parameter</td>
<td>201</td>
</tr>
<tr>
<td>AAOPRM parameter</td>
<td>202</td>
</tr>
<tr>
<td>AAOTSP parameter</td>
<td>202</td>
</tr>
<tr>
<td>BBIISP parameter</td>
<td>202</td>
</tr>
<tr>
<td>BBISSP parameter</td>
<td>202</td>
</tr>
<tr>
<td>BBKMON parameter</td>
<td>203</td>
</tr>
<tr>
<td>CMRBEX parameter</td>
<td>203</td>
</tr>
<tr>
<td>CMRDTL parameter</td>
<td>203</td>
</tr>
<tr>
<td>CMRPRB parameter</td>
<td>204</td>
</tr>
<tr>
<td>CMRSOP parameter</td>
<td>204</td>
</tr>
</tbody>
</table>
CMRTTH parameter .............................................................................................................204
BBIISPxx parameters ........................................................................................................204
  MAXREQ parameter ........................................................................................................205
  INTERVAL parameter ......................................................................................................205
  IRRI parameter .............................................................................................................206
  IRSYNCH parameter ......................................................................................................206
  AUTOID parameter ........................................................................................................207
  TARGET BLK and USRID parameters ...............................................................................207
  TRDIR and SUBSYS parameters .................................................................................208
BBIISPxx parameters for the BBI-SS PAS environment .....................................................208
  CASID parameter ........................................................................................................209
  CMDCHAR parameter ...................................................................................................210
  CMDPREF parameter ..................................................................................................210
  DAE parameter ............................................................................................................210
  DUMPCLAS parameter ...............................................................................................211
  DUMPS parameter .......................................................................................................211
  JESCNCT parameter ...................................................................................................211
  JOURNAL parameter ..................................................................................................212
  JRNLMSG parameter .................................................................................................212
  JRNLSTC parameter ...................................................................................................213
  LOGGING parameter ...................................................................................................213
  MAXPQE parameter .....................................................................................................213
  MAXUSER parameter ...................................................................................................214
  MVLOG parameter .......................................................................................................214
  MVLOGSS parameter .....................................................................................................215
  PRODUCT parameter .....................................................................................................215
  STALL parameter ..........................................................................................................215
BBIISPxx parameters for MainView AutoOPERATOR .........................................................216
  ALRTRCVE parameter .................................................................................................217
  ARMPOLICY parameter ...............................................................................................217
  AUDIT parameter ..........................................................................................................218
  CMDCON parameter ....................................................................................................218
  CMDWAIT parameter ...................................................................................................219
  CNSLEXEC parameter .................................................................................................219
  CONSOLES parameter ...............................................................................................219
  HILITE parameter .........................................................................................................220
  JESCHAR parameter ......................................................................................................220
  JES3CHAR parameter ....................................................................................................220
  JES3CMD parameter ....................................................................................................221
  JESFLTR parameter .......................................................................................................221
  JESTYPE parameter ......................................................................................................222
MCTSIZE parameter ................................................................. 222
MIGCONS parameter ................................................................. 222
NETVIEW parameter ................................................................. 223
PROFUSED parameter ............................................................... 223
XCFGROUP parameter ............................................................... 223
BBISSPxx parameter for MainView AutoOPERATOR for WebSphere MQ and MainView for WebSphere MQ ......................................................... 224
BBISSPxx parameter for MainView AutoOPERATOR for CICS and MainView for CICS ........................................................................ 225
BBISSPxx parameter for MainView for CICS MainView for DB2 MainView for DBCTL and MainView for IMS Online ...................................................... 225
BBIVARxx parameters ................................................................. 225
BBIVARxx processing rules ......................................................... 226
BBIVARxx syntax rules ............................................................... 227
BBTCP00 parameters ................................................................. 228
TCPNAME parameter ............................................................... 229
INFOMSG parameter ............................................................... 229
DEBUG parameter ................................................................. 230
BBPROF data set members ......................................................... 230
xxxCYC or CYCnn member ......................................................... 231
BBICMDS member ................................................................. 231
BBIPFK member ................................................................. 232
BBITSP00 member ............................................................... 232
LOGPFK member ............................................................... 232
xxxPFK member ................................................................. 233
System variables substitution in parameter library members .......... 233

Appendix D MainView Logger library members 237
MainView Logger started task procedure statements ........................ 237
MainView Logger initialization parameters .................................. 239
MVSSYSTEM parameter ......................................................... 240
LOGCYLINDERS or LOGTRACKS parameter .............................. 241
LOGPREFIX parameter ......................................................... 242
REGISTERDSN parameter ...................................................... 242
DIVDSN parameter ............................................................. 242
COMPRESSION parameter ..................................................... 243
DEFAULTLOGGER parameter ................................................. 243
DIVBLOCKS parameter ........................................................... 243
DUMPCOUNTMAX parameter ................................................ 244
LOGSSID parameter ............................................................. 244
LOGSPACE parameter .......................................................... 245
Appendix E  MainView Alarm Manager version 2.1 PAS library member  249

Appendix F  Migration of MainView Infrastructure and MainView products  253

Migrating to a new version of MainView Infrastructure  .................................................. 253
Migrating CAS definitions  ............................................................................................. 254
Running mixed versions of MainView Infrastructure ..................................................... 254
Migrating MainView products to a new system image  .................................................. 258
Migrating the MainView environment ............................................................................ 258
Migrating specific products ............................................................................................ 264
About this book

This book contains detailed information about customizing your MainView environment manually.

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

Note

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

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

Related publications

From the BMC Support Central website (http://www.bmc.com/support), you can use the following methods to access related publications that support your product or solution:

- Link to the BMC Documentation Center (https://webapps.bmc.com/infocenter/index.jsp) to browse documentation sets
- View BMC Quick Course Demos (short overviews of selected product concepts, tasks, or features), which are included in the BMC Documentation Center
- Read individual product documents (books and notices) within the “A – Z Supported Product List”

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

This document uses the following special conventions:

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

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

- This document uses a symbol to show menu sequences. For example, `Actions => Create Test` instructs you to choose the `Create Test` command from the `Actions` menu.
Overview of MainView Customization

This reference guide helps you customize your MainView environment to your specifications:

- Describes the customization methods that are available to set up the MainView environment, and briefly explains the MainView architecture
- Describes the libraries used in the MainView environment
  The MainView libraries are used by all of the customization methods. See MainView product libraries on page 21.
- Contains detailed information about manually customizing the MainView environment
  The detailed information is useful as a reference for any of the customization methods. See Manual customization on page 45.

Customization methods

The following methods are available to customize MainView Infrastructure and products:

- Customization within the Installation System (the recommended method):
  — OZI Customization for MainView Infrastructure and the Runtime Component System (RTCS)
  — MainView Customization for MainView products

For more information about these customization methods, see the Installation System User Guide.
MainView architecture

MainView is an integrated family of performance management and automation products that monitor and control traditional and parallel mainframes. MainView comprises performance monitors, automated operations, and automation applications.

MainView product integration provides host system monitoring and automation (even in remote locations) through a common terminal session, using the MainView Selection Menu. The integration of MainView products is provided through intercommunications technology known as BBI.

BBI integrates the MainView performance products within a common communications framework that operates across multiple machines in multiple locations. This integrated architecture allows a single terminal session, using one or more MainView products, to monitor and manage multiple local or remote targets.

For maximum flexibility and extensibility, MainView architecture includes built-in separation of the data, application, and end-user dimensions of systems management. BBI communications, data collection, and the end-user terminal session run in three distinct address spaces:

- Coordinating address space (CAS)
- Product address space (PAS)
- User address space (UAS)

Coordinating address spaces

The coordinating address space (CAS) runs as a subsystem and is used by most of the MainView products. The CAS manages communication with other CASs on other local and remote systems, and allows direct communication between an individual terminal session and a product address space.

Usually, one CAS runs per system image. A single CAS can communicate with an unlimited number of remote systems and CASs.
A product establishes an independent connection with its local CAS, so you can add new products or new upgrades to the architecture without affecting existing products or other configurations.

Each CAS contains a component called Plex Manager that provides administration and operations views that help you:

- Manage communication links with other CASs
- Monitor the activity of accessible products
- Create single system image (SSI) contexts
- Control security for products

### Product address spaces

The product address space (PAS) runs as a started task and comprises special routines, including data collectors, to support one or more MainView products.

**Table 1: Types of PASs**

<table>
<thead>
<tr>
<th>PAS type</th>
<th>Products supported</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS PAS</td>
<td>■ CMF MONITOR</td>
<td>One z/OS PAS runs per system image. The z/OS PAS always connects to the CAS on that system image.</td>
</tr>
<tr>
<td></td>
<td>■ MainView for z/OS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ MainView for UNIX System Services</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ MainView SYSPROG Services</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ MainView VistaPoint (for IBM MVS workloads)</td>
<td></td>
</tr>
<tr>
<td>PAS type</td>
<td>Products supported</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BBI-SS PAS</td>
<td>■ MainView AutoOPERATOR&lt;br&gt;■ MainView FOCAL POINT&lt;br&gt;■ MainView for CICS&lt;br&gt;■ MainView for DB2&lt;br&gt;■ MainView for DBCTL&lt;br&gt;■ MainView for IMS Online&lt;br&gt;■ MainView for WebSphere MQ&lt;br&gt;■ MainView VistaPoint (for CICS, DB2, DBCTL, and IMS workloads)</td>
<td>Multiple instances of the BBI-SS PAS can run on a single system image and run one or more products. Depending on the products that are installed, the BBI-SS PAS might or might not connect to a CAS on that system image. BBI-SS PASs on local and remote systems are linked together to provide cross-system communication for an individual terminal session.</td>
</tr>
<tr>
<td>Product-specific PAS</td>
<td>■ MainView for IP&lt;br&gt;■ MainView for Linux - Servers&lt;br&gt;■ MainView for VTAM&lt;br&gt;■ MainView for WebSphere Application Server&lt;br&gt;■ MainView Storage Resource Manager (SRM)&lt;br&gt;■ MainView Transaction Analyzer</td>
<td>For MainView for Linux - Servers, multiple instances of the PAS can run on a single system image. In the case of MainView for Linux - Servers, each PAS can monitor up to 500 Linux images. You can run multiple PASs to support the number of Linux images that you plan to monitor. For MainView for IP, MainView SRM, MainView Transaction Analyzer, and MainView for VTAM, only one product-specific PAS can be active for each product on a system image.</td>
</tr>
</tbody>
</table>

**User address spaces**

Each terminal session runs in its own user address space (UAS). A terminal session provides the end-user session for all MainView products.

The terminal session connects to a CAS (if a CAS is available), to a BBI-SS PAS, or to both.
Table 2: Types of UASs

<table>
<thead>
<tr>
<th>UAS type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MainView Host Gateway using MainView Explorer</td>
<td>Using the MainView Explorer web browser, your terminal session can access MainView products from a personal computer.</td>
</tr>
<tr>
<td>TSO address space</td>
<td>Using a TSO address space, your terminal session can access MainView products and perform other TSO/ISPF functions.</td>
</tr>
<tr>
<td>VTAM or EXCP address space using MainView Alternate Access</td>
<td>Using a separate address space that communicates with your terminal through either VTAM or EXCP, your terminal session can access MainView products and also perform other ISPF functions.</td>
</tr>
</tbody>
</table>

Runtime Component System (RTCS)

RTCS provides programming services to all CASs, PASs, and UASs. It runs as a started task that starts soon after a system IPL occurs.

RTCS has no user interfaces (other than console commands to modify its operation), operates continuously, and seldom if ever needs to be stopped.

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

MainView product libraries

This section identifies the types of libraries used in the MainView environment, explains the libraries to use when you make changes to a product, and explains how the libraries are created.

Note
Of the target libraries that are distributed with MainView products, this section describes only those libraries that are changed by customization. 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 version or applies product maintenance.

Library types

MainView provides several types of product libraries, as follows:

- Distributed target libraries
  Created during product installation, these libraries contain load modules, parameters, procedures, samples, views, screen definitions for views, and view Help.

  WARNING
  Never modify these libraries without SMP/E procedures.

- Site-customized product libraries
  These libraries contain versions of distributed library members that have been modified to your site requirements. You can let MainView Customization create these libraries for you, or create them manually.

- Product user libraries
  Each user can have his or her own version of views, screen definitions containing views, and view Help in a library with that user's ID. Depending on what products are installed, a user profile called uprefix userid BBPROF is created during terminal session initialization (if a profile does not exist already).
- Image and journal logs

Some products that run in the BBI-SS PAS use these logs for recording screen images or messages. You can let MainView Customization create these logs for you, or create them manually.

- Product support data sets

These data sets are required for some product functions. You can let MainView Customization create these data sets for you, or create them manually.

### Distributed libraries used as models during customization

You can use the distributed libraries as models to create your own site-customized product libraries.

Table 3 on page 22 shows the MainView distributed libraries from which customization can create your site-customized product libraries. For a complete list of the product libraries, see Product library customization summary on page 38.

**WARNING**

You should never change the distributed libraries themselves. *All changes that are described in this guide should be made only to site-customized product libraries.* If you make changes in the distributed libraries, subsequent SMP maintenance will overwrite those changes.

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIST library</td>
<td>BBCLIB</td>
<td>Contains MainView Customization CLIST and other CLISTs that specific MainView products use.</td>
</tr>
<tr>
<td>View Help library</td>
<td>BBHELP</td>
<td>Contains online Help text for products that provide views.</td>
</tr>
<tr>
<td>Link library</td>
<td>BBLINK</td>
<td>For MainView environment load modules and product-specific load modules.</td>
</tr>
<tr>
<td>Message library</td>
<td>BBMLIB</td>
<td>Contains messages and abend codes that are issued by MainView products.</td>
</tr>
<tr>
<td>Type</td>
<td>Name</td>
<td>Contents</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Parameter library</td>
<td>BBPARM</td>
<td>Contains parameter definitions for related products and product-specific parameter definitions.</td>
</tr>
<tr>
<td>Panel library</td>
<td>BBPLIB</td>
<td>Contains the ISPF panel definitions for the terminal session.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> User-customized panels should be placed in a separate library and concatenated in front of the distributed panel library.</td>
</tr>
<tr>
<td>Procedure library</td>
<td>BBPROC</td>
<td>Contains MainView AutoOPERATOR execute command lists (EXECs), which are user-defined automated applications.</td>
</tr>
<tr>
<td>Profile library</td>
<td>BBPROF</td>
<td>For products that run in full-screen mode, this library contains members that define a user’s:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ PF keys</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Target system defaults</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Primary Option Menu</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Unique application profiles</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Cycle refresh definitions</td>
</tr>
<tr>
<td>Sample library</td>
<td>BBSAMP</td>
<td>Contains macros, sample JCL, and sample user exit routines.</td>
</tr>
<tr>
<td>Screen library</td>
<td>BBSDEF</td>
<td>Contains screen definitions consisting of one or more views.</td>
</tr>
<tr>
<td>Table library</td>
<td>BBTLIB</td>
<td>Contains MainView product tables, MainView Customization tables, and some MainView product command tables.</td>
</tr>
<tr>
<td>View library</td>
<td>BBVDEF</td>
<td>Contains views that are used by products that operate in windows mode.</td>
</tr>
</tbody>
</table>
Customized libraries

Your customized libraries depend on the MainView products that are installed at your site.

**Note**
During the installation and customization process, you select whether to use runtime libraries or SMP/E libraries. If you use runtime libraries, your customized library names will start with BMC. If you use SMP/E libraries, your customized library names will start with BB, UBB, or TOSZ. A few library names start with IBB or SBB for runtime or SMP/E libraries. For a list of data set names, see *Product library customization summary* on page 38.

Throughout the MainView documentation, the BB*, UBB*, and TOSZ* data set names are used even if the actual library names start with BMC*.

Customized libraries include some or all of the following libraries:

- Your own versions of the distributed libraries shown in Table 3 on page 22

- Image and journal logs
  The image log is for screen images from timer-driven service requests. The journal log is for BBI-SS PAS and product commands, responses, and messages.
  These logs are used by the MainView AutoOPERATOR, MainView for CICS, MainView for DB2, MainView for DBCTL, and MainView for IMS Online products.

- The following product support data sets:
<table>
<thead>
<tr>
<th>Data set</th>
<th>Description</th>
<th>Products that use data set</th>
</tr>
</thead>
</table>
| BBPARM member BBIBBCFG | Identifies BBPARM configuration member suffixes to the BBI-SS PAS | — MainView AutoOPERATOR  
— MainView FOCAL POINT  
— MainView for CICS  
— MainView for DB2  
— MainView for DBCTL  
— MainView for IMS Online  
— MainView for WebSphere MQ  
— MainView VistaPoint |
| BBIDIV | Stores binary large objects and data across product restarts | MainView AutoOPERATOR and MainView Alarm Manager version 2.1 |
| BBIVARS | Functions as a profile variable pool | MainView AutoOPERATOR |
| PARMFILE | Stores workload configuration definitions | MainView for z/OS and for products that support historical data processing with the TIME command Data set |

Related Information

■ “Product library customization summary” on page 38

Link library

This library is concatenated in the target system's STEPLIB DD statements for some MainView products.

The link library must be authorized. Executable modules are obtained from this library. If this library is not specified, an error message is generated.
During the installation and customization process, you select whether to use runtime libraries or SMP/E libraries. If you use runtime libraries, your customized library names will start with **BMC**. If you use SMP/E libraries, your customized library names will start with **BB**, **UBB**, or **TOSZ**. A few library names start with **IBB** or **SBB** for runtime or SMP/E libraries.

Throughout the MainView documentation, the **BB***, **UBB***, and **TOSZ*** data set names are used even if the actual library names start with **BMC***.

### Related Information

- “Product library customization summary” on page 38

---

**Parameter libraries**

Each member in the parameter library contains parameters for a separate product or group of related products.

Your site can have several parameter libraries, including a distributed version and one or more site-customized versions.

Multiple parameter library data sets can be concatenated together. For example, a site parameter library can be created and a data set that is customized for each PAS can be concatenated before the site library.

Some parameter library members support the substitution of system variables.

**Note**

During the installation and customization process, you select whether to use runtime libraries or SMP/E libraries. If you use runtime libraries, your customized library names will start with **BMC**. If you use SMP/E libraries, your customized library names will start with **BB**, **UBB**, or **TOSZ**. A few library names start with **IBB** or **SBB** for runtime or SMP/E libraries.

Throughout the MainView documentation, the **BB***, **UBB***, and **TOSZ*** data set names are used even if the actual library names start with **BMC***.

### Related Information

- “Product library customization summary” on page 38
- “System variables substitution in parameter library members” on page 233
BBPARM

BBPARM is a distributed target library that is created during product installation.

**WARNING**

This version of the parameter library should never be modified. Subsequent SMP maintenance will overwrite any changes that you make. The distributed name is used in this guide for reference only.

UBBPARM

UBBPARM contains copies of members from the BBPARM distributed parameter library. This customized library is created automatically by MainView Customization, or you can create it manually.

Use the UBBPARM data set to make any parameter library changes that are described in this guide.

MainView Customization creates a UBBPARM data set as `hlq.IMAGE.UBBPARM` (for example, `hlq.IMAGSYSDD.UBBPARM`). This method allows each system image to have its own BBPARM data set for unique parameters.

A single UBBPARM data set, however, can be shared among multiple systems. BMC recommends sharing a single UBBPARM when unique parameters are not required.

Customized IMS parameters

You can define unique parameters for individual IMS systems, either by copying and renaming specific members and including them in UBBPARM or by creating user-defined parameter data sets (referred to as *IBBPARM data sets*).

This topic applies to the following products only:

- MainView AutoOPERATOR for IMS
- MainView for DBCTL
- MainView for IMS Online
- MainView for IMS Offline

For more information about customizing parameters for individual IMS systems, see the product-specific customization guides.
Customized IMFSYS00, IMFECP00, and AAOPRM00 members

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 system and the BBI-SS PAS. The name of an IMS-specific system parameter member must be in the following format:

\textit{imsidSYS} (where \textit{imsid} is the four-character IMS system identifier)

You can also include a copied, renamed version of IMFECP00 to customize the Event Collector parameters for an IMS system. The name of an IMS-specific Event Collector parameter member must be in the following format:

\textit{imsidECP}

You can include an AAOPRMxx member by using the AAOPRM parameter in the IMFSYS00 (or \textit{imsidSYS}) member to specify the suffix.

If you do not need to customize other UBBPARM members for an IMS system, you do not need to create and allocate a separate IBBPARM parameter data set for that system.

User-defined parameter data set

You can allocate an IBBPARM parameter data set to contain any members that you want to make unique to an IMS system.

\textit{Note}

The IBBPARM data set must be allocated manually; it is not created through MainView Customization.

If you do not need to customize any parameter members other than IMFSYS00, IMFECP00, or AAOPRM00 for an IMS system, an IBBPARM data set is not required. Instead, you can include renamed versions of the members in UBBPARM, as described in “Customized IMFSYS00, IMFECP00, and AAOPRM00 members” on page 28.

Sample library

This library provides macros, sample JCL, and sample user exit routines.
The members in the sample library contain:

- Sample JCL that can be edited and submitted to perform specified functions
- Macros that are referenced during assembly of user-written services
- Sample user exit routines

Your site can have several sample libraries, including a distributed version and one or more site-customized versions. Some members are for more than one product and some members are product specific.

**Note**
During the installation and customization process, you select whether to use runtime libraries or SMP/E libraries. If you use runtime libraries, your customized library names will start with BMC. If you use SMP/E libraries, your customized library names will start with BB, UBB, or TOSZ. A few library names start with IBB or SBB for runtime or SMP/E libraries.

Throughout the MainView documentation, the BB*, UBB*, and TOSZ* data set names are used even if the actual library names start with BMC*.

**Related Information**

- "Product library customization summary" on page 38

---

**BBSAMP**

BBSAMP is a distributed target library that is created during product installation.

**WARNING**
This version of the sample library should never be modified. Subsequent SMP maintenance will overwrite any changes that you make. The distributed name is used in this guide for reference only.

---

**UBBSAMP**

UBBSAMP contains copies of members from the distributed sample library. This customized library is created automatically by MainView Customization, or you can create it manually.

Use the UBBSAMP data set to make any changes to sample members that are described in this guide or to make changes to meet your site’s needs.
Profile libraries

Your site can have several profile libraries, including a distributed version and one or more site-customized versions. Members in this library contain profile information and cycle refresh definitions. Other members are created dynamically.

**WARNING**
Do not change any members in the profile library unless instructed to do so.

**Tip**
This topic applies to the following products only:
- MainView AutoOPERATOR
- MainView for CICS
- MainView for DB2
- MainView for DBCTL
- MainView for IMS Online

You can have a site profile library and a user profile library. The site library can be created automatically by MainView Customization, or you can create it manually. The site library contains a common profile that is shared by all site users.

The MainView CLIST creates a user profile automatically (if a profile 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
- Unique set of application profiles

The user profiles and site profile should be concatenated before the distributed profile. When a profile is saved, it is stored in the first profile library that is defined in the concatenation.
During the installation and customization process, you select whether to use runtime libraries or SMP/E libraries. If you use runtime libraries, your customized library names will start with BMC. If you use SMP/E libraries, your customized library names will start with BB, UBB, or TOSZ. A few library names start with IBB or SBB for runtime or SMP/E libraries.

Throughout the MainView documentation, the BB*, UBB*, and TOSZ* data set names are used even if the actual library names start with BMC*.

---

**Note**

**Related Information**

- “Product library customization summary” on page 38

---

**BBPROF**

BBPROF is a distributed target library that is created during product installation.

---

**WARNING**

This version of the profile library should never be modified. Subsequent SMP maintenance will overwrite any changes that you make. The distributed name is used in this guide for reference only.

---

**SBBPROF**

SBBPROF is an optional data set. It is created automatically by MainView Customization, or you can create it manually.

Use SBBPROF to make any changes that you want to be shared by all users at your site.

---

**User BBPROF**

A user BBPROF data set contains profile members that are customized by a user.

A profile data set should exist for each user so that each user has an individual application profile. The MainView CLIST creates a user profile automatically (if a profile does not exist already). This profile is called userid.BPROF, where userid is the user's logon ID.
Procedure libraries

Your site can have several procedure libraries, including a distributed version and one or more site-customized versions.

**Tip**
This topic applies to MainView AutoOPERATOR only.

Members in this library contain executable procedures that are used by MainView AutoOPERATOR. These procedures are execute command lists (EXECs) that automate site functions. For more information about EXECs, see the MainView AutoOPERATOR product documentation.

**Note**
During the installation and customization process, you select whether to use runtime libraries or SMP/E libraries. If you use runtime libraries, your customized library names will start with BMC. If you use SMP/E libraries, your customized library names will start with BB, UBB, or TOSZ. A few library names start with IBB or SBB for runtime or SMP/E libraries. Throughout the MainView documentation, the BB*, UBB*, and TOSZ* data set names are used even if the actual library names start with BMC*.

**Related Information**
- "Product library customization summary" on page 38

**BBPROC**

BBPROC is a distributed target library that is available when MainView AutoOPERATOR is installed.

**WARNING**
This version of the procedure library should never be modified. Subsequent SMP maintenance will overwrite any changes that you make. The distributed name is used in this manual for reference only.

**UBBPROC**

This library contains new user-written EXECs or customized EXECs from the distributed BBPROC library. This customized library is created automatically by MainView Customization, or you can create it manually.
If you used MainView Customization when you installed MainView AutoOPERATOR (as described in the Installation System User Guide), MainView Customization created a UBBPROC data set for you.

**To use a specific EXEC sample**

1. Verify that the member was not already copied to UBBPROC by MainView Customization.

2. Copy the sample member that you need from the distributed BBPROC library to the UBBPROC data set.

3. Make the change in the copied member.

**To customize MainView AutoOPERATOR manually**

1. Create your own UBBPROC data set.

2. Copy the members that you need from the distributed BBPROC library into the UBBPROC data set that you created.

3. Change the members that you copied in UBBPROC.

---

**View screen configuration and Help libraries**

This topic applies to MainView products that operate in windows mode only and describes the windows mode libraries.

For MainView products that operate in windows mode only, the types of libraries are as follows:

- **Views**
  The SMP target view library is allocated in the CAS and PAS startup procedure. This library contains a uniquely named set of views for each product. View tables allow multiple products to have the same view names. For example, MainView for z/OS has one version of the VIEWS view, while CMF MONITOR has a completely different version of VIEWS; they are kept separated in the library by being stored in tables with unique names.

- **Screen definitions**
  A set of screen definitions is allocated by the MainView CLIST.

- **Configuration**
  Configuration definitions are used by MainView Explorer.
Help text
A set of members that contains online Help text for products that provide views.

**Note**
During the installation and customization process, you select whether to use runtime libraries or SMP/E libraries. If you use runtime libraries, your customized library names will start with BMC. If you use SMP/E libraries, your customized library names will start with *BB*, *UBB*, or *TOSZ*. A few library names start with *IBB* or *SBB* for runtime or SMP/E libraries. Throughout the MainView documentation, the *BB*, *UBB*, and *TOSZ* data set names are used even if the actual library names start with *BMC*.

**Related Information**
- “Product library customization summary” on page 38

**Site libraries**
The following site libraries are created for you during MainView Customization:

- View library, allocated as *hlq.SBBVDEF*
- Screen definition library, allocated as *hlq.SBBSDEF*
- Configuration library, allocated as *hlq.SBBCDEF*

The first *S* in *SBBVDEF*, *SBBSDEF*, and *SBBCDEF* represents *site*.

When the MainView CLIST is used to access the MainView Selection Menu, these data sets are allocated to the user address space (UAS) by using the following DD names: *BBVDEF*, *BBSDEF*, and *BBCDEF*.

**WARNING**
A system administrator should retain sole authority for adding and deleting views, screen definitions, configuration, or Help text in site libraries. Granting write authority to more than one user can result in overwriting valuable data. Users who want to contribute to a site library should contact the system administrator.

If you have an external security system (such as IBM RACF, or Computer Associates CA ACF2 or CA Top Secret), you can use it to restrict change authority to the system administrator's user ID.
User libraries

User libraries contain customized views, screen definitions, configurations, or Help text that are available to an individual user only.

If a user does not have access to a user library, any changes that the user makes are stored in the site library by default (unless the system administrator has prevented write access to the site library). A user library should be created for each user at your site.

To create a user library

1. Create a standard partitioned data set for a user view, screen definition, configuration, and Help text library.

Observe the following guidelines:

- Make the data set a fixed block data set with LRECL=80.
- Use the following naming convention:
  
  uprefix.userid.BBVDEF | BBSDEF | BBCDEF | BBHELP

  The variables are as follows:

  — uprefix is a user’s TSO prefix. (You can use the UPREFIX parameter in the MainView CLIST to specify any prefix that you want. This parameter is useful for users who have more than one TSO ID and want to use the same user BBxDEF data set with all of their IDs.)

  — userid is a user’s TSO ID.

  — BBVDEF is a view library.

  — BBSDEF is a screen definition library.

  — BBCDEF is a configuration library used by MainView Explorer.

  — BBHELP is a Help text library.

2. Change the user's allocations to include user and site data sets.

   Ensure that the user library is searched before the site library.

   **Note**

   If the MainView CLIST is used to access the MainView Selection Menu, the CLIST concatenates the view and screen libraries. The user Help text library must be added to the concatenation manually.
You can add as many data sets to the concatenation as your site allows. For example, you might want to create libraries that only people in certain departments can access. Then, all people in a given department would name the same data sets on their BBxDEF concatenation.

Shared libraries

User libraries that are allocated to a UAS are shared by all products that provide views.

Therefore, if you have more than one of these products installed on your system, use caution when saving views and screen definitions.

To avoid possible confusion, you should establish a naming convention at both the site and user levels (for example, C xxxxxxx for CMF MONITOR views and M xxxxxxxx for MainView for z/OS views).

Example
Suppose you create a MainView for z/OS view called MYVIEW and save it in your user view library. You then access CMF MONITOR and create another view, also called MYVIEW. When you try to save the second MYVIEW, a message warns you that MYVIEW already exists because you created the MainView for z/OS MYVIEW previously. If you save the second version anyway, the first version of MYVIEW is overwritten.

Library concatenation

When views, screen definitions, or Help text are changed or created, they are saved in the following order by default:

1 User library (if one exists)

2 Site library (if one exists)

If neither library exists, an error message is displayed.

Note
The distributed view library cannot be overwritten because it is allocated to a PAS, not a UAS.
When views, screen definitions, or Help text are requested, the libraries are searched in the following order by default:

1. Your user library (if one exists)
2. Your site library (if one exists)
3. The distributed library (for views and Help text only)

Image and journal logs

Two image and two journal logs are allocated in the BBI-SS PAS startup procedure for use by the following products:

- MainView AutoOPERATOR (journal logs only)
- MainView for CICS
- MainView for DB2
- MainView for DBCTL
- MainView for IMSOnline
- MainView for WebSphere MQ (journal logs only)
- MainView Alarm Manager version 2.1 (journal logs only)

Note
During the installation and customization process, you select whether to use runtime libraries or SMP/E libraries. If you use runtime libraries, your customized library names will start with BMC. If you use SMP/E libraries, your customized library names will start with BB, UBB, or TOSZ. A few library names start with IBB or SBB for runtime or SMP/E libraries.

Throughout the MainView documentation, the BB*, UBB*, and TOSZ* data set names are used even if the actual library names start with BMC*.

The BBI-SS PAS image log records screen images that are produced automatically by timer-driven analyzer and monitor services. Image logging can be disabled by removing the appropriate statement from the BBI-SS PAS startup procedure.

The BBI-SS PAS journal log records

- all commands and responses that are issued from a terminal session that is assigned to the BBI-SS PAS
all commands and responses that are issued automatically by MainView AutoOPERATOR EXECs

time stamps for startup and shutdown of the BBI-SS PAS and the target system

BBI-SS PAS informational, error, and audit messages

service commands and messages

DB2 commands and messages

Because all operational and diagnostic messages are written to the BBI-SS PAS journal log, BMC recommends that you avoid disabling the log. If necessary, however, BBI-SS PAS journal logging can be disabled by removing appropriate DD statements from the BBI-SS PAS startup procedure. BBI-SS PASs cannot share journal data sets.

Note
Image and journal logs, as well as a screen print data set, are allocated to a user's terminal session by the MainView CLIST.

Related Information

“Product library customization summary” on page 38

Product library customization summary

This topic lists the MainView environment libraries and product libraries, and summarizes how they should be configured. It also indicates what these libraries are named before and after customization.

The following configuration requirements apply to product libraries:

- User and site-customized libraries should be concatenated before the distributed libraries.

- IMS and CICS target allocation of the BBLINK link library requires IMS and CICS JCL modifications. UAS allocation of the link library uses a DD name of BBLOAD and BBILINK.

- IMS and CICS target allocation of the BBPARM parameter library requires IMS and CICS JCL modifications.

- Help text for views is distributed as a BBLINK load module.
Table 4 on page 39 summarizes the MainView environment libraries and how they should be configured. Table 5 on page 43 summarizes the MainView product libraries and how they should be configured. It also lists the names of the libraries before and after customization.

Table 4: MainView environment libraries

<table>
<thead>
<tr>
<th>Distributed target library name</th>
<th>Name after customizing for SMP/E target-zone data sets</th>
<th>Created by MainView Customization?</th>
<th>Allocated to which address space?</th>
<th>Allocated to what DD name?</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>&amp;VSPREFIX..&amp;SSID.. BBIDIV</td>
<td>Yes, MainView Customization creates this data set for MainView AutoOPERATOR and MainView Alarm Manager (version 2.1).</td>
<td>BBI-SS PAS</td>
<td>BBIDIV</td>
</tr>
<tr>
<td>None</td>
<td>&amp;PRDPREFIX..&amp;SSID.. BBIIMAG n</td>
<td>Yes, MainView Customization creates dual image log data sets for:</td>
<td>BBI-SS PAS</td>
<td>BBIIMAG1, BBIIMAG2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ MainView for CICS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ MainView for DB2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ MainView for DBCTL</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ MainView for IMS Online</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Chapter 2  MainView product libraries  39
<table>
<thead>
<tr>
<th>Distributed target library name</th>
<th>Name after customizing for SMP/E target-zone data sets</th>
<th>Created by MainView Customization?</th>
<th>Allocated to which address space?</th>
<th>Allocated to what DD name?</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>&amp;PRDPREFIX..&amp;SSID.. BBIJRNL n</td>
<td>Yes, MainView Customization creates dual journal log data sets for:</td>
<td>PAS</td>
<td>BBIJRNL1, BBIJRNL2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• MainView AutoOPERATOR</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• MainView for CICS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• MainView for DB2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• MainView for DBCTL</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• MainView for IMS Online</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• MainView for WebSphere MQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• MainView Alarm Manager (version 2.1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>None</td>
<td>&amp;PRDPREFIX..&amp;SSID.. BBIVARS</td>
<td>Yes, MainView Customization creates this data set but for MainView AutoOPERATOR only.</td>
<td>BBI-SS PAS</td>
<td>BBIVARS</td>
</tr>
<tr>
<td>None</td>
<td>&amp;PRDPREFIX..&amp;SYSNAME.. BBSECURE</td>
<td>Yes, MainView Customization creates a hlq.BBSECURE site security interface library. The MainView CLIST allocates it.</td>
<td>all PASs and CAS</td>
<td>BBSECURE</td>
</tr>
<tr>
<td>None</td>
<td>&amp;UPREFIX..BBIIMAGL</td>
<td>No, MainView Customization does not create a user image log. It is created by the MainView CLIST.</td>
<td>UAS</td>
<td>BBIIMAGL</td>
</tr>
<tr>
<td>None</td>
<td>&amp;UPREFIX..BBIJRNLL</td>
<td>No, MainView Customization does not create a user journal log. It is created by the MainView CLIST.</td>
<td>UAS</td>
<td>BBIJRNLL</td>
</tr>
<tr>
<td>Distributed target library name</td>
<td>Name after customizing for SMP/E target-zone data sets</td>
<td>Created by MainView Customization?</td>
<td>Allocated to which address space?</td>
<td>Allocated to what DD name?</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------</td>
<td>-----------------------------------</td>
<td>----------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>None</td>
<td>&amp;VSPREFIX..&amp;SSID.. PARMFILE</td>
<td>Yes, MainView Customization creates a PARMFILE dynamic history file management file.</td>
<td>PAS</td>
<td>PARMFILE</td>
</tr>
<tr>
<td>BBCLIB</td>
<td>&amp;PRDPREFIX..&amp;SYSNAME.. SBBCLIB</td>
<td>Yes, MainView Customization creates an SBBCLIB CLIST library.</td>
<td>UAS</td>
<td>BBCLIB</td>
</tr>
<tr>
<td>BBHELP</td>
<td>&amp;UPREFIX..BBHDEF</td>
<td>No, &amp;UPREFIX..BBHDEF is a user-defined Help library that must be created manually. If it exists, the MainView CLIST allocates it.</td>
<td>UAS</td>
<td>BBHDEF</td>
</tr>
<tr>
<td></td>
<td>&amp;PRDPREFIX..&amp;SYSNAME.. SBBHDEF</td>
<td>No, MainView Customization does not create a hlq.SBBHELP site Help library. You must create it manually. If it exists, the MainView CLIST allocates it.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BBPARM</td>
<td>UBBPARM &amp;PRDPREFIX..&amp;SYSNAME..UBBPARM</td>
<td>Yes, MainView Customization creates a UBBPARM parameter library.</td>
<td>PAS CAS LAS/VTAS</td>
<td>PARMLIB, BBIPARM, BBVPARM, or SVOSPARM</td>
</tr>
<tr>
<td></td>
<td>&amp;PRDPREFIX..&amp;SYSNAME..MVAPARM</td>
<td>Yes, MainView Customization creates an MVAPARM parameter library.</td>
<td>Alarm Manager 2.1 PAS</td>
<td>BBIPARM</td>
</tr>
<tr>
<td>BBPARM member BBIBBCFG</td>
<td>&amp;PRDPREFIX..&amp;SYSNAME..UBBPARM member CFG ssid</td>
<td>Yes, MainView Customization creates the UBBPARM member.</td>
<td>BBI-SS PAS</td>
<td>BBCFG</td>
</tr>
<tr>
<td>BBPROC</td>
<td>&amp;PRDPREFIX..&amp;SYSNAME..UBBPROC</td>
<td>Yes, MainView Customization creates a UBBPROC procedure library but for MainView AutoOPERATOR only.</td>
<td>BBI-SS PAS</td>
<td>SYSPROC</td>
</tr>
<tr>
<td>Distributed target library name</td>
<td>Name after customizing for SMP/E target-zone data sets</td>
<td>Created by MainView Customization?</td>
<td>Allocated to which address space?</td>
<td>Allocated to what DD name?</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------------------</td>
<td>-----------------------------------</td>
<td>---------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>BBPROF</td>
<td>&amp;UPREFIX..BBPROF</td>
<td>No, MainView Customization does not create a user profile. It is created by the MainView CLIST.</td>
<td>UAS</td>
<td>BBIPROF</td>
</tr>
<tr>
<td></td>
<td>&amp;PRDPREFIX..&amp;SYSNAME..SBBPROF</td>
<td>Yes, MainView Customization creates an SBBPROF site profile.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BBSAMP</td>
<td>&amp;PRDPREFIX..&amp;SYSNAME..UBBSAMP</td>
<td>Yes, MainView Customization creates a UBBSAMP sample library.</td>
<td>LAS/VTAS</td>
<td>SYSPROC</td>
</tr>
<tr>
<td>BBSDEF</td>
<td>&amp;UPREFIX..BBSDEF</td>
<td>No, MainView Customization does not create a user screen definition library. It is created by the MainView CLIST.</td>
<td>UAS</td>
<td>BBSDEF</td>
</tr>
<tr>
<td></td>
<td>&amp;PRDPREFIX..&amp;SYSNAME..SBBVDEF</td>
<td>Yes, MainView Customization creates a hlq.SBBVDEF site screen definition library. The MainView CLIST allocates it.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BBVDEF</td>
<td>&amp;UPREFIX..BBVDEF</td>
<td>No, MainView Customization does not create a user view definition library. It is created by the MainView CLIST.</td>
<td>UAS</td>
<td>BBVDEF</td>
</tr>
<tr>
<td></td>
<td>&amp;PRDPREFIX..&amp;SYSNAME..SBBVDEF</td>
<td>Yes, MainView Customization creates a hlq.SBBVDEF site view library. The MainView CLIST and various PASs allocate it.</td>
<td>PAS</td>
<td>SBBVDEF</td>
</tr>
</tbody>
</table>
### Table 5: MainView product libraries

<table>
<thead>
<tr>
<th>Distributed target library name</th>
<th>Name after customizing for SMP/E target-zone data sets</th>
<th>Created by MainView Customization?</th>
<th>Allocated to which address space?</th>
<th>Allocated to what DD name?</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>&amp;PRDPREFIX..&amp;SSID..DATA</td>
<td>Yes, MainView Customization creates a MainView for Linux data file.</td>
<td>MainView PAS</td>
<td>MMLDATA</td>
</tr>
<tr>
<td>None</td>
<td>&amp;PRDPREFIX..&amp;SSID..IMAGE</td>
<td>Yes, MainView Customization creates a MainView for Linux image file.</td>
<td>MainView PAS</td>
<td>MMLIMAGE</td>
</tr>
<tr>
<td>None</td>
<td>&amp;PRDPREFIX..&amp;SYSNAME..SBBCDF</td>
<td>Yes, MainView Customization creates an SBBCDF Explorer configuration library.</td>
<td>MainView Explorer PAS</td>
<td>SBBCDF</td>
</tr>
<tr>
<td>None</td>
<td>&amp;PRDPREFIX..&amp;SYSNAME..UBBPARM member MMLPRM&amp;SYSCLONE</td>
<td>Yes, MainView Customization creates a MainView for Linux parameter member.</td>
<td>MainView PAS</td>
<td>SYSIN</td>
</tr>
<tr>
<td>None</td>
<td>&amp;VSPREFIX..&amp;SYSNAME..SGPRDFn</td>
<td>Yes, MainView Customization creates a set of MainView SRM Performance Collector Database files.</td>
<td>MainView PAS</td>
<td>SGPHDBnn</td>
</tr>
<tr>
<td>None</td>
<td>&amp;VSPREFIX..&amp;SYSNAME..SGPRDF1</td>
<td>Yes, MainView Customization creates a MainView SRM Performance Collector Index file.</td>
<td>MainView PAS</td>
<td>SGPHDSNI</td>
</tr>
<tr>
<td>Distributed target library name</td>
<td>Name after customizing for SMP/E target-zone data sets</td>
<td>Created by MainView Customization?</td>
<td>Allocated to which address space?</td>
<td>Allocated to what DD name?</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------------------------------------------------------</td>
<td>-----------------------------------</td>
<td>----------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>BBSLIB</td>
<td>&amp;PRDPREFIX..&amp;SYSNAME..UBBSLIB</td>
<td>Yes, MainView Customization creates a MainView SRM skeleton library.</td>
<td>MainView SRM PAS</td>
<td>BBSLIB</td>
</tr>
<tr>
<td>None</td>
<td>&amp;VSPREFIX..ZS&amp;SYSCLONE..CPMOUTn</td>
<td>Yes, MainView Customization creates dual CPMOUT CMF output data sets.</td>
<td>MainView for z/OS PAS</td>
<td>CMFCPMn</td>
</tr>
<tr>
<td>None</td>
<td>&amp;VSPREFIX..ZS&amp;SYSCLONE..HST1DSnn</td>
<td>Yes, MainView Customization creates MainView for z/OS long term history data sets.</td>
<td>MainView for z/OS PAS</td>
<td>HST1DSnn</td>
</tr>
<tr>
<td>None</td>
<td>&amp;VSPREFIX..ZS&amp;SYSCLONE..HST1DSnn</td>
<td>Yes, MainView Customization creates MainView for DB2 (version 11.1.00 and later) data sets.</td>
<td>BBI-SS PAS</td>
<td>HST1DSnn</td>
</tr>
<tr>
<td>None</td>
<td>&amp;VSPREFIX..ZS&amp;SYSCLONE..IPMOUTn</td>
<td>Yes, MainView Customization creates dual IPMOUT CMF alternate output data sets.</td>
<td>MainView for z/OS PAS</td>
<td>CMFIPMn</td>
</tr>
</tbody>
</table>
Manual customization

This topic describes the tasks for customizing the MainView environment manually. You can also use this information as a reference when using MainView Customization.

You can use manual customization for all MainView products and MainView Infrastructure, which includes Alarm Management, Alert Manager, and MainView Logger.

Note
During installation and online customization, you select whether to use runtime libraries or SMP/E libraries. If you use runtime libraries, your customized library names will start with BMC. If you use SMP/E libraries, your customized library names will start with BB, UBB, or TOSZ. A few library names start with IBB or SBB for runtime or SMP/E libraries. For a list of data set names, see Product library customization summary on page 38. MainView documentation uses the BB*, UBB*, and TOSZ* data set names, but your names might start with BMC*.

Overview of manual customization tasks

This topic lists the manual customization tasks that you can complete to customize your MainView environment to best suit your needs. Manual customization is time-consuming but provides the greatest flexibility.

For each manual customization task that you can perform, Table 6 on page 45 indicates whether the task is required or optional.

<table>
<thead>
<tr>
<th>Task</th>
<th>Required or optional?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specifying jobcards and other operational defaults on page 56</td>
<td>Required</td>
</tr>
<tr>
<td>Generating JCL to run the installation verification program on page 57</td>
<td>Optional</td>
</tr>
<tr>
<td>Task</td>
<td>Required or optional?</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Implementing Katakana terminal support on page 58</td>
<td>Optional</td>
</tr>
<tr>
<td>Apply passwords with the Product Authorization Utility on page 60</td>
<td>Required, except not applicable to MainView FOCAL POINT or MainView SRM</td>
</tr>
<tr>
<td>Creating site data sets on page 60</td>
<td>Required</td>
</tr>
<tr>
<td>Creating site security data set on page 62</td>
<td>Required, except not applicable to MainView FOCAL POINT</td>
</tr>
<tr>
<td>Adding the BMC load library to your system link list on page 64</td>
<td>Optional</td>
</tr>
<tr>
<td>Adding the BMC load library to your system link list on page 64</td>
<td>Optional</td>
</tr>
<tr>
<td>Creating a CLIST for invoking MainView products on page 65</td>
<td>Required</td>
</tr>
<tr>
<td>Adding TSF authorization for products to run under ISPF on page 68</td>
<td>Required for the following products, but not applicable to others:</td>
</tr>
<tr>
<td>■ MainView FOCAL POINT</td>
<td></td>
</tr>
<tr>
<td>■ MainView AutoOPERATOR</td>
<td></td>
</tr>
<tr>
<td>■ MainView for CICS</td>
<td></td>
</tr>
<tr>
<td>■ MainView for DB2</td>
<td></td>
</tr>
<tr>
<td>■ MainView for DBCTL</td>
<td></td>
</tr>
<tr>
<td>■ MainView for IMSOnline</td>
<td></td>
</tr>
<tr>
<td>■ MainView for IMSOffline</td>
<td></td>
</tr>
<tr>
<td>■ MainView for WebSphere MQ</td>
<td></td>
</tr>
<tr>
<td>Configure RTCS on page 69</td>
<td>Required, except not applicable to MainView FOCAL POINT</td>
</tr>
<tr>
<td>Creating the CAS startup procedure on page 69</td>
<td>Required, except not applicable to MainView FOCAL POINT</td>
</tr>
<tr>
<td>Creating MainView host server startup procedure on page 72</td>
<td>Optional, except not applicable to MainView FOCAL POINT</td>
</tr>
<tr>
<td>Allocate MainView Infrastructure common registry on page 74</td>
<td>Required</td>
</tr>
<tr>
<td>Defining MainView Alarm Manager initialization parameters on page 77</td>
<td>Optional, except not applicable to MainView FOCAL POINT</td>
</tr>
<tr>
<td>Task</td>
<td>Required or optional?</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Allocating MainView Alarm Manager journal logs on page 80</td>
<td>Optional, except not applicable to MainView FOCAL POINT</td>
</tr>
<tr>
<td>Allocating MainView Alarm Manager DIV data set on page 81</td>
<td>Optional, except not applicable to MainView FOCAL POINT</td>
</tr>
<tr>
<td>Creating MainView Alarm Manager startup procedure on page 82</td>
<td>Optional, except not applicable to MainView FOCAL POINT</td>
</tr>
<tr>
<td>Migrating BBMTYB00 (CAS definitions) on page 84</td>
<td>Optional, except not applicable to MainView FOCAL POINT</td>
</tr>
<tr>
<td>Verification of MainView Logger data sets on page 85</td>
<td>Required, except not applicable to MainView FOCAL POINT</td>
</tr>
<tr>
<td>Creating MainView Logger LOGGERxx PARMLIB member on page 88</td>
<td>Required, except not applicable to MainView FOCAL POINT</td>
</tr>
<tr>
<td>Creating the MainView Logger started task procedure on page 90</td>
<td>Required, except not applicable to MainView FOCAL POINT</td>
</tr>
<tr>
<td>Setting up a product-specific logspace on page 91</td>
<td>Optional, except not applicable to MainView FOCAL POINT or MainView Transaction Analyzer</td>
</tr>
<tr>
<td>Task</td>
<td>Required or optional?</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Historical data set determination on page 95</td>
<td>Optional for the following products, but not applicable to others:</td>
</tr>
<tr>
<td></td>
<td>- CMF MONITOR</td>
</tr>
<tr>
<td></td>
<td>- MainView for z/OS</td>
</tr>
<tr>
<td></td>
<td>- MainView for UNIX System Services (USS)</td>
</tr>
<tr>
<td></td>
<td>- MainView SYSPROG Services</td>
</tr>
<tr>
<td></td>
<td>- MainView VistaPoint</td>
</tr>
<tr>
<td></td>
<td>- MainView AutoOPERATOR</td>
</tr>
<tr>
<td></td>
<td>- MainView for CICS</td>
</tr>
<tr>
<td></td>
<td>- MainView for DB2</td>
</tr>
<tr>
<td></td>
<td>- MainView for DBCTL</td>
</tr>
<tr>
<td></td>
<td>- MainView for IMSOnline</td>
</tr>
<tr>
<td></td>
<td>- MainView for IMS Offline</td>
</tr>
<tr>
<td></td>
<td>- MainView for IP</td>
</tr>
<tr>
<td></td>
<td>- MainView for Linux - Servers</td>
</tr>
<tr>
<td></td>
<td>- MainView for VTAM</td>
</tr>
<tr>
<td></td>
<td>- MainView for WebSphere Application Server</td>
</tr>
<tr>
<td></td>
<td>- MainView for WebSphere MQ</td>
</tr>
</tbody>
</table>

Overview of manual customization tasks
<table>
<thead>
<tr>
<th>Task</th>
<th>Required or optional?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating a z/OS PAS PARMFILE data set on page 100</td>
<td>Required for the following products, but not applicable to others:</td>
</tr>
<tr>
<td></td>
<td>■ CMF MONITOR</td>
</tr>
<tr>
<td></td>
<td>■ MainView for z/OS</td>
</tr>
<tr>
<td></td>
<td>■ MainView for UNIX System Services (USS)</td>
</tr>
<tr>
<td></td>
<td>■ MainView SYSPROG Services</td>
</tr>
<tr>
<td></td>
<td>■ MainView VistaPoint</td>
</tr>
<tr>
<td>Creating the z/OS PAS startup procedure on page 101</td>
<td>Required for the following products, but not applicable to others:</td>
</tr>
<tr>
<td></td>
<td>■ CMF MONITOR</td>
</tr>
<tr>
<td></td>
<td>■ MainView for z/OS</td>
</tr>
<tr>
<td></td>
<td>■ MainView for UNIX System Services (USS)</td>
</tr>
<tr>
<td></td>
<td>■ MainView SYSPROG Services</td>
</tr>
<tr>
<td></td>
<td>■ MainView VistaPoint</td>
</tr>
<tr>
<td>Creating a BBI-SS PAS PARMFILE data set on page 103</td>
<td>Optional for the following products, but not applicable to others:</td>
</tr>
<tr>
<td></td>
<td>■ MainView AutoOPERATOR</td>
</tr>
<tr>
<td></td>
<td>■ MainView for CICS</td>
</tr>
<tr>
<td></td>
<td>■ MainView for DB2</td>
</tr>
<tr>
<td></td>
<td>■ MainView for DBCTL</td>
</tr>
<tr>
<td></td>
<td>■ MainView for IMSOnline</td>
</tr>
<tr>
<td></td>
<td>■ MainView for IMSOffline</td>
</tr>
<tr>
<td></td>
<td>■ MainView for WebSphere MQ</td>
</tr>
</tbody>
</table>
### Overview of manual customization tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>Required or optional?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defining BBI-SS PAS suffixes and target system parameters on page 105</td>
<td>Required for the following products, but not applicable to others:</td>
</tr>
<tr>
<td>■ MainView FOCAL POINT</td>
<td>● MainView AutoOPERATOR</td>
</tr>
<tr>
<td>■ MainView for CICS</td>
<td>■ MainView for DB2</td>
</tr>
<tr>
<td>■ MainView for DB2</td>
<td>■ MainView for DBCTL</td>
</tr>
<tr>
<td>■ MainView for DBCTL</td>
<td>■ MainView for IMSOnline</td>
</tr>
<tr>
<td>■ MainView for IMSOffline</td>
<td>■ MainView for IMSOffline</td>
</tr>
<tr>
<td>■ MainView for IMSOffline</td>
<td>■ MainView for WebSphere MQ</td>
</tr>
<tr>
<td>■ MainView for IMSOffline</td>
<td></td>
</tr>
<tr>
<td>■ MainView for IMSOffline</td>
<td></td>
</tr>
<tr>
<td>■ MainView for IMSOffline</td>
<td></td>
</tr>
<tr>
<td>■ MainView for IMSOffline</td>
<td></td>
</tr>
<tr>
<td>■ MainView for IMSOffline</td>
<td></td>
</tr>
<tr>
<td>■ MainView for IMSOffline</td>
<td></td>
</tr>
<tr>
<td>■ MainView for WebSphere MQ</td>
<td></td>
</tr>
</tbody>
</table>

| Defining configuration members to BBI-SS PAS on page 111            | Optional for the following products, but not applicable to others:                   |
| ■ MainView FOCAL POINT                                              | ● MainView AutoOPERATOR                                                               |
| ■ MainView for CICS                                                  | ■ MainView for DB2                                                                    |
| ■ MainView for DB2                                                   | ■ MainView for DBCTL                                                                   |
| ■ MainView for DBCTL                                                 | ■ MainView for IMSOnline                                                              |
| ■ MainView for IMSOffline                                           | ■ MainView for IMSOffline                                                              |
| ■ MainView for IMSOffline                                           | ■ MainView for IMSOffline                                                              |
| ■ MainView for IMSOffline                                           | ■ MainView for WebSphere MQ                                                            |

50 MainView Customization Reference
<table>
<thead>
<tr>
<th>Task</th>
<th>Required or optional?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Defining BBI-SS PAS interval services parameters on page 114</strong></td>
<td>Optional for the following products, but not applicable to others:</td>
</tr>
<tr>
<td>■ MainView FOCAL POINT</td>
<td>■ MainView AutoOPERATOR</td>
</tr>
<tr>
<td>■ MainView AutoOPERATOR</td>
<td>■ MainView for CICS</td>
</tr>
<tr>
<td>■ MainView for CICS</td>
<td>■ MainView for DB2</td>
</tr>
<tr>
<td>■ MainView for DB2</td>
<td>■ MainView for DBCTL</td>
</tr>
<tr>
<td>■ MainView for DBCTL</td>
<td>■ MainView for IMSOnline</td>
</tr>
<tr>
<td>■ MainView for IMSOnline</td>
<td>■ MainView for IMSOffline</td>
</tr>
<tr>
<td>■ MainView for IMSOffline</td>
<td>■ MainView for WebSphere MQ</td>
</tr>
<tr>
<td><strong>Setting the BBI-SS PAS historical recording interval on page 117</strong></td>
<td>Optional for the following products, but not applicable to others:</td>
</tr>
<tr>
<td>■ MainView AutoOPERATOR</td>
<td>■ MainView for CICS</td>
</tr>
<tr>
<td>■ MainView for CICS</td>
<td>■ MainView for DB2</td>
</tr>
<tr>
<td>■ MainView for DB2</td>
<td>■ MainView for DBCTL</td>
</tr>
<tr>
<td>■ MainView for DBCTL</td>
<td>■ MainView for IMSOnline</td>
</tr>
<tr>
<td>■ MainView for IMSOnline</td>
<td>■ MainView for IMSOffline</td>
</tr>
<tr>
<td>■ MainView for IMSOffline</td>
<td>■ MainView for WebSphere MQ</td>
</tr>
<tr>
<td>Task</td>
<td>Required or optional?</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Allocating BBI-SS PAS journal and image logs on page 118</td>
<td>Optional for the following products, but not applicable to others:</td>
</tr>
<tr>
<td>■ MainView FOCAL POINT</td>
<td></td>
</tr>
<tr>
<td>■ MainView AutoOPERATOR</td>
<td></td>
</tr>
<tr>
<td>■ MainView for CICS</td>
<td></td>
</tr>
<tr>
<td>■ MainView for DB2</td>
<td></td>
</tr>
<tr>
<td>■ MainView for DBCTL</td>
<td></td>
</tr>
<tr>
<td>■ MainView for IMSOnline</td>
<td></td>
</tr>
<tr>
<td>■ MainView for IMSOffline</td>
<td></td>
</tr>
<tr>
<td>■ MainView for WebSphere MQ</td>
<td></td>
</tr>
<tr>
<td>Allocating MainView Total Object Manager data sets on page 120</td>
<td>Optional for MainView AutoOPERATOR, but not applicable to other products</td>
</tr>
<tr>
<td>Allocating MainView AutoOPERATOR data sets on page 121</td>
<td>Required for MainView AutoOPERATOR, but not applicable to other products</td>
</tr>
<tr>
<td>Creating a BBI-SS PAS startup procedure on page 121</td>
<td>Required for the following products, but not applicable to others:</td>
</tr>
<tr>
<td>■ MainView FOCAL POINT</td>
<td></td>
</tr>
<tr>
<td>■ MainView AutoOPERATOR</td>
<td></td>
</tr>
<tr>
<td>■ MainView for CICS</td>
<td></td>
</tr>
<tr>
<td>■ MainView for DB2</td>
<td></td>
</tr>
<tr>
<td>■ MainView for DBCTL</td>
<td></td>
</tr>
<tr>
<td>■ MainView for IMSOnline</td>
<td></td>
</tr>
<tr>
<td>■ MainView for IMSOffline</td>
<td></td>
</tr>
<tr>
<td>■ MainView for WebSphere MQ</td>
<td></td>
</tr>
<tr>
<td>Adding the CICS EXCI library to the BBI-SS startup procedure on page 126</td>
<td>Required for the MainView AutoOPERATOR for CICS option and for MainView for CICS, but not applicable to other products</td>
</tr>
<tr>
<td>Task</td>
<td>Required or optional?</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------------------------------------------------------</td>
</tr>
<tr>
<td>Creating the MainView Total Object Manager start procedure on page 127</td>
<td>Optional for MainView AutoOPERATOR, but not applicable to other products</td>
</tr>
<tr>
<td>Creating automatic BBI-SS PAS log handling procedures on page 128</td>
<td>Required for the following products, but not applicable to others:</td>
</tr>
<tr>
<td></td>
<td>■ MainView FOCAL POINT</td>
</tr>
<tr>
<td></td>
<td>■ MainView AutoOPERATOR</td>
</tr>
<tr>
<td></td>
<td>■ MainView for CICS</td>
</tr>
<tr>
<td></td>
<td>■ MainView for DB2</td>
</tr>
<tr>
<td></td>
<td>■ MainView for DBCTL</td>
</tr>
<tr>
<td></td>
<td>■ MainView for IMS</td>
</tr>
<tr>
<td></td>
<td>■ MainView for WebSphere MQ</td>
</tr>
<tr>
<td></td>
<td>■ MainView Alarm Manager version 2.1</td>
</tr>
<tr>
<td>Defining terminal session parameters on page 131</td>
<td>Optional for the following products, but not applicable to others:</td>
</tr>
<tr>
<td></td>
<td>■ MainView FOCAL POINT</td>
</tr>
<tr>
<td></td>
<td>■ MainView AutoOPERATOR</td>
</tr>
<tr>
<td></td>
<td>■ MainView for CICS</td>
</tr>
<tr>
<td></td>
<td>■ MainView for DB2</td>
</tr>
<tr>
<td></td>
<td>■ MainView for DBCTL</td>
</tr>
<tr>
<td></td>
<td>■ MainView for IMS</td>
</tr>
<tr>
<td></td>
<td>■ MainView for WebSphere MQ</td>
</tr>
<tr>
<td>Define cross-system communication on page 136</td>
<td>Optional</td>
</tr>
</tbody>
</table>
## Overview of manual customization tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>Required or optional?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specifying product initialization within a BBI-SS PAS on page 164</td>
<td>Required for the following products, but not applicable to others:</td>
</tr>
<tr>
<td>■ MainView FOCAL POINT</td>
<td></td>
</tr>
<tr>
<td>■ MainView AutoOPERATOR</td>
<td></td>
</tr>
<tr>
<td>■ MainView for CICS</td>
<td></td>
</tr>
<tr>
<td>■ MainView for DB2</td>
<td></td>
</tr>
<tr>
<td>■ MainView for DBCTL</td>
<td></td>
</tr>
<tr>
<td>■ MainView for IMS</td>
<td></td>
</tr>
<tr>
<td>■ MainView for WebSphere MQ</td>
<td></td>
</tr>
<tr>
<td>Enabling Automatic Restart Manager (ARM) support on page 166</td>
<td>Optional for MainView AutoOPERATOR, but not applicable to other products</td>
</tr>
<tr>
<td>Adding the CAS ID to the BBI-SS PAS on page 166</td>
<td>Required for the following products, but not applicable to others:</td>
</tr>
<tr>
<td>■ MainView AutoOPERATOR</td>
<td></td>
</tr>
<tr>
<td>■ MainView for CICS</td>
<td></td>
</tr>
<tr>
<td>■ MainView for DB2</td>
<td></td>
</tr>
<tr>
<td>■ MainView for DBCTL</td>
<td></td>
</tr>
<tr>
<td>■ MainView for IMS Online</td>
<td></td>
</tr>
<tr>
<td>■ MainView for IMS Offline</td>
<td></td>
</tr>
<tr>
<td>■ MainView for WebSphere MQ</td>
<td></td>
</tr>
<tr>
<td>Task</td>
<td>Required or optional?</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------------------</td>
</tr>
</tbody>
</table>
| Adding extended function data sets to the BBI-SS PAS startup procedure on page 167 | Optional for the following products, but not applicable to others:  
  - MainView AutoOPERATOR  
  - MainView for CICS  
  - MainView for DB2  
  - MainView for DBCTL  
  - MainView for IMSOnline  
  - MainView for IMSOffline  
  - MainView for WebSphere MQ |
| Update command processor table on page 169                           | Optional              |
| Specifying product initialization within the z/OS PAS on page 171     | Optional for the following products, but not applicable to others:  
  - CMF MONITOR  
  - MainView for z/OS  
  - MainView for UNIX System Services (USS)  
  - MainView SYSPROG Services  
  - MainView VistaPoint |
<p>| Customize MainView Alternate Access, which includes creating TAS and LAS initialization parameters and startup procedures. See the MainView Alternate Access Implementation and User Guide. | Optional         |</p>
<table>
<thead>
<tr>
<th>Task</th>
<th>Required or optional?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set up one or more product-specific PASs for products that do not use a BBI-SS or z/OS PAS. See the product-specific customization guides.</td>
<td>Required for the following products, but not applicable to others:  ■ MainView for IP  ■ MainView for Linux® - Servers  ■ MainView for VTAM  ■ MainView for WebSphere Application Server  ■ MainView Storage Resource Manager (SRM)  ■ MainView Transaction Analyzer</td>
</tr>
</tbody>
</table>
| Implement AutoRule | Optional for MainView AutoOPERATOR, but not applicable to other products  
AutoRule is an optional event-management utility. You can use this utility to create and maintain MainView AutoOPERATOR Rule Sets, which are built by using supplied knowledge bases, input from system MPF lists, and other existing rules. See the MainView AutoOPERATOR Customization Guide. |
| Implement security | Optional  
| Customize your product | Optional  
See the product-specific customization guides. |

**Specifying jobcards and other operational defaults**

In customization tasks that involve data set allocations, you must specify the volume serial ID (VOLSER) and UNIT name.

In other tasks, you must supply the high-level qualifier (hlq) for the product libraries.

To prepare for these tasks, make a record of the VOLSER and UNIT name for the product libraries, along with the following information:
The SYSID value is used to build names for data sets that cannot be shared across multiple systems. In all cases where hlq is referenced, this value is the prefix value for your MainView product libraries and the SYSID value.

### Generating JCL to run the installation verification program

Use the following procedure to verify MainView product installation by running BBAIVP, a batch job that examines load modules. You can also run BBAIVP after applying maintenance to verify the applied maintenance.

**Tip**

This task is optional for all products.

Member BBAIVP contains sample JCL that you can use to run program BBAIVP. BBAIVP reports:

- Any modules that are marked as not executable
- Programs that should be marked as re-entrant but are not marked
- Modules that contain unexpected unresolved references
- The library and number of modules that BBAIVP checked

#### To verify product installation

1. In BBSAMP member BBAIVP, replace ??????? with your high-level qualifier.

   The REPORT statement (or a SYSPRINT statement) shown in Figure 1 on page 57 specifies the data set for the BBAIVP report. The SYSLIB statement specifies the data set for BBAIVP to check.

**Figure 1: BBAIVP JCL to verify product installation**

//IVP EXEC PGM=BBAIVP
//STEPLIB DD DISP=SHR,DSN=????????.BBLOAD

---

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prefix</td>
<td>High-level qualifier for product libraries</td>
</tr>
<tr>
<td>Unit-T</td>
<td>Generic UNIT name for temporary data sets</td>
</tr>
<tr>
<td>Unit-P</td>
<td>Generic UNIT name for permanent data sets</td>
</tr>
<tr>
<td>VOLSER</td>
<td>VOLSER for permanent data sets</td>
</tr>
<tr>
<td>SYSID</td>
<td>System image ID (not to exceed eight characters)</td>
</tr>
</tbody>
</table>
Implementing Katakana terminal support

Use the following procedure to enable MainView products to use a Katakana terminal.

**Tip**
This task is optional for all products.

In this procedure, you will use BBILIB member @@YZZ001 to convert the following data sets:

- BBILIB for installation JCL
- BBMLIB for messages
- BBTLIB for tables

@@YZZ001 is a utility program that translates lowercase characters to uppercase. This utility uses ISPF Table Services to convert the BBTLIB members, executes the Terminal Monitor Program (TMP) in batch mode, and invokes ISPF.

---

2 Submit BBAIVP for processing.

3 Check your job output for the following condition codes:

<table>
<thead>
<tr>
<th>Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>No errors were found.</td>
</tr>
<tr>
<td>08</td>
<td>One or more unresolved references were found (see the report for details).</td>
</tr>
<tr>
<td>12</td>
<td>One or more modules were found not marked as re-entrant but should be (see the report for details).</td>
</tr>
<tr>
<td>16</td>
<td>An operational error occurred. A message was written to the REPORT file or to SYSLOG.</td>
</tr>
</tbody>
</table>

**Note**
The product might operate properly even if errors are found; in that case, contact BMC Customer Support to determine whether a problem exists.
To implement Katakana terminal support

1 Convert the BBLIB, BBMLIB, and BBTLIB data sets to uppercase:

   a Allocate new `hlq.BBILIB`, `hlq.BBMLIB`, and `hlq.BBTLIB` data sets to store converted installation JCL, message members, and table members.

      This step preserves the original distribution members so that non-Katakana terminals can use them.

      **Note**
      The allocated output data sets must have the same block sizes as the corresponding input data sets.

   b In BBILIB member `@@YZZ001`, modify the following DD statements to specify input and output data set names for the BBILIB, BBMLIB, and BBTLIB data sets:

      | DD name       | Description                                           |
      |---------------|-------------------------------------------------------|
      | LCBBLIB       | Lowercase input data from BBILIB and BBMLIB           |
      | UCBBLIB       | Uppercase translated BBILIB and BBMLIB output data    |
      | LCBBTLIB      | Lowercase input tables from BBTLIB                    |
      | UCBBTLIB      | Uppercase translated BBTLIB output tables             |

   c Compress the data sets and submit the utility program for processing.

      **Note**
      If the UCBBLIB and UCBBTLIB DD statements point to the same data sets as the LCBBLIB and LCBBTLIB DD statements, respectively, the original members are overwritten.

      Members from the input data sets are translated to uppercase and written to the output data sets. The utility program performs the translations in two stages. At the completion of each stage, the following WTO message is issued:

      *Conversion successfully completed.*

   d Check the job output for condition code 0, which indicates successful job completion.

2 Specify `CAPS=Y` for the EXEC statement PARM parameter in the CAS started task procedure.
Apply passwords with the Product Authorization Utility

To activate products, you must apply product passwords by using the Password Authorization Utility.

**Tip**
This task is required for all MainView products except MainView FOCAL POINT and MainView SRM, to which it does not apply.

For an explanation of the Password Authorization Utility, in see the *Installation System User Guide*.

Creating site data sets

Use this procedure to create the necessary site data sets.

**Tip**
This task is required for all products.

The required site data sets are as follows:

<table>
<thead>
<tr>
<th>Data set</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UBBPARM</td>
<td>Site-modified parameter members</td>
</tr>
<tr>
<td>SBBVDEF</td>
<td>Site-customized view definitions <em>(not used for MainView FOCAL POINT)</em></td>
</tr>
<tr>
<td>SBBHELP</td>
<td>Site-customized Help text <em>(not used for MainView FOCAL POINT)</em></td>
</tr>
<tr>
<td>SBBPROF</td>
<td><em>(optional)</em> Site-customized common profile library shared by all site users</td>
</tr>
<tr>
<td>SBBSDEF</td>
<td>Site-customized screen definitions <em>(not used for MainView FOCAL POINT)</em></td>
</tr>
</tbody>
</table>
If maintenance needs to be applied to the site data sets, you are alerted when distributed versions change. PTFs that affect any members in these data sets contain the following statement. You can then save your changes before replacing the members.

```++HOLD SYSTEM REASON(ACTION) CLASS(lib_name)```

To create site data sets

1. Use BBILIB member @@YZZ011 to create the necessary site data sets.

   Follow the instructions at the top of @@YZZ011 to modify the JCL to your site requirements. The site data set allocations require the attributes that are shown in the following table:

<table>
<thead>
<tr>
<th>Data set</th>
<th>Record format (RECFM)</th>
<th>Logical record length (LRECL)</th>
<th>Block size (BLKSIZE)</th>
<th>Primary tracks</th>
<th>Secondary tracks</th>
<th>Directory blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>UBBPARM</td>
<td>FB</td>
<td>80</td>
<td>6160</td>
<td>60</td>
<td>15</td>
<td>80</td>
</tr>
<tr>
<td>SBBVDEF</td>
<td>FB</td>
<td>80</td>
<td>6160</td>
<td>30</td>
<td>15</td>
<td>30</td>
</tr>
<tr>
<td>SBBHELP</td>
<td>FB</td>
<td>80</td>
<td>6160</td>
<td>30</td>
<td>15</td>
<td>30</td>
</tr>
<tr>
<td>SBBSED</td>
<td>FB</td>
<td>80</td>
<td>6160</td>
<td>30</td>
<td>15</td>
<td>30</td>
</tr>
<tr>
<td>UBBsamp</td>
<td>FB</td>
<td>80</td>
<td>6160</td>
<td>30</td>
<td>15</td>
<td>80</td>
</tr>
<tr>
<td>SBBPROF</td>
<td>FB</td>
<td>80</td>
<td>6160</td>
<td>30</td>
<td>15</td>
<td>30</td>
</tr>
</tbody>
</table>

2. Submit the job.

3. Check the job output for a condition code of 0 for successful job completion.

   **Note**

   If you use another method to allocate these data sets, such as the ISPF Data Set Utility, do not specify the PACK option. MainView product site data sets cannot be packed when allocated.
Creating site security data set

Although you cannot customize default global security parameters, you can create customized security for MainView resources at your site. Use the following procedure to define a security data set to contain these customized security definitions for your site.

**Tip**
This task is required for all MainView products except MainView FOCAL POINT, to which it does not apply.

In this procedure, you will define a security data set with the BBSECURE DD statement in the CAS and PAS started task procedures.

**Note**
The default global settings (default global security parameters, resource class properties, and product-specific resource definitions) are distributed in the BBACTDEF data set.

For more information about setting up security and using the MainView Plex Manager security views to customize security at your site, see the *MainView Security Guide*.

**To create a site security (BBSECURE) data set**

1. Create a data set with the following DCB characteristics:
   - RECFM=FB
   - LRECL=80
   - BLKSIZE= any multiple of 80
   Use the same high-level qualifier for this data set that you used for your target libraries during the installation process.

   **Note**
   If you use the ISPF Data Set Utility, do not specify the PACK option. MainView product site data sets cannot be packed when allocated.

2. In the CAS started task procedure, specify the name of the data set that you created on the DSN parameter of the BBSECURE DD statement.

3. In each PAS started task procedure, specify the name of the data set that you created on the DSN parameter of the BBSECURE DD statement.
Adding the BMC load library to your system APF list

Use the following procedures to temporarily or permanently authorize the BMC load library.

Tip
This task is optional for all products.

To temporarily authorize the BBLINK library

1 Create UBBSAMP member BAIAPF.

2 Copy BBILIB member BAIAPF to the new member.

3 Change the PROC statement PREFIX parameter from @PREFIX to prefix, where prefix is the high-level qualifier for the MainView product libraries.

4 Change the PROC statement LIB parameter from @BBIAPF to the name of a previously authorized library.

5 Change the PROC statement VOL parameter from @BBIVOL to the VOLSER of the volume where MainView product libraries reside.

6 Submit the job.

To permanently authorize the BBLINK library

1 Use the Authorized Program Facility (APF) to update either SYS1.PARMLIB member IEAAPFxx or, for SP 5 and above, SYS1.PARMLIB member PROGxx:
### Library member | Variable description
--- | ---
IEAAPF\textit{xx} | \textit{xx} identifies the APF suffix specified in the IEASYS member that was used for the last system IPL. An IPL is required before the SYS1.PARMLIB change can take effect.

\textit{PROG}\textit{xx} | \textit{xx} identifies the member in SYS1.PARMLIB that contains the parameters that define the list of APF-authorized libraries. 

| Note: You can specify multiple \textit{PROG}\textit{xx} members with the PROG parameter as follows: \textbf{PROG=}( \textit{xx ..}, \textit{xx ..}) \textit{xx} can be any two alphanumeric characters. An example follows: \textbf{PROG=(11,12)} |

Libraries are placed in the APF list as follows:

- Libraries specified in the \textit{PROG}\textit{xx} members are processed.

- If you have an existing IAEAxFF\textit{xx} member and \textit{PROG}\textit{xx}, both are processed. To use only \textit{PROG} \textit{xx}, change IAEAxFF\textit{xx} to \textit{PROG}\textit{xx} and remove the APF parameters from IEASYS\textit{xx} and IEASYS00.

- If you have both \textit{PROG}\textit{xx} and EXIT\textit{xx}, \textit{PROG}\textit{xx} is processed first, then EXIT\textit{xx}. To use only \textit{PROG}\textit{xx}, change EXIT\textit{xx} to \textit{PROG}\textit{xx} and remove the EXIT parameters from IEASYS\textit{xx} and IEASYS00.

An IPL is required before the SYS1.PARMLIB change can take effect. Alternatively, you can authorize the new libraries with the MVS SET command as follows:

\textbf{T PROG=} \textit{xx} 

\textit{xx} identifies the SYS1.PARMLIB member.

For more information about APF authorization, see the IBM Initialization and Tuning Reference manual for your operating system.

---

### Adding the BMC load library to your system link list

Use the following procedure to add the BMC load library, \textit{h1q.BBLINK}, to your system link list. BBLINK contains authorized programs.

\textbf{Tip}

This task is optional for all products.
Before you begin

Keep the following considerations in mind:

- IBM requires that any data sets that are added to your system link list be cataloged in the master catalog.

- If you want to run COMMON STORAGE MONITOR (CSM) as a subsystem, you must perform this task.

- BBLINK should be placed after your SAS/C runtime libraries in the system link list.

- If BBLINK is added to the system link list when starting IMS, the AOI exits are dynamically invoked if MainView AutoOPERATOR is installed, regardless of whether MainView AutoOPERATOR for IMS is activated.

To add BBLINK to your system link list

1. Add the following line to your SYS1.PARMLIB(LNKLSTxx) member:
   
   \texttt{hlq.BBLINK},

   Use the same \texttt{hlq} (high-level qualifier) that you used for your target libraries during the installation process.

2. In the MainView startup procedures (such as the CAS and PAS started task procedures), remove the STEPLIB DD statement for \texttt{hlq.BBLINK}.

3. Update your system link list dynamically.

Creating a CLIST for invoking MainView products

Use the following procedure to create a CLIST to invoke MainView products under ISPF. This CLIST allocates all required libraries and connects to them through the ISPF LIBDEF.

\textbf{Tip}

This task is required for all products.

If you have only MainView FOCAL POINT installed, MainView Customization uses TSCLIST. To manually customize TSCLIST, you can use BBSAMP member TSCLIST.
You can use BBILIB member BAIMAINV as a model to help you create a MainView CLIST for accessing all MainView products.

**To create the MainView CLIST**

1. In `hlq.UBBPARM`, create a member and give it a name such as `MainView`.

   **Note**
   You can specify a name other than `MainView` for the CLIST. However, the remainder of this procedure uses `MainView`.

2. Copy the sample CLIST from BBILIB member BAIMAINV to the newly created `MainView` member in UBBPARM.

3. Edit the `MainView` CLIST as follows:

   a. In the `PREFIX('xxxxx')` parameter, replace `xxxxx` with the high-level qualifier for your target libraries.

   b. In the `PREFIX('bbbbb')` parameter, replace `bbbbb` with the `hlq.BBLINK` name for your target libraries.

   c. *(optional)* If you have installed any MainView products that run in the BBI-SS PAS, use the `SS( )` parameter to specify the subsystem ID for that PAS.

      If you want MainView to automatically discover the running BBI-SS PASs on a system and present a list to users during startup, specify `SS(????)`.

      This feature allows users to:

      - Select the PAS to which they want to connect at startup
      - Set a personal default connection for future sessions

      **Note**
      When a user selects a PAS, the target PAS must be at the same release and maintenance level as the code running in the MainView CLIST.

4. *(optional)* If you require support for GDDM high-resolution graphics, use the `ADMSYMBL` ddname to add a TSO ALLOCATE statement for the GDDM symbol library.

   If your level of GDDM supports printing, use the `ADMPRINT` ddname to add a TSO ALLOCATE statement for an optional GDDM PRINT data set. For more information about how to allocate the GDDM PRINT data set, refer to the GDDM member SYS1.GDDMSAM(ADMQFMT).
To invoke the CLIST to access MainView products, perform one of the following tasks:

- Issue a TSO command from any ISPF COMMAND line.

- Add a menu item to an ISPF menu to execute the MainView CLIST:
  
  In the )BODY section of the ISPF menu panel, add the following statement:
  
  `MV +MainView - Invoke MainView Products`

  In the )PROC section of the menu panel, add the following selection command for the MV option:

  `MV,'CMD(EX "hlq.UBBPARM(MainView)")'`

- Activate a MainView Alternate Access VTAM or EXCP session that displays the MainView Selection Menu.

  All MainView products include a component called MainView Alternate Access that provides EXCP and VTAM communication to the products through ISPF without requiring a TSO subsystem to be active. MainView Alternate Access fully supports:

  - EXCP or VTAM terminal sessions (including the MainView AutoOPERATOR CSM application and RxD2)
  - All supported releases of MainView products

  **Note**
  
  MainView Alternate Access requires ISPF. If ISPF is not available in your environment, MainView Alternate Access cannot be used.

BMC recommends that you use MainView Customization to customize MainView Alternate Access. The required tasks in MainView Customization show BBV as the product code.

If you decide not to use MainView Customization, see the *MainView Alternate Access Implementation and User Guide* for manual customization instructions.

  **Note**
  
  Changes made to the MainView CLIST support the MainView Alternate Access AUTOLOGON feature. Following MainView Customization of the MainView CLIST, review and replace any copies of TSCLIST that you have already made.
Adding TSF authorization for products to run under ISPF

To run your products in both windows of an ISPF split-screen session, you must add TSF authorization and add BBLINK to either STEPLIB or LINKLIST. Otherwise, split-screen access is available only when the products are invoked from native TSO.

**Tip**
This task is required for the following products, but does not apply to other products:

- MainView FOCAL POINT
- MainView AutoOPERATOR
- MainView for CICS
- MainView for DB2
- MainView for DBCTL
- MainView for IMS Online
- MainView for IMS Offline
- MainView for WebSphere MQ

TSF authorization is also required for the MainView TRAnsf er command that enables users to switch to product services that operate in the ISPF panel environment.

**To add TSF authorization**

1. Add **PMGLAUTH** to the AUTHTSF list in SYS1.PARMLIB(IKJTSOxx).

2. To ensure the addressability of the libraries, perform one of the following tasks:
   
   - Concatenate BBLINK to the STEPLIB in the ISPF logon procedures that are to be used for your BBI-based products.

   **Note**
   All libraries in the STEPLIB concatenation must be APF authorized.
   
   - Add BBLINK to your system link list.
Configure RTCS

The Runtime Component System (RTCS) runs as a started task that is started soon after the system IPL occurs. This address space provides programming services to all of the CASs, PASs, and UASs.

RTCS is designed for continuous operation and seldom if ever needs to be stopped.

Before using RTCS, you must complete the configuration tasks, which are documented in Runtime Component System Configuration and Administration Guide.

**Tip**
This task is required for all MainView products except MainView FOCAL POINT, to which it does not apply.

Creating the CAS startup procedure

Use the following procedure to set up your CAS started task procedure. You can use member @@YZZ022 as a model to help you.

**Tip**
This task is required for all MainView products except MainView FOCAL POINT, to which it does not apply.

**Before you begin**

When you create the CAS started task procedure, use the SSID parameter on the PROC statement to specify a subsystem ID for the CAS.

The CAS subsystem ID is used:

- To identify the CAS to the operating system
- In the startup procedure of each PAS that connects to the CAS
- In the Session Control Parameters dialog when a terminal session starts

**Note**
The CAS and PAS started task procedures and the Session Control parameters dialog must all specify the same subsystem ID.
To create the CAS started task procedure

1. Copy BBILIB member @@YZZ022 to SYS1.PROCLIB or another system procedure library and give it an appropriate name for a started task procedure.

   Note
   - By default, MainView Customization creates a member called BBMCAS.
   - BMC recommends changing the started task name to MVCAS to be consistent with other MainView started task names.

2. Follow the instructions at the top of the member to modify the procedure to your site requirements.

3. Define the CAS started task procedure to your IPL procedure so that the CAS starts at system initialization.

   Note
   The CAS must run as a started task.

4. Modify the COMMNDxx member in SYS1.PARMLIB to add a START command for the CAS procedure:

   COM='S procname, SUB=MSTR'

   The variable procrate is the name of the CAS started task procedure.

   Note
   If you plan to use TCP/IP for CAS-to-CAS communication, do not specify SUB=MSTR on the CAS START command. SUB=MSTR allows the CAS to start before OMVS, which can result in the CAS being unable to communicate via TCP/IP (even after OMVS and TCP/IP start).

Where to go from here

To support the MainView Infrastructure (MVI) common registry, the CAS started task procedure must have:

- A UNIX System Services security context, called an OMVS segment, defined for it
- Read and write access to the UNIX System Services file system that you allocate in Allocate MainView Infrastructure common registry on page 74
Allocating the CASPERM registry data set

Use the following procedure to allocate a data set for the CASPERM registry by using BBSAMP member BBMCASPR.

The CASPERM registry is a Runtime Component System (RTCS) product registry that stores MainView object definitions (such as dynamic threshold definitions). One CASPERM registry is required for each CAS in your MainView environment; a CASPERM registry cannot be shared by multiple CASs.
To allocate a CASPERM registry data set

1. Follow the instructions in BBSAMP member BBMCASPR to specify the required information, including:
   - Data set name for the CASPERM registry, which must be unique to the local CAS
   - Name and subsystem ID (SSID) of the local CAS
   - Space, in cylinders, for the registry data set (minimum of 185 cylinders)

2. Save the modified BBMCASPR member and submit the job.

Creating MainView host server startup procedure

Use the following procedure to create MainView host server startup procedure.

The MainView host server is a single address space that must be started to support all clients.

Following are examples of such clients:

- Discovery Agent (provides communications via the infrastructure APIs)
- MainView Explorer (gives you access to MainView views from a web browser)
  MainView Explorer is a client/server application that you can use to access MainView products from a web browser.

*Tip*
This task is required for all MainView products except MainView FOCAL POINT, to which it does not apply.

To create the host server startup procedure

You can use member BBMXPJCL as a model for the host server startup procedure.

1. Copy BBSAMP member BBMXPJCL to a procedure library.
   You can change the member name if you want. The member name is the name that you will specify on the START command when you start the host server.

2. Modify member BBMXPJCL as follows:
a Change SSID=?SSID to **SSID=** ssid, where ssid is the subsystem ID of the CAS to which you want to connect.

The default is BBCS.

b Change PORT=?PORT to **PORT=** nnnnn, where nnnnn is a unique port number for the host server.

You can use any integer between 1024 and 65535. The default is 3940.

---

**Note**

If you change the port number from the default of 3940, you must:

- Specify a unique port number for each application that runs on a given system image
- Specify the new port number when you start the client

c *(optional)* Use or remove the USERDS parameter to control the allocation of user data sets.

The BBVDEF and BBCDEF user data sets hold customized views and configurations, respectively. You can specify the USERDS parameter as follows:

- USERDS= template defines a template that can be made up of the TSO prefix (%UPFX), the TSO user ID (%USERID), and the data set name (%BBDEF). One or both of the user tokens can be specified, and they can appear in any order. %BBDEF, which resolves to BBVDEF or BBCDEF, is required.
  
The default template is %UPFX.%USERID.%BBDEF. The template that you specify appears on the MainView Explorer Logon dialog box. This template cannot be changed by the user.

- USERDS=NO disables the allocation of user data sets. The MainView Explorer Logon dialog box is truncated; the fields that are related to the allocation of user data sets do not appear.

If you remove the USERDS parameter from the host server startup procedure, users can define their own data set template when they log on to MainView Explorer.
The following are examples of valid templates and the user data sets that they would allocate for a user ID of MYID01:

- **%USERID.TEST.%BBDEF** allocates MYID01.TEST.BBVDEF (and BBCDEF).
- **PROD.U%BBDEF.%USERID** allocates PROD.UBBVDEF.MYID01 (and UBBCDEF).
- **%USERID.&SYSNAME..%BBDEF** allocates MYID01.SYSC.BBVDEF (on SYSC).

Including the system variable &SYSNAME., as shown in the last example, enables you to use the same data set template on different system images.

If you changed the member name to something other than BBMXPJCL, update the PROC statement to specify the new member name.

1. Save the member.

**To create BBTTCP00**

The BBTTCP00 member is optional. If you want to connect to a secondary TCP/IP stack or debug TCP/IP related problems, use member BBTTCP00.

1. Copy BBSAMP member BBTTCP xx to the UBBPARM data set.
2. Modify the member with the desired parameters.
3. Rename the modified member to BBTTCP00.

**Related Information**

- “BBTTCP00 parameters” on page 228

**Allocate MainView Infrastructure common registry**

The MainView Infrastructure (MVI) common registry requires a UNIX System Services (USS) HFS or zFS file system. The MVI common registry stores information for Alarm Management, deployment, and user preferences.
Tip
This task is required for all products.

Note
In previous releases, the file system for the MVI common registry was allocated as
the Alarm Management file system. If you already have a file system allocated for
use by Alarm Management, you do not need to allocate a new file system.

This task assumes that you are familiar with z/OS UNIX System Services (or have
access to someone who is). For information about UNIX System Services, see the
following IBM documentation:

- UNIX System Services z/OS Implementation Guide
- z/OS Distributed File Service zFS Administration
- z/OS UNIX System Services Planning

Before allocating the MainView Infrastructure common registry

Consider the following items before allocating the MainView Infrastructure (MVI)
common registry:

- Decide whether you want to use HFS or zFS for the MVI common registry file
  system.
  If you are not sure which type of file system to use, check with the system
  programmer who is responsible for UNIX System Services at your site. BMC
  recommends using zFS.

- Determine how many MVI common registry file systems you will need. Each CAS
  can have a MVI common registry, or CASs can share a single MVI common
  registry. For more information about the MVI common registry, see the MainView
  Administration Guide.

- BMC recommends that the file system be mounted off a user mount point or the
  IBM /var mount point. You must create a directory to use as a mount point for
  the MVI common registry file system. Do not choose an existing directory that is
  used for some other purpose.
WARNING

- Mounting a file system on an existing directory will make all existing data inaccessible.

- If the mount point is the system root, the MVI common registry will not initialize.

BMC has no dependency on the location of this directory. The best location will depend on your site’s configuration and use of the HFS or zFS file system.

- It is very important that the file system that you allocate (either HFS or zFS) is sized properly and configured to grow dynamically. Be sure to allocate an appropriate number of primary and secondary extents on a volume that has available free space.

When you allocate the initial file system, specify at least 75 MB of initial storage. Since the majority of the storage is used for alarm definitions, you can use the following formula to estimate the amount of space that is required for your alarm definitions:

\[
\text{Size (in bytes)} = \text{Number of Alarms} \times 1.5 \times 100,000
\]

- If you decide to use zFS, keep the following considerations in mind:
  
  — zFS must be defined on your z/OS system by using the IOEFSPRM PARMLIB member.
  
  — You should allocate the aggregate in compatibility mode.
  
  — To ensure that the file system can grow dynamically if it requires more space, use the AGGRGROW IOEFSPRM option or the AGGRGROW MOUNT parameter.

- The CAS and PAS started task procedures must have read and write access to the file system that you allocate.

Allocating the MVI common registry with BBSAMP members HFSALLOC and ZFSALLOC

You can use one of the following members as a model for allocating and mounting the MVI common registry file system:

- HFSALLOC to allocate and mount an HFS file system
- ZFSALLOC to allocate and mount a zFS file system
To allocate the MVI common registry file system

1. Copy the appropriate BBSAMP member (HFSALLOC or ZFSALLOC) to your UBBSAMP data set.

2. Modify HFSALLOC or ZFSALLOC as described in the member.

3. Save the modified member and submit the job.

4. Repeat these steps for each file system that you want to allocate.

Where to go from here

Before you can use the allocated file system, you must perform the following tasks:

- Mount the file system (HFS or zFS) on the operating system.
  You must add a MOUNT statement to the BPXPRMxx member in SYS1.PARMLIB. The MOUNT statement will mount the file system on subsequent IPLs.

- Identify a mount point for the file system to each CAS.
  You must use the Plex Manager HFSPATH view to identify the mount point. For more information, see the MainView Administration Guide.

Defining MainView Alarm Manager initialization parameters

Use the following procedure to enable MainView Alarm Manager for initialization within its own PAS.

Tip

- This task is optional for all MainView products except MainView FOCAL POINT, to which it does not apply.

- If you are using MainView Alarm Management version 5.0 or later, and you want to use Alert Management, this task is required. For information about MainView Alarm Management and Alert Management, see the MainView Alarm Management Guide.

MainView Alarm Manager cannot reside in the same BBI-SS PAS as any other MainView product.
The BBISSPxx and BBIJNT00 parameter members, which are used in this task, support the substitution of system variables.

MainView Alarm Manager requires its own BBISSPxx member. This member cannot be shared by any other BBI-SS PAS.

Other considerations are as follows:

- The BBISSPxx member for MainView Alarm Manager resides in a separate parameter library with a low-level qualifier of MVAPARM.
- You should not include the UBBPARM data set in the BBPARM concatenation of the MainView Alarm Manager PAS JCL.

**To update BBISSPxx for MainView Alarm Manager**

1. Specify the following PRODUCT parameter:

   ```
   PRODUCT=MVA
   ```

   **Note**
   
   PRODUCT=MVA must be the only PRODUCT parameter in the BBISSPxx member for MainView Alarm Manager. If other PRODUCT parameters are found, MainView Alarm Manager will not initialize.

2. Specify the ALRTRCVE parameter as follows:

   ```
   ALRTRCVE=YES
   ```

   YES indicates that this MainView Alarm Manager PAS is eligible to receive alerts from other BMC products. The default is NO.

3. Specify the CASID parameter as follows:

   ```
   CASID=ssid
   ```

   `ssid` is the four-character subsystem ID of the CAS to which the MainView Alarm Manager PAS will connect.

4. *(optional)* Specify the AUDIT parameter as follows:

   ```
   AUDIT=YES|NO
   ```

   This parameter defines whether a request to delete an alert on an Alert Management view should be entered as an audit record in the BBI-SS journal.

   The default value is NO.
To specify BBPARM member BBIJNT00

1 Specify the TARGET parameter as follows:

```
TARGET=subsys,TYPE=MVS,SUBSYS=subsys
```

`subsys` is the subsystem that is specified in the MVALSS parameter of the MainView Alarm Manager PAS started task procedure.

To specify BBPARM member MVAPRM00

1 Specify values for the ALRMSFX and MVALARM parameters:

- Specify the ALRMSFX parameter as follows:
  ```
  ALRMSFX=xx
  ```
  `xx` is the suffix of the MainView Alarm Manager member to use.

  **Tip**
  To identify more than one member, separate each suffix with a comma and enclose the list in parentheses:
  ```
  ALRMSFX=00,xx(00,xx,..)
  ```

- Specify the MVALARM parameter as follows:
  ```
  MVALARM=value
  ```

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| NO      | Disables MainView Alarm Manager processing in the version 2.1 or earlier PAS, but continues to run the Alert Management feature. Use this value if you are running MainView Alarm Management 5.0 or later and want to use Alert Management. The following messages are issued when the version 2.1 PAS is started:

  - BBHEM0102I MVALARM=NO in BBIPARM member MVAPRM00 is detected
  - BBHEM0103I MVALARM product is disabled
  - BBHEM0104I MVALENT product initialization is being processed
  - BBHEM033I MainView Alert Manager Startup Complete - V2.1.02(BPY7458) |
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>Enables MainView Alarm Manager processing in the version 2.1 or earlier PAS, and runs the Alert Management feature</td>
</tr>
</tbody>
</table>

**Related Information**

- “System variables substitution in parameter library members” on page 233

---

**Allocating MainView Alarm Manager journal logs**

Use the following procedure to allocate MainView Alarm Manager journal logs.

MainView Alarm Manager uses journal log data sets to record:

- Product alerts, warnings, and messages
- Commands and responses that the Alert Management component issues

**Tip**

- This task is optional for all MainView products except MainView FOCAL POINT, to which it does not apply.
- If you are using MainView Alarm Management version 5.0 or later, and you want to use Alert Management, this task is required. For information about MainView Alarm Management and Alert Management, see the *MainView Alarm Management Guide*.

BBSAMP member SSLOG contains sample JCL to allocate MainView Alarm Manager journal log data sets.
Note

- Member SSLOG contains sample JCL to allocate MainView Alarm Manager journal log data sets. A set of journal log data sets must be allocated for every MainView Alarm Manager PAS.

- Do not alter the log attributes; the default attributes are reset automatically if you change them.

- If you are using products on multiple systems and want to use the same JCL procedure for all of them, use the MainView Alarm Manager SSID in the data set name of the logs. All of the subsystems can then use the same PROC.

To allocate the MainView Alarm Manager journal log data sets

1. In BBSAMP member SSLOG, make the following changes:
   a. Change the PREFIX parameter to the prefix that was used to load the product libraries to DASD.
   b. Set the SS parameter value to the subsystem ID of the MainView Alarm Manager PAS.
   c. Set the DV parameter value to the volume serial number of the allocation volume for the journal data sets.
   d. Set the DU parameter value to the allocation volume unit type.

2. Submit the modified job.

Allocating MainView Alarm Manager DIV data set

Use the following procedure to allocate MainView Alarm Manager DIV data set.

The Alert Management component of MainView Alarm Manager uses a data-in-virtual (DIV) data set to store binary large objects and data (such as non-volatile alerts) across restarts.
Tip

- This task is optional for all MainView products except MainView FOCAL POINT, to which it does not apply.
- If you are using MainView Alarm Management version 5.0 or later, and you want to use Alert Management, this task is required. For information about MainView Alarm Management and Alert Management, see the MainView Alarm Management Guide.

Before you begin

Note

- Each MainView Alarm Manager PAS must have its own copy of the DIV data set; the data set cannot be shared.
- The MainView Alarm Manager PAS must be shut down when this data set is allocated.

To allocate the MainView Alarm Manager DIV data set

1. In BBSAMP member DIVDEF, make the following changes to the procedure header:
   a. Change *HILEVEL* to the high-level qualifier for MainView product data sets.
   b. Change *SSID* to the four-character subsystem ID of the MainView Alarm Manager PAS.
   c. Change *VOL* to the volume serial number of the DASD volume where the linear data set will be allocated.

2. Submit the modified job.

Creating MainView Alarm Manager startup procedure

Use the following procedure to set up a started task procedure for the MainView Alarm Manager PAS.
This task is optional for all MainView products except MainView FOCAL POINT, to which it does not apply.

If you are using MainView Alarm Management version 5.0 or later, and you want to use Alert Management, this task is required. For information about MainView Alarm Management and Alert Management, see the MainView Alarm Management Guide.

BMC recommends that you run only one MainView Alarm Manager PAS per system image.

The recommended dispatching priority of the MainView Alarm Manager PAS is 255.

The Alarm Manager PAS must have access to an OMVS segment with read/write access to the MainView Infrastructure home directory. The home directory is specified by using the HFSPATH view in PLEXMGR. For more information, see the MainView Administration Guide.

To create BBHPAS

1. Copy BBSAMP member BBHPAS to SYS1.PROCLIB or another system procedure library.

2. Follow the instructions at the top of the member to modify the procedure for your site requirements.

3. (optional) Define the MainView Alarm Manager started task procedure to your IPL procedure so that the PAS starts at system initialization.

   BMC recommends that you modify the COMMNDxx member in SYS1.PARMLIB to add a START command for the MainView Alarm Manager PAS:

   COM='Sprocname'

   procname is the name of the MainView Alarm Manager PAS started task procedure.
**Note**
The START commands are issued in the order that they appear in COMMNDxx. The MainView Alarm Manager PAS START command must occur *after* the CAS START command. For more information about defining the CAS procedure, see Creating the CAS startup procedure on page 69.

**Where to go from here**

MainView Alarm Manager uses the AOAnywhere facility to display product alerts. AOAnywhere, which is packaged with all MainView products, enables certain automation functions to be shared among products. To use AOAnywhere, associate the MainView Alarm Manager PAS with a user ID that has READ access to the following resources:

- `prefix.ssid.BBI.target.ACCESS`
- `prefix.ssid.AAO.target.AOEXEC.ALRTREAD`
- `prefix.ssid.AAO.target.AOEXEC.ALRTUPD`
- `prefix.ssid.AAO.target.EXEC`

For information about securing these resources, see the discussion of resources used by multiple products in the *MainView Security Reference Manual*.

In addition, if you run MainView AutoOPERATOR and you plan to launch EXECs from alarm definitions, grant authority to MainView Alarm Manager to schedule those EXECs. For information about granting access to specific EXECs, see the discussion of MainView AutoOPERATOR EXECs in the *MainView Security Reference Manual*.

**Related Information**

- "MainView Alarm Manager version 2.1 PAS library member" on page 249

---

**Migrating BBMTYB00 (CAS definitions)**

You can use this procedure to migrate your CAS definitions if a new version of MainView Infrastructure is installed separately from any other MainView product. You can also use this procedure when migrating from an earlier version of MainView Infrastructure.
Tip
This task is optional for all MainView products except MainView FOCAL POINT, to which it does not apply.

CAS definitions are stored in member BBMTYB00 of the hlq.UBBPARM data set.

To migrate your existing CAS definitions

1 Copy member BBMTYB00 from your existing hlq.UBBPARM data set to the new hlq.UBBPARM data set that was created for MainView Infrastructure.

The new CAS that you are setting up will now know about all of your existing CASs.

Verification of MainView Logger data sets

MainView Logger is a stand-alone logging facility that provides any BMC product or MainView Infrastructure a way to log important messages or events in a centralized repository.

Tip
This task is required for all MainView products except MainView FOCAL POINT, to which it does not apply.

MainView Logger runs as a started task and supports multiple tasks, called logspaces, within the same address space.

One logspace is identified as the primary logspace. The primary logspace is shared by MainView Infrastructure and all MainView products. Additional product-specific logspaces might be required by MainView products that have special data requirements. In these instances, after installing the primary logspace, you install a product-specific logspace for each product that requires it.

For more information about MainView Logger, including how to manage the retention of log file data and the associated register entries, see the MainView Administration Guide.

See related tasks for procedures to allocate the data sets that are required for the MainView Logger primary logspace.
Note

- Each logspace requires its own register and data-in-virtual (DIV) data sets. These data sets cannot be shared between logspaces.

- MainView Logger must not be running when the data sets for the primary logspace are allocated.

- If you are customizing MainView Transaction Analyzer, you must set up a product-specific logspace.

- If you are customizing MainView for WebSphere MQ, you might need to set up product-specific logspaces.

Related Information

- “Setting up a product-specific logspace” on page 91

Allocating the register data set BBSAMP member LOGRREG

Use member LOGRREG to allocate a register data set for MainView Logger.

This data set stores log file names and statistics, and the log maintenance control settings.

To allocate the register data set

1. Copy BBSAMP member LOGRREG to your UBBSAMP data set.

2. Change *REGISTER* to the register data set name.

   You will specify this data set name on the REGISTERDSN parameter of the LOGGERxx PARMLIB member.
MainView Logger uses EXCP processing to provide low overhead processing of log data. For this reason, none of the log files or the register can be defined as Extended data sets.

BMC recommends that the MainView Logger register data set not be under the control of the Storage Management System (SMS). If the register data set must be under SMS control, be sure to specify storage class and management class values that allow the data set to be kept indefinitely. The register data set must not be deleted.

3 Change *UNIT*, *VOL*, *STORCLAS*, and *MGMTCLAS* to meet your site requirements.

4 For *SIZE*, specify an initial allocation of \( \text{SPACE}=(\text{CYL},(0,1)) \) for the register data set.

In this initial allocation, the zero primary extent tells MainView Logger to initialize one track’s worth of records. After that amount has been used up, the logger will initialize an equal amount of space to store more records. Each initialization group of records (except the first group) can hold 384 data set names on a 3390 device.

As a rule, you should expect to create from 1 to 3 log files in a single day for an instance of MainView Logger. At that rate, one track could hold from 4 to 12 months of log file names.

5 Save the member and submit the job.

**Related Information**

- “Creating MainView Logger LOGGERxx PARMLIB member” on page 88

---

**Allocating the DIV data set BBSAMP member LOGRDIV**

Use member LOGRDIV to allocate a data-in-virtual (DIV) data set for MainView Logger.

This data set is used to buffer log requests and writing log records to DASD.

**To allocate the DIV data set**

1. Copy BBSAMP member LOGRDIV to your UBBSAMP data set.
2 Change *DIV* to the DIV data set name.

You will specify this data set name on the DIVDSN parameter of the LOGGERxx PARMLIB member.

3 Change *SYSOUT*, *VOL*, and *STORCLAS* to meet your site requirements.

4 Change *SIZE* to the number of cylinders that you want to allocate for the DIV data set.

An allocation of one cylinder is recommended.

5 Save the member and submit the job.

---

### Related Information

- “Creating MainView Logger LOGGERxx PARMLIB member” on page 88

---

## Creating MainView Logger LOGGERxx PARMLIB member

Use the following procedure to define the initialization parameters for MainView Logger.

### Tip

This task is required for all MainView products except MainView FOCAL POINT, to which it does not apply.

### Note

Each MainView Logger logspace requires its own LOGGERxx PARMLIB member.

You will use member LOGGER00 to create initialization parameters for the primary logspace. As you modify the PARMLIB member, keep the following considerations in mind:

- By default, MainView Logger reads member LOGGER00 from the BBIPARM DD statement. However, if PARM=‘SYSP=x | ( .x.x,xx,...)’ is specified in the started task PROC statement, one or more LOGGERxx members can be read.

- MainView Logger performs syntax checking, starting with the group of parameters that follows MVSSYSTEM=\( sysname \) (where \( sysname \) equals the current system name in CVTSNAME). Syntax checking stops with the next MVSSYSTEM parameter or the end of the member.
- LOGGER00 supports the substitution of system variables.

**To create a LOGGERxx PARMLIB member**

1. Copy BBSAMP member LOGGER00 to your UBBPARM data set.

   This data set is the one that you will identify on the BBIPARM DD statement in the MainView Logger started task procedure.

2. Specify the following required parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVSSYSTEM=sysName</td>
<td>Identifies the system to which the parameters in this member apply. The sysname variable must match the current system name in CVTSNAME. If you specify the system variable &amp;SYSNAME, a single set of parameter statements can be used to start MainView Logger on multiple system images.</td>
</tr>
<tr>
<td>LOGCYLINDERS=nn</td>
<td>Specifies the size of the allocated log file data sets. Use either LOGCYLINDERS or LOGTRACKS, but not both. Valid values per logspace are 5 to 50 cylinders or 75 to 750 tracks.</td>
</tr>
<tr>
<td>LOGTRACKS=nnn</td>
<td></td>
</tr>
<tr>
<td>LOGPREFIX=dsnPrefix</td>
<td>Identifies the data set prefix for the allocated log data sets. MainView Logger supplies the low-level qualifier for the log file data sets in the form of Ayydddnn, where yy is the year, ddd is the Julian day, and nn is a sequence number.</td>
</tr>
<tr>
<td>REGISTERDSN=dsName</td>
<td>Identifies the name of the register data set that you allocated in <strong>Verification of MainView Logger data sets on page 85</strong>. The register data set name must be unique for each logspace.</td>
</tr>
<tr>
<td>DIVDSN=dsname</td>
<td>Identifies the name of the DIV data set that you allocated in <strong>Verification of MainView Logger data sets on page 85</strong>. The DIV data set name must be unique for each logspace.</td>
</tr>
</tbody>
</table>
Creating the MainView Logger started task procedure

Use the following procedure to set up a started task procedure for MainView Logger.

**Tip**
This task is required for all MainView products except MainView FOCAL POINT, to which it does not apply.

You use BBSAMP member MVLOGGER as a model to help you set up the started task procedure.

**To create the started task procedure**

1. Copy BBSAMP member MVLOGGER to SYS1.PROCLIB or another system procedure library.

2. Follow the instructions at the top of the member to modify the procedure to your site requirements.

3. *(optional)* Define the MainView Logger started task procedure to your IPL procedure so that the logger starts at system initialization.

---

**Related Information**

- "Verification of MainView Logger data sets" on page 85
- "Creating the MainView Logger started task procedure" on page 90
- "Setting up a product-specific logspace" on page 91
- "MainView Logger library members" on page 237
- "System variables substitution in parameter library members" on page 233
Setting up a product-specific logspace

Some MainView products require or can use a product-specific logspace.

**Tip**
This task is optional for all MainView products except MainView FOCAL POINT and MainView Transaction Analyzer version 3.2 or later, to which it does not apply.

If you are customizing a product that uses a product-specific logspace, perform this task after configuring the primary logspace.

**Note**
If you are using MainView for WebSphere MQ and you want to offload trace data, you should allocate one logspace per queue manager.

Each logspace requires its own:

- Register data set
- Data-in-virtual (DIV) data set
- LOGGERxx PARMLIB member

Additionally, a started task procedure is required on each system image where logspaces are used.

**Related Information**
- “Verification of MainView Logger data sets” on page 85

Allocating the register data set BBSAMP member LSALCREG

Use the following procedure (which uses member LSALCREG) to allocate a register data set for a product-specific logspace.

**To allocate the register data set**

1. Copy BBSAMP member LSALCREG to your UBBSAMP data set.

2. Change *REGISTER* to the register data set name.
You will specify this data set name on the REGISTERDSN parameter of the LOGGERxx PARMLIB member.

--- Note ---

BMC recommends that the MainView Logger register data set not be under the control of the Storage Management System (SMS). If the register data set must be under SMS control, be sure to specify storage class and management class values that allow the data set to be kept indefinitely. The register data set must not be deleted.

3 Change *UNIT*, *VOL*, and *STORCLAS* to meet your site requirements.

4 Save the member and submit the job.

**Allocating the DIV data set BBSAMP member LSALCDIV**

Use the following procedure (which uses member LSALCDIV) to allocate a data-in-virtual (DIV) data set for a product-specific logspace.

**To allocate the DIV data set**

1 Copy BBSAMP member LSALCDIV to your UBBSAMP data set.

2 Change *DIV* to the DIV data set name.

You will specify this data set name on the DIVDSN parameter of the LOGGERxx PARMLIB member.

3 Change *SYSOUT*, *VOL*, and *STORCLAS* to meet your site requirements.

4 Change *SIZE* to the number of cylinders that you want to allocate for the DIV data set.

An allocation of 50 cylinders is recommended for a product-specific logspace.

5 Save the member and submit the job.

**Creating LOGGERxx PARMLIB member with BBPARM member LSLOGGER**

Use the following procedure (which uses member LSLOGGER) to create initialization parameters for a product-specific logspace.
To create a LOGGERxx PARMLIB member

1 Copy BBSAMP member LSLOGGER to your UBBPARM data set.

This data set is the one that you will identify on the BBIPARM DD statement in the MainView Logger started task procedure.

2 Rename member LSLOGGER to LOGGERxx, where xx is a two-character suffix that is unique to this logspace.

During initialization, MainView Logger reads member LOGGERxx from the BBIPARM DD statement. The parameter member to be read is specified by using the SYSP=xx parameter on the started task PROC statement, where xx is the member suffix. A single started task procedure can be used to start multiple logspaces by specifying SYSP=(xx, xx, xx,...).

3 In the LOGGERxx member, the following parameters require user-defined values.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVSSYSTEM=sysName</td>
<td>Identifies the system to which the parameters in this member apply</td>
</tr>
<tr>
<td></td>
<td>The sysname variable must match the current system name in CVTSNAME.</td>
</tr>
<tr>
<td>LOGPREFIX=dsnPrefix</td>
<td>Identifies the data set prefix for the allocated log data sets</td>
</tr>
<tr>
<td></td>
<td>MainView Logger supplies the low-level qualifier for the log file data sets in the form of Ayydddn, where yy is the year, ddd is the Julian day, and nn is a sequence number.</td>
</tr>
<tr>
<td>REGISTERDSN=dsName</td>
<td>Identifies the name of the register data set that you allocated in Verification of MainView Logger data sets on page 85</td>
</tr>
<tr>
<td></td>
<td>The register data set name must be unique for each logspace.</td>
</tr>
<tr>
<td>DIVDSN=DIVDSN=dsname</td>
<td>Identifies the name of the DIV data set that you allocated in Verification of MainView Logger data sets on page 85</td>
</tr>
<tr>
<td></td>
<td>The DIV data set name must be unique for each logspace.</td>
</tr>
<tr>
<td>LOGSSID=ssid</td>
<td>Specifies the one- to four-character subsystem ID that is to be used by the started task procedure</td>
</tr>
<tr>
<td>LOGSPACE=xxxxxxxx</td>
<td>Specifies a three- to eight-character name for the logspace to which this LOGGERxx member applies</td>
</tr>
</tbody>
</table>
The following parameters are set to the recommended value for a product-specific logspace:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGCYLINDERS=nn</td>
<td>Specifies the size of the allocated log file data sets. The maximum value of 150 cylinders for a product-specific logspace is recommended for optimum logspace performance. Valid values per logspace are 5 to 150 cylinders.</td>
</tr>
<tr>
<td>DIVBLOCKS=nnnn</td>
<td>Specifies the number of blocks to be used by the DIV data set. For each cylinder that was allocated for the DIV data set, specify 180 blocks (one cylinder on a 3390 equals 180 blocks). For a product-specific logspace, where the recommended DIV data set allocation is 50 cylinders, specify the maximum value of 9000.</td>
</tr>
</tbody>
</table>

Related Information

- “MainView Logger library members” on page 237
- “Verification of MainView Logger data sets” on page 85

Creating a started task procedure for BBSAMP member LSLOGSTC

Use the following procedure to create a started task procedure for BBSAMP member LSLOGSTC.

A started task procedure is required on each system image where logspaces are used.

You can:

- Use the existing MainView Logger started task procedure
- Create a separate procedure for this product-specific logspace

If you decide to create another started task procedure, you can use BBSAMP member LSLOGSTC as a model.
To create a started task procedure

1. Copy BBSAMP member LSLOGSTC to SYS1.PROCLIB or another system procedure library.

2. Follow the instructions at the top of the member to modify the procedure to your site requirements.

3. (optional) Define the MainView Logger started task procedure to your IPL procedure so that the logger starts at system initialization.

Related Information

- “MainView Logger library members” on page 237

---

Historical data set determination

To help you decide the number of historical data sets you will need, the following table lists the type of PAS in which a product runs:

<table>
<thead>
<tr>
<th>PAS type</th>
<th>Products supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS PAS</td>
<td>■ CMF MONITOR</td>
</tr>
<tr>
<td></td>
<td>■ MainView for UNIX System Services</td>
</tr>
<tr>
<td></td>
<td>■ MainView for z/OS</td>
</tr>
<tr>
<td></td>
<td>■ MainView SYSPROG Services</td>
</tr>
<tr>
<td></td>
<td>■ MainView VistaPoint (for IBM MVS workloads)</td>
</tr>
</tbody>
</table>
### PAS type

<table>
<thead>
<tr>
<th>PAS type</th>
<th>Products supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBI-SS PAS</td>
<td>- MainView AutoOPERATOR</td>
</tr>
<tr>
<td></td>
<td>- MainView for CICS</td>
</tr>
<tr>
<td></td>
<td>- MainView for DB2</td>
</tr>
<tr>
<td></td>
<td>- MainView for DBCTL</td>
</tr>
<tr>
<td></td>
<td>- MainView for IMSOnline</td>
</tr>
<tr>
<td></td>
<td>- MainView for WebSphere MQ</td>
</tr>
<tr>
<td></td>
<td>- MainView VistaPoint (for CICS, DB2, DBCTL, and IMS workloads)</td>
</tr>
<tr>
<td>Product-specific PAS</td>
<td>- MainView for IP</td>
</tr>
<tr>
<td></td>
<td>- MainView for Linux - Servers</td>
</tr>
<tr>
<td></td>
<td>- MainView for VTAM</td>
</tr>
<tr>
<td></td>
<td>- MainView for WebSphere Application Server</td>
</tr>
<tr>
<td></td>
<td>- MainView Storage Resource Manager (SRM)</td>
</tr>
<tr>
<td></td>
<td>- MainView Transaction Analyzer</td>
</tr>
</tbody>
</table>

### Before creating historical data sets

Consider the following information before creating historical data sets:

- Each PAS requires its own set of historical data sets if any product in the PAS intends to keep historical data.

- If you are running MainView for z/OS, and you want to implement long-term history recording, you need to define a second set of historical data sets. For long-term history files, the low-level qualifier begins with HST1DSnn (rather than HISTDSnn). After the long-term files are defined, see the MainView for z/OS Customization Guide for information about activating long-term data collection.

- If you are running version 3.0.00 (or later) of MainView for z/OS or version 5.8.00 (or later) of CMF MONITOR, and you want to implement short-term history
recording, you need to define another set of historical data sets. For short-term history files, the low-level qualifier begins with HST2DSnn (rather than HISTDSnn). After the short-term files are defined, see the MainView for z/OS Customization Guide for information about activating short-term data collection.

- If you do not want to use historical data for products that run in the z/OS PAS, delete the HISTDSnn and HST1DSnn DD statements from the PAS started task procedure that is described in Creating the CAS startup procedure on page 69.

- If you are running version 11.1.00 (or later) of MainView for DB2, you need to define a set of historical data sets to contain one-minute statistics records. The low-level qualifier begins with HST1DSnn (rather than HISTDSnn).

  **Note**
  
  If these data sets are not allocated, the MainView for DB2 health analysis views and MainView Explorer configurations are disabled.

- You can also add historical data sets dynamically from the DSLIST view, as described in the MainView Administration Guide.

- If you have already defined the appropriate historical data sets, you do not need to perform this task.

**Related Information**

- "Creating the z/OS PAS startup procedure" on page 101

### Allocating historical data set BBILIB member @@YZZ051

Use the following procedure (which uses BBILIB member @@YZZ051 as a model) to allocate and format historical data sets.
Tip
This task is optional for the following products and does not apply to other products:

- CMF MONITOR
- MainView AutoOPERATOR
- MainView for CICS
- MainView for DB2
- MainView for DBCTL
- MainView for IMS Online
- MainView for IP
- MainView for Linux - Servers
- MainView for UNIX System Services (USS)
- MainView for VTAM
- MainView for WebSphere Application Server
- MainView for WebSphere MQ
- MainView for z/OS
- MainView Storage Resource Manager (SRM)
- MainView SYSPROG Services
- MainView Transaction Analyzer
- MainView VistaPoint

Historical data sets are VSAM files that store data at the end of a recording interval. You can display historical data in product views only if historical data sets are allocated.

To allocate and format the historical data sets

1. Determine how many VSAM files you need to define and how large the files should be.
You can allocate a maximum of 100 data sets. Default customization creates three 30-cylinder data sets (on a 3390). When one data set is full, recording starts with the next data set until all data sets are full. Recording then begins again with the first allocated data set and overwrites any existing data.

The appropriate number of historical data sets for your site depends on several factors, including system load, system configuration, and the specific MainView products that are sharing the z/OS PAS or the BBI-SS PAS. BMC suggests that you follow this procedure to determine the historical data set space requirements for your site:

a Decide how many historical data sets you want to use.

There is no rule about how many historical data sets a site should have. At some sites, the number is based on the number of times that the system workload characteristics change during the day: typically three times, corresponding to shift changes. However, you might decide that a different number is appropriate for your site.

b Determine the size of the historical data sets.

After you complete customizing all products, allow the PAS to run for approximately 24 hours. During the 24-hour period, note when the PAS receives messages indicating that the historical data set is full and is being switched. By using this method, you can determine how much space is typically required to hold historical data over a given amount of time. You can then adjust the size of the historical data sets according to whether you want them to hold data for a longer or shorter period of time.

(Note) The sample VSAM definitions do not contain space allocations for the DATA and INDEX components. The definitions rely on VSAM to divide the amount of space in the cluster between the DATA and INDEX components. If you add specific space allocations, do not assign too much space to the INDEX component (or use a primary and secondary allocation). An incorrect space allocation for the INDEX component could cause loss of data.

2 Copy BBILIB member @@YZZ051 to a UBBSAMP data set.

@@YZZ051 contains sample JCL to allocate three VSAM data sets with IDCAMS DEFINE CLUSTER statements.

3 Follow the instructions at the top of BBILIB member @@YZZ051 to modify the JCL according to your site's needs.

4 Submit the @@YZZ051 job.
The historical data sets are allocated and formatted by using an initial record format that is shipped in BBILIB member @@YZZ052.

The allocated data sets use a default data set name of hlq.smfid.HISTDSnn (for regular history), hlq.smfid.HST1DSnn (for long-term history or one-minute statistics for MainView for DB2 11.1.00 or later), or hlq.smfid.HST2DSnn (for short-term history).

The variables are as follows:

- **hlq** is the high-level qualifier for your BMC product libraries.
- **smfid** is the SMF identifier of the system for which the data will be collected.
- **nn** is a unique value from 00 to 99. Each historical data set must have a unique name.

5 Check the job output for condition code 0.

If the condition code is not 0, review the job output messages, correct the problems, and submit the job again.

6 Ensure that a //HISTDSnn, //HST1DSnn, or //HST2DSnn DD statement is defined in the PAS started task procedure for each historical data set that you allocate

---

**Related Information**

- “Creating the z/OS PAS startup procedure” on page 101
- “Creating a BBI-SS PAS startup procedure” on page 121

---

**Creating a z/OS PAS PARMFILE data set**

Use the following procedure (which uses member @@YZZ061 as a model) to create the PARMFILE data set that is required for historical data processing and workload configuration definitions.
Tip
This task is required for the following products but does not apply to other products:

- CMF MONITOR
- MainView for z/OS
- MainView for UNIX System Services (USS)
- MainView SYSPROG Services
- MainView VistaPoint

A PARMFILE data set is a VSAM data set that stores the names and current status of historical data sets. When historical data sets are listed in PARMFILE, you can dynamically enable them with commands from the DSLIST view.

Note
A separate PARMFILE data set is required for each PAS.

To allocate a PARMFILE data set

1. Follow the instructions at the top of BBILIB member @@YZZ061 to modify the JCL to your site's requirements.

2. Submit the job.

3. Check the job output for condition code 0.

   If the condition code is not 0, review the job output messages.

Creating the z/OS PAS startup procedure

Use the following procedure (which uses member BBMPAS as a model) to set up your z/OS PAS started task procedure.
This task is required for the following products but does not apply to other products:

- CMF MONITOR
- MainView for z/OS
- MainView for UNIX System Services (USS)
- MainView SYSPROG Services
- MainView VistaPoint

Note
When you create the z/OS PAS started task procedure, specify the same subsystem ID on the SSID parameter of the PROC statement as you did in the CAS started task procedure. The PAS uses this SSID during initialization to identify its CAS connection.

To create the started task procedure

1. Copy BBILIB member BBMPAS to SYS1.PROCLIB or another system procedure library and give it an appropriate name for a started task procedure.
   
   By default, MainView Customization creates a member called MVSPAS.

2. Follow the instructions at the top of the member to modify the procedure to your site requirements.

3. (optional) Define the z/OS PAS started task procedure to your IPL procedure so that the PAS starts at system initialization.

   BMC recommends that you modify the COMMNDxx member in SYS1.PARMLIB to add a START command for the PAS procedure:

   `COM='S proname`

   `proname` is the name of the z/OS PAS started task procedure.

   Note
   The START commands are issued in the order that they appear in COMMNDxx. The z/OS PAS START command must occur after the CAS START command.

4. If you do not define the z/OS PAS to start at IPL, see the MainView Administration Guide for information about manually issuing a START command for the PAS.
Where to go from here

During PAS execution, BMC recommends that you:

- Initiate the z/OS PAS as a started task (STC) through a system IPL procedure
- Execute only one z/OS PAS per system image
- Execute the z/OS PAS with a dispatching priority of 255

**Related Information**

- "Creating the CAS startup procedure" on page 69
- "z/OS PAS library member" on page 181

Creating a BBI-SS PAS PARMFILE data set

Use the following procedure (which uses member @@YZZ061 as a model) to create the BBI-SS PAS PARMFILE data set.

**Tip**

This task is optional for the following products and does not apply to other products:

- MainView AutoOPERATOR
- MainView for CICS
- MainView for DB2
- MainView for DBCTL
- MainView for IMS Online
- MainView for IMS Offline
- MainView for WebSphere MQ

You can use @@YZZ061 as a model to help you create the PARMFILE data set that is required for a BBI-SS PAS.
A PARMFILE data set is a VSAM data set that stores the names and current status of historical data sets. When historical data sets are listed in PARMFILE, you can dynamically enable them with commands from the DSLIST view.

**Note**
A separate PARMFILE data set is required for each PAS.

### To allocate a PARMFILE data set

1. Copy BBILIB member @@YZZ061 to a UBBSAMP data set.

   @@YZZ061 contains sample JCL to allocate a VSAM data set with IDCAMS DEFINE CLUSTER statements.

2. Edit the @@YZZ061 member as follows:

   a. Change all instances of ?BAVSHLQ to the high-level qualifier that you chose for your PARMFILE data set.

   b. Change all instances of ?BBCHILV to the high-level qualifier that you chose for your installation libraries.

   c. Change all instances of ?BAVIMID to the subsystem ID of the BBI-SS PAS.

   d. Change all instances of ?BBVVOL to the name of the volume where the data sets will be allocated.

   e. Change all instances of ?BBPUNIT to the unit name of the volume that will store the PARMFILE data set.

3. Submit the @@YZZ061 job.

   The PARMFILE data set is allocated and formatted using an initial record format that is shipped in BBILIB member @@YZZ062.

4. Check the job output for condition code 0.

   If the condition code is not 0, review the job output messages.
Defining BBI-SS PAS suffixes and target system parameters

Use the following procedures to define all eligible target systems and associate them with the subsystem IDs of their BBI-SS PASs.

**Tip**

This task is required for the following products but does not apply to other products:

- MainView FOCAL POINT
- MainView AutoOPERATOR
- MainView for CICS
- MainView for DB2
- MainView for DBCTL
- MainView for IMS Online
- MainView for IMS Offline
- MainView for WebSphere MQ

You need to:

- Decide which target you want to associate with which BBI-SS PAS
- Specify the targets and BBI-SS PASs in BBPARM member BBIJNT00

When a PAS first starts, the PAS contacts the CAS and retrieves a list of defined targets and targets that are active on other PASs from the CAS. The PAS appends this information to a list of targets that are built during startup from BBPARM member BBIJNT00.

**Note**

For MainView for CICS, you can use an administrative dialog to define and add targets. Target definitions that are created with this dialog are activated only when the INSTall command is entered. Existing targets that are defined in BBIJNT00 cannot be changed with the administrative dialog. For more information, see the information about managing targets and target contexts in the MainView Administration Guide.
To define BBI-SS PAS suffixes and target system parameters

1 Define a one- to four-character subsystem ID to dynamically establish a formal MVS subsystem.

The name that you specify:

- Must be unique from any other subsystem name and cannot be in the MVS subsystem name table (IEFSSN.xx)

**Note**
If the subsystem ID that you specify is in IEFSSN.xx, message SS0974E is issued and initialization terminates.

- Cannot be DB2, IMS, CICS, or JES, which are reserved by MVS
- Must not be the same as the name of any other BBI-SS PAS or CAS that is currently in use

2 Define a one- to eight-character job name for each target to be managed by a BBI-SS PAS.

**Note**
If the target is a DB2 system, specify the one- to four-character name of that system. If the target is a queue manager, specify the one- to four-character name of that queue manager.

Each BBI-SS PAS can manage the following targets on a single system image:

- 128 CICS regions
- 1 IMS control region or DBCTL region
- 15 DB2 regions at the same release level

A MainView for DB2 PAS can support monitoring of DB2 subsystems at two different DB2 releases. To do this monitoring, the appropriate Call Attach (CAF) compatibility level must be available in the DB2 load library used in the PAS. Contact IBM for further information about the two releases that you want to access. Depending on the release level, this access might require upward-compatible PTFs and the use of the lower version load library. With DB2 Version 8.1, downward compatibility might be available.

- System where the BBI-SS PAS runs
- Multiple queue managers

BMC recommends that you monitor no more than 15 MVS queue managers per PAS. If you monitor more than 15, S32E-11C abends might occur. These abends
are caused by WebSphere MQ issuing too many STIMERM SET requests on behalf of the PAS.

3 Define the type of target, which can be CICS, DB2, IMS, MVS, or QMGR.

4 For DB2 or IMS targets, identify the target system release.

5 For IMS targets, identify the type of target as DBDC or DBCTL.

6 (optional) Provide a 1- to 22-character description of the target system.

7 Use the information that you gathered in the preceding steps to modify the parameters in your copy of BBPARM member BBIJNT00.

These parameters specify a target/BBI-SS PAS pair and other systems with which this BBI-SS PAS must connect. You can specify multiple TARGET statements for different target systems that are monitored from the same BBI-SS PAS.

To specify a target system for this BBI-SS PAS

In this procedure, you will use member BBIJNT00, which supports the substitution of system variables. Use the following conventions when editing BBIJNT00:

- All statements must be contained within columns 1 through 71.
- Statements can be continued across multiple lines.
- If a statement is continued, the TARGET parameter must appear on the first line.
- Comments, which begin with an asterisk (*) in column 1, can be inserted anywhere, even between continued statements.
- The TARGET parameter and comments must begin in column 1; continuations do not have to begin in column 1.

1 Make the following entry in BBIJNT00 to specify a target system for this BBI-SS PAS:

```
TARGET=tgtname,TYPE=type,
IMSTYPE=systype,
SUBSYS=subsys,RELEASE=vr0
[DESC=desc][,IMSID=ssid][,ALIAS=name][,ATTACH=IMM | DEFER]
```

Use the following table to specify the parameters of the entry:
### Defining BBI-SS PAS suffixes and target system parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TARGET</td>
<td>Specify a one- to eight-character name of a target system to be monitored by a BBI-SS PAS as follows:</td>
</tr>
<tr>
<td></td>
<td>- For MainView AutoOPERATOR, specify the name of a target CICS, IMS, or MVS, such as TARGET=MVSPROD.</td>
</tr>
<tr>
<td></td>
<td>- For MainView for WebSphere MQ, specify the queue manager subsystem ID (one to four characters), such as TARGET=CSQ1. It can be any name used to relate to a non-MVS queue manager name. The target name must be unique within a plex of BBI-SS PASs.</td>
</tr>
<tr>
<td></td>
<td>- For MainView for CICS, specify the job or step name of a target CICS, such as TARGET=CICSA1. The target name must be unique within a plex of BBI-SS PASs.</td>
</tr>
<tr>
<td></td>
<td>- For MainView for IMS Online, specify the name of a target IMS, such as TARGET=IMS71X.</td>
</tr>
<tr>
<td></td>
<td>- For MainView for DBCTL, specify the job name or IMS ID of the DBCTL region, such as TARGET=IMSDBCP.</td>
</tr>
<tr>
<td></td>
<td>- For MainView for DB2, specify a DB2 subsystem ID (one to four characters), such as TARGET=DB2P.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| TYPE      | Specify the type of target as follows:  
- For MainView AutoOPERATOR, specify TYPE=CICS for a CICS target or TYPE=IMS for an IMS target. MainView AutoOPERATOR always requires one TYPE=MVS specification for each BBI-SS PAS.  
- For MainView for WebSphere MQ, specify TYPE=QMGR for a queue manager target.  
- For MainView for CICS, specify TYPE=CICS for a CICS target.  
- For MainView for IMS Online, specify TYPE=IMS for an IMS target.  
- For MainView for DBCTL, specify TYPE=IMS for an IMS target.  
- For MainView for DB2, specify TYPE=DB2 for a DB2 target. |
| IMSTYPE   | Specify the type of IMS target. Specify IMSTYPE=DBDC for an IMS DB/DC system (the default) or IMSTYPE=DBCTL for a DBCTL system. The IMSTYPE parameter is valid only when TYPE=IMS is specified. |
| SUBSYS    | Specify a one- to four-character subsystem ID that is assigned to a BBI-SS PAS. If only one BBI-SS PAS is created, use the value SSA1. |
| RELEASE   | For IMS, DB2 and DBCTL targets, specify the four-digit release number of the target. For example, RELEASE=0810 specified with TYPE=DB2 indicates DB2 Version 8.1. |
| DESC      | *(optional)* For MainView FOCAL POINT, specify a 1- to 22-character description of the target system. If the description contains any embedded commas or blanks, it must be enclosed in single quotation marks:  
'xxx ... xxx' |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| IMSID     | *(optional)* For IMS targets, specify a unique one- to four-character subsystem ID of the IMS control region that you want to monitor.  
**Note:**  
- For IMS targets, if you specify IMSID, it overrides what was specified by the TARGET statement.  
- If you specify IMSID, you cannot specify ALIAS for an IMS target. |
| ALIAS     | *(optional)* Specify the ALIAS parameter as follows:  
- For MainView AutoOPERATOR for IMS, MainView for DB2, or MainView for IMS Online, specify a one- to eight-character alias name for the target subsystem. If you have two targets with the same tgtname controlled by different BBI-SS PASs, you must define an alias name, such as ALIAS=PRODDB2.  
**Note:**  
- If you specify ALIAS, it overrides what was specified by the TARGET statement.  
- If you specify ALIAS for an IMS target, you cannot specify IMSID.  
- For MainView for CICS, specify the one- to eight-character APPLID that is specific to the CICS region. The APPLID (which is used as the alias name) must be unique within a plex of BBI-SS PASs.  
**Note:**  
- If you do not specify ALIAS for a CICS target, the alias name defaults to the TARGET name (which must also be unique within a plex of BBI-SS PASs).  
- The alias name for a CICS target can only be used in full-screen mode. |
For Group 3 products, the ATTACH parameter requests immediate or deferred target recognition when the BBI-SS PAS starts.

- DEFER (the default) causes the service point to be created after the target is active. Data is not supplied in response to queries until the target becomes active or a monitor is started that collects data from that region. In most cases, ATTACH=DEFER should be used.

- IMM causes the service point to become active when the BBI-SS PAS initializes. It remains active regardless of the target state, which is useful for viewing historical data from that target when it is no longer active. The target status is shown as active in the Plex Manager views. The PAS can collect data from the target, but that target itself is not active.

---

**Related Information**

- “System variables substitution in parameter library members” on page 233

---

**Defining configuration members to BBI-SS PAS**

Use the following procedure to define configuration members to BBI-SS PAS.
This task is optional for the following products and does not apply to other products:

- MainView FOCAL POINT
- MainView AutoOPERATOR
- MainView for CICS
- MainView for DB2
- MainView for DBCTL
- MainView for IMS Online
- MainView for IMS Offline
- MainView for WebSphere MQ

Use member BBIBBCFG as a model for the CFGssidA member.

CCFssidA defines the member name suffixes of the BBPARM configuration members. These members control processing, such as:

- How the BBI-SS PAS operates
- How certain MainView AutoOPERATOR functions operate
- How certain MainView for CICS functions operate

The BBPARM configuration members are described in "BBIBBCFG parameters".

If you specify different member name suffixes, multiple BBI-SS PASs can process different configuration members from the same BBPARM library. The CFG ssidA member is read during BBI-SS PAS startup to determine the suffixes of the configuration members to be used by that PAS.

**To define configuration members to the BBI-SS PAS**

1. In your UBBPARM data set, create a member named CFGssidA, where ssid is the BBI-SS PAS subsystem ID.

2. Copy the distributed BBPARM member BBIBBCFG into the CFGssidA member that you created.

3. In CFGssidA, specify any of the following parameters:
AAOALS=xx
AAOARP=xx
AAOEXP=xx
AAOGME=xx
AAOMQL=xx
AAOPRM=xx
AAOTSP=xx
BBIISP=xx
BBISSP=xx
BBKMON=xx
CMRBEX=xx
CMRDTL=xx

xx is the two-character suffix of a configuration member.

Some of these configuration members support the substitution of system variables. The use of system variables and member suffixes enables multiple systems to share a single library. For example, in BBIBBCFG, you can specify BBIISP=&SYSCLONE and use a single BBIBBCFG member for all systems (if they are on different system images). If all that is required is a different TARGET value for different subsystems, you can specify TARGET=SS&SYSCLONE rather than using suffixes to create separate members.

4 In the BBI-SS PAS startup procedure, add a BBCFG DD statement that refers to CFGssidA, as described in Creating a BBI-SS PAS startup procedure on page 121.

Related Information

- “System variables substitution in parameter library members” on page 233
- “Creating a BBI-SS PAS startup procedure” on page 121
- “BBIBBCFG parameters” on page 199
Defining BBI-SS PAS interval services parameters

Use the following procedure to specify parameters that initialize timer services automatically at BBI-SS PAS startup and identify the trace directory data set.

Tip
This task is optional for the following products and does not apply to other products:

- MainView FOCAL POINT
- MainView AutoOPERATOR
- MainView for CICS
- MainView for DB2
- MainView for DBCTL
- MainView for IMS Online
- MainView for IMS Offline
- MainView for WebSphere MQ

Use member BBIISPxx to set up timer services and the trace directory.

To define BBI-SS PAS interval services parameters

1. In member BBIISPxx, specify the name of a BBPARM member that defines the group of timer-driven services that are to be started at BBI-SS PAS startup:

   AUTOID=userid
   TARGET=tgtname, BLK=blkName[, USRID=userid][, ALERTQ=qname]

<table>
<thead>
<tr>
<th>AUTOID</th>
<th>TARGET</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTOID</td>
<td>Specify the logon ID of a user that is to be associated with automatically started timer requests. AUTOID is the default if USERID is not specified. AUTOID must be on a separate line from TARGET and BLK, as in the following example:</td>
</tr>
<tr>
<td>ADMIN</td>
<td>AUTOID=ADMIN</td>
</tr>
<tr>
<td></td>
<td>TARGET=DB2A, BLK=BLKDB2A</td>
</tr>
<tr>
<td></td>
<td>TARGET=DB2B, BLK=BLKDB2B</td>
</tr>
<tr>
<td>TARGET</td>
<td>Specify the target system that is to receive the requests for BLK. It can be the target system ID or the subsystem ID. TARGET must be on the same line as BLK and can be repeated.</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BLK</td>
<td>Specify the one- to eight-character name of a BBPARM member that defines a collection of timer-driven SET requests for product services that can be initiated at the same time. BLK must be on the same line as TARGET and can be repeated. The BBPARM members are as follows:</td>
</tr>
<tr>
<td></td>
<td><strong>MainView for CICS</strong></td>
</tr>
<tr>
<td>CMRBLK1</td>
<td>Contains a group of background service requests that are used when MainView for CICS data collection is not active</td>
</tr>
<tr>
<td>CMRBLK2</td>
<td>Contains a group of background service requests that are used when MainView for CICS data collection is active</td>
</tr>
<tr>
<td>CMRBLK3</td>
<td>Contains a group of background service requests that are used when MainView for CICS data collection is active and MainView FOCAL POINT is installed</td>
</tr>
<tr>
<td></td>
<td><strong>MainView for DB2</strong></td>
</tr>
<tr>
<td>BLKDMRW</td>
<td>Can be used to create a starter set of monitors and traces to provide recent DB2 history</td>
</tr>
<tr>
<td></td>
<td><strong>MainView for DBCTL</strong></td>
</tr>
<tr>
<td>BLKDBTW</td>
<td>Can be used to create a starter set of monitors for a DBCTL system</td>
</tr>
<tr>
<td></td>
<td><strong>MainView for IMS Online</strong></td>
</tr>
<tr>
<td>BLKIMFW</td>
<td>Can be used to create a starter set of monitors for IMS</td>
</tr>
<tr>
<td><strong>Note:</strong> You can use the BLKxxxx members as a base from which to build a list of monitors to be started automatically. These members define suggested warning thresholds; some show only your site's activity.</td>
<td></td>
</tr>
</tbody>
</table>
**USRID**

*(optional)* Specify the logon ID of a user that is to be associated with the timer requests in the member specified by the BLK parameter. This user has the authority to modify or purge the timer requests.

The value for *userid* must match a BBPARM user authorization member (see BBPARM member USERID).

USRID is optional and is written on the same line as TARGET and BLK. It supersedes AUTOID and associates the specified user with the group of requests defined for that BLK parameter, as in the following example:

```
TARGET=DB2A,BLK=BLKDB2A,USRID=$TDB2AR
```

If USRID is not specified, the default is AUTOID. If AUTOID and USRID are not specified, a default user ID of USERID is used. A USERID member must exist in the BBPARM data set.

**ALERTQ**

*(optional)* Specify an alert queue name for CICS, DB2, IMS, or DBCTL targets. This parameter activates automatic creation of alerts for monitor exception messages, including background exception messages in MainView for DB2. The alerts are displayed in the MainView Alarm Manager version 2.1 alert views.

---

**Note**

If these exception messages are already being processed into alerts by MainView AutoOPERATOR rules for journal messages, the ALERTQ parameter is not needed. Specifying ALERTQ could result in duplicate alerts.

---

2. Identify the name of the data set that will be used as a trace directory by the following products:

- MainView for CICS
- MainView for DB2
- MainView for DBCTL
- MainView for IMS Online

---

**Note**

This data set *cannot* be shared by multiple BBI-SS PASs.

---

Specify the data set name of the trace directory as:

```
TRDIR=dsn[,SUBSYS=ssid]
```

- *dsn* is a valid data set name.
- *ssid (optional)* is the subsystem name to which this trace directory applies. This parameter allows multiple BBI-SS PASs to share the same BBIISPxx member.

Trace logging can be set up automatically by using the MainView Customization procedures that are described in the *Installation System User Guide*. Your product customization guide describes how to set up and maintain trace logging manually.

---

# Setting the BBI-SS PAS historical recording interval

Use the following procedure (which uses member BBIISPxx) to specify a recording interval when records are to be written to a historical data set.

---

**Tip**

This task is optional for the following products and does not apply to other products:

- MainView AutoOPERATOR
- MainView for CICS
- MainView for DB2
- MainView for DBCTL
- MainView for IMS Online
- MainView for IMS Offline
- MainView for WebSphere MQ

Records are written from the interval recorder to a historical data set at the end of every recording interval. The IRRI statement in member BBIISPxx specifies the length of the recording interval. Views that display current data have interval and real-time data-collection periods that are based on the IRRI value.

### To set the recording interval

1. Add an IRRI statement to member BBIISPxx.

2. On the IRRI statement, specify a recording interval in hours and minutes, as follows:
IRRI=hh:nn

For example, to set the historical interval recording interval to 10 minutes, specify

IRRI=00:10

The shortest recording interval that you can specify is two minutes (IRRI=00:02). The longest is one hour (IRRI=01:00).

Note
If the IRRI statement is not specified in member BBIISP.xx, the length of the recording interval uses a default value of 15 minutes.

Allocating BBI-SS PAS journal and image logs

Use the following procedure to allocate the journal and image log data sets that are used during execution of a BBI-SS PAS.

Tip
This task is optional for the following products and does not apply to other products:

- MainView FOCAL POINT
- MainView AutoOPERATOR
- MainView for CICS
- MainView for DB2
- MainView for DBCTL
- MainView for IMS Online
- MainView for IMS Offline
- MainView for WebSphere MQ

Member SSLOG contains sample JCL to allocate BBI-SS PAS journal and image log data sets. For MainView AutoOPERATOR, it also contains a sample statement to allocate a profile variable pool called BBIVARS for MainView AutoOPERATOR EXECs.
Note

- Do not alter the log attributes. The default attributes are required and are reset automatically if you change them.

- If you are using products on multiple systems and want to use the same JCL procedure for all of them, include the SSID in the data set name of the logs. All of the subsystems can then use the same procedure.

- BMC recommends the use of image logs for MainView for CICS, MainView for IMS Online, MainView for DB2, and MainView for DBCTL.

- BBI-SS PAS image logs are not used by MainView AutoOPERATOR or MainView for WebSphere MQ.

To allocate the BBI-SS PAS journal and image log data sets

1. In BBSAMP member SSLOG, make the following global changes (including the S1, S2, and S3 statements at the end of this job):
   a. Change the PREFIX parameter to the prefix that was used to load the product libraries to DASD.
   b. Set the SS parameter to the subsystem ID of the BBI-SS PAS.
   c. Set the DV parameter to the volume serial of the allocation volume for the BBI-SS PAS journal data sets.
   d. Set the DU parameter to the allocation volume unit type.

2. If you do not have MainView for CICS, MainView for IMS Online, MainView for DBCTL, or MainView for DB2, comment out the ALLOCIMG (S1) procedure.

3. If you do not have MainView AutoOPERATOR, comment out the ALLOCBVP (S3) procedure.

Note

If you have MainView AutoOPERATOR, the ALLOCBVP statement is required to allocate the BBIVARS profile variable pool for MainView AutoOPERATOR EXECs.
Allocating MainView Total Object Manager data sets

Use the following procedure to allocate the data sets that are required by MainView Total Object Manager (TOM).

**Tip**
This task is optional for MainView AutoOPERATOR and does not apply to other products.

This procedure uses member TOMALLOC to allocate a TOM registry data set and two log data sets for TOM.

**To allocate the TOM data sets**

1. Copy BBSAMP member TOMALLOC to your UBBSAMP data set.

2. Make the following changes to TOMALLOC:
   
   a. Add a valid JOB card for your site.
   
   b. Change all occurrences of `#prefix#` to the high-level qualifier for the TOM data sets.
   
   c. Change all occurrences of `#ssid#` to the subsystem ID of the BBI-SS PAS with which this instance of TOM will be associated.
   
   d. Change all occurrences of `#volume#` to the DASD volume where you want to store the TOM data sets.

3. Save the member and submit the job.

**Note**
TOM will issue a WTOR message whenever it encounters an uninitialized registry or log data set. This process ensures that the TOMALLOC JCL specifies the correct data set name. After you reply to the WTOR, TOM formats the data set and continues its initialization.
Allocating MainView AutoOPERATOR data sets

Use the following procedure to allocate the BBIDIV data set that is required for MainView AutoOPERATOR.

Tip
This task is required for MainView AutoOPERATOR and does not apply to other products.

Use member DIVDEF to allocate a BBIDIV data set, which is required to operate MainView AutoOPERATOR. This data set stores binary large objects and data across restarts.

Before you begin

Note
■ The BBI-SS PAS must be shut down when the BBIDIV data set is allocated.
■ Each BBI-SS PAS must have its own copy of the BBIDIV data set; it cannot be shared.

To allocate the BBIDIV data set

1. In BBSAMP member DIVDEF, modify the job statement to your site’s requirements.

2. Make the following changes to the procedure header:
   a. Change *HILEVEL* to the high-level qualifier for MainView product data sets.
   b. Change *SSID* to the four-character subsystem name.
   c. Change *VOL* to the volume serial number of the DASD volume where the linear data set will be allocated.

3. Submit the modified job.

Creating a BBI-SS PAS startup procedure

Use the following procedure to set up a started task procedure for the BBI-SS PAS.
This task is required for the following products but does not apply to other products:

- MainView FOCAL POINT
- MainView AutoOPERATOR
- MainView for CICS
- MainView for DB2
- MainView for DBCTL
- MainView for IMS Online
- MainView for IMS Offline
- MainView for WebSphere MQ

Use member SSJCL to help you set up your BBI-SS PAS started task procedure.

Note

- The default region size for the BBI-SS PAS is 64 MB.
  If you reduce this allocation and not enough working storage is available after initialization, the BBI-SS PAS terminates with an error message.

- The job contained in member SSJCL should run as a started task; if not, a JES initiator is occupied for the duration of the BBI-SS PAS session.
  To run SSJCL as a started task, remove the JOBCARD statement and the PEND and EXEC statements at the end of the JCL. Then copy the SSJCL JCL procedure to a JES PROCLIB.

- For optimum BBI-SS PAS performance, the BBI-SS PAS dispatching priority should be equal to or greater than any of the target subsystems that the PAS manages.
  If the dispatching priority is too low, loss of data or excessive storage usage could occur.

- If you have IBM RACF or an equivalent external security manager (ESM), verify that the BBI-SS PAS has authorization to access and update all MainView product data sets.

- The PARM statement in member SSJCL defines the subsystem ID that is assigned to the BBI-SS PAS.
To create a BBI-SS PAS started task procedure

1. In BBSAMP member SSJCL, make the following changes:
   a. Change the PREFIX parameter to the prefix that was used to load the product libraries to DASD.
   b. Change the SS parameter value to the subsystem ID of the BBI-SS PAS.
   c. Use the default parameter START=WARM.

   START=COLD should be used only as instructed by BMC Customer Support or when the BBI-SS PAS requests it through WTOR after maintenance.

2. Ensure that your UBBPARM data set is concatenated before the BBPARM data set.

3. (MainView AutoOPERATOR and MainView for WebSphere MQ only) Complete this step only if one of the following conditions applies to you:
   - You plan to use MainView AutoOPERATOR General Message Exchange (GME) or MainView AutoOPERATOR Command MQ Automation Power Line (APL).
   - You plan to connect a BBI-SS PAS from MainView for WebSphere MQ without using a PATROL for WebSphere MQ node manager.

   If either condition applies, perform the following actions to enable the BBI-SS PAS to use the IBM TCP/IP product or the BMC Generic TCP/IP Server (GTS):

   a. If the following TCP/IP libraries are not available in the system linklist, add the libraries to the STEPLIB:

      \[\text{tcpprefix.SEZATCP}\]
      \[\text{tcpprefix.SEZALINK}\]

      \[\text{tcpprefix}\] is the prefix that was defined when you installed TCP/IP.

   b. (MainView for WebSphere MQ only) If the following TCP/IP library is not available in the system linklist, add the library to STEPLIB:

      \[\text{tcpprefix.SEZALPA}\]

   c. (MainView for WebSphere MQ only) Add the following WebSphere MQ-authorized program library to the STEPLIB DD statement:

      \[\text{wmqprefix.SCSQAUTH}\]
wmqprefix is the prefix that was defined when you installed WebSphere MQ.

d If you are running multiple TCP/IP stacks and you want to specify which stack the PAS should use, add the following TCP/IP configuration DD statement to member SSJCL:

//SYSTCPD DD DISP=SHR,DSN=tcpprefix.CNTL(tcpipdata)

tcpipdata is the name of the member that contains the TCP/IP configuration information that is required by TCP/IP client programs.

You do not need to identify a specific TCP/IP stack to the BBI-SS PAS. If more than one stack is running, the PAS will use the first stack that is found with the latest release.

e To add BMC Generic TCP/IP Server (GTS) support, in BBPARM member BBISSPxx, specify

GTS=xx

xx is the suffix of your BBTTCPxx member.

f (required for MainView for WebSphere MQ; optional for MainView AutoOPERATOR) Create BBPARM member BBTTCPxx in your copy of the BBPARM library where xx is 00 or the suffix that you specified with the GTS parameter in member BBISSPxx.

Note

If you decide not to complete this step, you must specify GTS=00 in BBPARM member BBISSPxx (see Step 3.e on page 124).

BBTTCPxx defines your TCP/IP environment and has the following statements:

TCPNAME=name
BMC recommends that you not specify any parameters in member BBTTCPxx. 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 active on the system.

If you are starting the BBI-SS PAS address space before TCP/IP, GME will not activate during BBI-SS initialization. Both GTS and GME must be activated after TCP/IP is started. You can activate them by issuing the following two BBI control commands:

```
.S GTS
.E P AAOGME,RECYCLE
```

For complete information about BBTTCPxx, see BBTCP00 parameters on page 228.

4 (MainView AutoOPERATOR only) Perform the following steps:

a Create a SYSPROC DD statement that points to the libraries that contain MainView AutoOPERATOR EXEC members:

- Uncomment the SYSPROC DD statement.
- Ensure that the UBBPROC data set is concatenated before the BBPROC data set.

b Allocate a BBIDIV data set to store binary large objects and data across MainView AutoOPERATOR restarts:

- Insert the following DD statement immediately before the PEND statement:

  ```
  //BBIDIV DD DSN=&PREFIX.&SS..BBIDIV,DISP=OLD
  ```

Note
Do not specify the BBIDIV DD statement by using a DUMMY (or null) data set. Such a statement might cause an S013-64 abend.

- For information about how to allocate the BBIDIV data set, see Allocating MainView AutoOPERATOR data sets on page 121.

5 (MainView for DB2 only) If you are also installing the MainView for DB2 Data Collector component, include the Data Collector load library in the PAS STEPLIB concatenation with the correct high-level qualifier.

You can use the sample CDCLOAD DD statement that is included in the procedure as an example.

6 (optional) Add the following statement to your configuration members:
Adding the CICS EXCI library to the BBI-SS startup procedure

Use the following task to add the CICS EXCI library to the BBI-SS startup procedure.

The external CICS interface modules are supplied in the CICS data set SDFHEXCI.

**Tip**

This task is required for the MainView AutoOPERATOR for CICS option and for MainView for CICS, and does not apply to other products.

The SDFHEXCI data set must be available to the MainView AutoOPERATOR for CICS and MainView for CICS' BBI-SS PAS for normal system searches.

**To add the CICS EXCI library to the BBI-SS startup procedure**

1. If the SDFHEXCI data set is not defined to the system LINKLIST, add it with a DD statement in the STEPLIB concatenation for the BBI-SS PAS.

The data set must be APF authorized.

IBM recommends that you specify the CICS data set of the latest version of CICS that communicates with the BBI-SS PAS. For example, if you are running a CICS 4.1.0 and a CICS 5.1.0 system that both communicate with the BBI-SS PAS, the JCL update might look like this example:

```
//STEPLIB DD DISP=SHR,DSN=hlq.BBLINK
//      DD DISP=SHR,DSN=CICS510.SDFHEXCI (FOR EXCI)
```
2 Include a SYSMDUMP DD statement in the BBI-SS PAS start up JCL.

The external CICS interface uses SYSMDUMP for some error conditions and BMC recommends that you include a SYSMDUMP DD statement in the BBI-SS PAS start up JCL.

Example

//SYSMDUMP DD SYSOUT=*  

3 If you start the BBI-SS PAS before JES, ensure that you add the DD statement for SYSMDUMP to the list of data sets that must point to real data sets, and enter the SYSMDUMP statement as follows:

//SYSMDUMP DD DISP=SHR,DSN=sysmdump.dataset,VLSER=.....,UNIT=.....  

4 The REGION parameter must specify a region large enough to accommodate the internal trace table that is specified by the TRACESZE parameter (the default setting is 16K) in the DFHXCOPT macro, EXCI options table.

For more information, see the IBM book *CICS External Interfaces Guide*.

Creating the MainView Total Object Manager start procedure

Use the following procedure to set up a started task procedure for MainView Total Object Manager (TOM).

Tip

This task is optional for MainView AutoOPERATOR and does not apply to other products.

Use member TOMJCL as a model to help you set up your TOM started task procedure.

To create the TOM started task procedure

1 Copy BBSAMP member TOMJCL to SYS1.PROCLIB or another system procedure library.

2 Make the following changes to modify TOMJCL to your site requirements:

- Change PREFIX to the high-level qualifier for the TOM data sets, as specified in Allocating MainView Total Object Manager data sets on page 120.
■ Change ID to a four- to eight-character alphanumeric ID for this TOM address space.

  **Note**
The ID should start with an alphabetic character.

■ Change SS to the subsystem ID of the MainView AutoOPERATOR BBI-SS PAS with which this instance of TOM will be associated.

  **Note**
Ensure that the PAS that is identified in the SS parameter is running before you start this TOM address space. TOM depends on services that are provided by MainView AutoOPERATOR.

**Related Information**

■ "Allocating MainView Total Object Manager data sets" on page 120

---

**Creating automatic BBI-SS PAS log handling procedures**

Use the following procedures to create an automatic journal log print and reset procedure and an image log print procedure for the BBI-SS PAS.
**Tip**

This task is required for the following products but does not apply to other products:

- MainView FOCAL POINT
- MainView AutoOPERATOR
- MainView for CICS
- MainView for DB2
- MainView for DBCTL
- MainView for IMS Online
- MainView for IMS Offline
- MainView for WebSphere MQ
- MainView Alarm Manager version 2.1

**Note**

Use member BBISSP.xx to specify started task procedures for automatic journal and image log handling.

---

**To set up a started task to print and reset the BBI-SS PAS journal log**

Use member DLOGJCL to set up a started task to print and reset the BBI-SS PAS journal log.

1. To create an automatic journal log print and reset procedure, perform the following steps:

   a. In BBPARM member BBISSP.xx, specify a started task name with the following statement:

      ```
      JRNLSTC=xxxxxxxx
      ```

      `xxxxxxxx` is a one- to eight-character name of a started task that is used to print and reset the BBI-SS PAS journal logs. BBIDLOG is the default. The value that you enter must match the name of the started task that you add to your JES PROCLIB.

   b. Copy BBSAMP member DLOGJCL into your JES PROCLIB data set and rename the member to the value that you specified for JRNLSTC in BBISSP.xx.

   c. Make the following changes to the JCL:
In the PROC statement, change BBIDLOG to the name that you want to use for your journal log print and reset procedure.

In the PROC statement, change the PREFIX parameter to the high-level qualifier that was used to load product libraries to DASD.

d (optional) To print the BBI-SS PAS journal log to a data set on disk or tape, add `DCB=(RECFM=FBA,LRECL=121)` to the BBIPRINT DD statement.

Any suitable BLKSIZE is allowed.

The journal log print and reset procedure is ready for use. If the product is already running when this step is completed, the BBISSPxx parameter change does not take effect until the BBI-SS PAS is restarted.

**Note**
When the BBI-SS PAS journal log is full, or if an I/O error is encountered, a SWITCH JOURNAL command is issued and user exit BBIUSR01 is invoked. BBIUSR01 invokes a started task that prints and resets the journal log. In the case of a journal I/O error, an I/O error message is issued before the started task is invoked.

The source for BBIUSR01 is distributed in the BBSAMP data set and can be modified. BBSAMP member ASMLKED contains sample JCL that you can use to assemble and link-edit the BBIUSR01 user exit after modification. However, BMC recommends not modifying BBIUSR01. BMC Customer Support can offer assistance only if the exit does not operate as described and it has not been modified. In BBPARM member BBISSP, copy BBSAMP member DLOGJCL into your JES PROCLIB data set. Rename the member to the value that you specified for JRNLSTC in BBISSP.

### To create an automatic image log print procedure

Use ILOGJCL to set up a started task to print the BBI-SS PAS image log, if your products use this image log.

**Note**
MainView Alarm Manager version 2.1 does not use an image log; it uses only a journal log. If you are customizing MainView Alarm Manager version 2.1, you can skip the information related to ILOGJCL and the image log.

1. To create an automatic image log print procedure, perform the following steps:

   a. In BBPARM member BBISSPxx, specify a started task name with the following statement:

      `IMAGSTC=xxxxxxxx`
xxxxxx is a one- to eight-character name of a started task that is used to print the BBI-SS PAS image logs. BBIILOG is the default. The value that you enter must match the name of the started task that you add to your system PROCLIB.

b. Copy BBSAMP member ILOGJCL into your JES PROCLIB data set and rename the member to the value that you specified for IMAGSTC in BBISSPxx.

c. Make the following changes to the JCL:

- In the PROC statement, change BBIILOG to the name that you want to use for your image log print procedure.
- In the PROC statement, change the PREFIX parameter to the high-level qualifier that was used to load product libraries to DASD.
- Change all occurrences of SSID to the subsystem ID of the BBI-SS PAS.

The image log print procedure is ready for use. If the product is already running when this step is completed, the BBISSPxx parameter change does not take effect until the BBI-SS PAS is restarted.

Note

When the BBI-SS PAS image log is full, or if an I/O error is encountered, a SWITCH IMAGE command is issued and user exit BBIUSR02 is invoked. BBIUSR02 invokes a started task that prints the image log. In the case of an image log I/O error, an I/O error message is issued before the started task is invoked.

The source for BBIUSR02 is distributed in the BBSAMP data set and can be modified. BBSAMP member ASMLKED contains sample JCL that you can use to assemble and link-edit the BBIUSR02 user exit after modification. However, BMC recommends not modifying BBIUSR02. BMC Customer Support can offer assistance only if the exit does not operate as described and it has not been modified.

**Defining terminal session parameters**

Use the following procedure to define terminal session target system and primary product application defaults, and unique terminal session user profiles.
Tip
This task is optional for the following products and does not apply to other products:

- MainView FOCAL POINT
- MainView AutoOPERATOR
- MainView for CICS
- MainView for DB2
- MainView for DBCTL
- MainView for IMS Online
- MainView for IMS Offline
- MainView for WebSphere MQ

Note
The applicable products use the parameters in BBPROF members BBITSP00 and xxxxPFK when the terminal session initializes. Instead of manually customizing BBITSP00, you can use Option 0 from the MainView Selection Menu to set up terminal session parameters. Using Option 0, you can also select the terminal session options for all of your other MainView products.

Use member BBITSP00 to set up your terminal session user profiles, target system, and primary product application.

To set up your terminal session user profiles, target system, and primary product application

1. Create the following user profile:

   `userid.BBPROF`

   `userid` is the user's TSO logon ID. The user profile can be concatenated in front of the site's common BBPROF data set.

   Each user should have his or her own BBPROF data sets in order to specify:

   - Unique PF keys
   - CYCLE commands
   - Target system defaults
- A Primary Option Menu
- A unique set of application profiles
- User graph members (MainView for CICS)

When you create a user profile, the MainView CLIST creates a user BBPROF automatically with a PROFILE(YES) statement. If PROFILE(NO) is specified, the BBPROF data set is shared.

**Tip**
You can also create a site-specific profile called SBBPROF. If you use a common BBPROF data set, profile members for product applications are updated automatically by the last person who used that application.

2 In BBPROF member BBITSP00, specify the following terminal session parameters:

```
TARGET=tgtname,TYPE=appltype,PRIMMNU=pnlnname
PRIMAPPL=appltype
ISPF=YES | NO
[BBIRESP=LOCAL]
[IMSRESP=LOCAL]
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TARGET</td>
<td>Specify a one- to eight-character target name as follows:</td>
</tr>
<tr>
<td></td>
<td>- CICS target (MainView for CICS)</td>
</tr>
<tr>
<td></td>
<td>- CICS target DB2 target (MainView for DB2)</td>
</tr>
<tr>
<td></td>
<td>- IMS or DB2 target (MainView for IMSOnline or MainView for DBCTL)</td>
</tr>
<tr>
<td></td>
<td>- IMS, CICS, or MVS target (MainView AutoOPERATOR)</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>TYPE</strong></td>
<td>Specify the product that is related to the target specified in the TARGET parameter as follows:</td>
</tr>
<tr>
<td></td>
<td>■ AO (MainView AutoOPERATOR)</td>
</tr>
<tr>
<td></td>
<td>■ CICS (MainView for CICS)</td>
</tr>
<tr>
<td></td>
<td>■ DB2 (MainView for DB2)</td>
</tr>
<tr>
<td></td>
<td>■ IMS (MainView for IMS Online and MainView for DBCTL)</td>
</tr>
<tr>
<td><strong>PRIMAPPL</strong></td>
<td>Specify the primary application to be displayed when the terminal session starts as follows:</td>
</tr>
<tr>
<td></td>
<td>■ AO (MainView AutoOPERATOR)</td>
</tr>
<tr>
<td></td>
<td>■ CICS (MainView for CICS)</td>
</tr>
<tr>
<td></td>
<td>■ IMS (MainView for IMS Online and MainView for DBCTL)</td>
</tr>
<tr>
<td></td>
<td>■ DB2 (MainView for DB2)</td>
</tr>
<tr>
<td></td>
<td>■ NONE (MainView Primary Option Menu)</td>
</tr>
</tbody>
</table>

To display the same Primary Option Menu for multiple targets, list the TARGET statements followed by the PRIMAPPL statement; for example:

```
TARGET=DETROIT,TYPE=AO
TARGET=IMS22X,TYPE=IMS
TARGET=DB2P,TYPE=DB2
PRIMAPPL=DB2
```

**Note:** If PRIMAPPL is not specified, the Primary Option Menu lists all available products when the terminal session starts.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| PRIMMENU | By default, a Primary Option Menu is selected for the product or combination of products that you have installed for a particular target. If you have performance management products and the MainView AutoOPERATOR option for IMS or CICS installed, the default menu shows that option and the performance management products. If you want to limit access to MainView AutoOPERATOR, PRIMMENU can be specified as follows:  
\[ \text{TARGET=IMS22K,TYPE=IMS,PRIMMENU=IMS@MVMR} \]  
\[ \text{TARGET=CICSA1,TYPE=CICS,PRIMMENU=CICS@MR} \]  
The default for PRIMMENU is determined by the products that are installed. The values for PRIMMENU can be as follows: |
| IMS@MVA | *(MainView for IMS Online, MainView for DBCTL, and MainView AutoOPERATOR for IMS)* The Primary Option Menu for the MainView for IMS Online performance management products and MainView AutoOPERATOR for IMS (if installed). |
| IMS@MVMR | *(MainView for IMS Online and MainView for DBCTL)* The Primary Option Menu for the performance management components (Resource Analyzer, Resource Monitor, Workload Analyzer, and Workload Monitor). |
| DB2@MVMR | *(MainView for DB2)* The Primary Option Menu for all of the MainView for DB2 product services. |
| DB2@RXnn | *(MainView for DB2)* The Primary Option Menu for all of the MainView for DB2 product services, including an option to access RxD2, where nn represents the MainView for DB2 release number. |
| CICS@MVA | *(MainView for CICS and MainView AutoOPERATOR for CICS)* The Primary Option Menu for the MainView for CICS performance management services and MainView AutoOPERATOR for CICS (if installed). |
| CICS@MR | *(MainView for CICS)* The Primary Option Menu for the MainView for CICS performance management services. |
| ISPF | Indicate whether to initialize the product in an ISPF window (YES) or under TSO (NO). The default is YES. |
Define cross-system communication

Use the following procedure to set up CAS-to-CAS communication and BBI-SS PAS to BBI-SS PAS communication.

**Tip**
This task is optional for all products.

With cross-system communication, a single terminal session, using one or more MainView products, can monitor and manage multiple local or remote targets (the operating system itself in a sysplex or non-sysplex environment, and subsystems like CICS, IMS, and DB2).

A terminal session connects to:

- A CAS (if a CAS is available)

A CAS communicates with other CASs on other local and remote systems and allows direct communication between an individual terminal session and a PAS. Usually, one CAS runs per system image, but there is no limit to the number of remote CASs with which a single CAS can communicate.

### Note
If you see an error message advising you that BBPROF member BBICMDS should be at a more recent PTF level, check your data set concatenation.

This message indicates that your BBPROF data set contains a BBICMDS member that is not compatible with the current terminal session version. BBICMDS should not be in any data set that is concatenated in front of the distributed BBPROF data set.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBIRESP</td>
<td><em>(MainView AutoOPERATOR only)</em> LOCAL returns responses to BBI control commands that are issued from a terminal session to your terminal session journal and BBI-SS PAS journal. If BBIRESP is not used, BBI control command responses are written to the BBI-SS PAS journal only.</td>
</tr>
<tr>
<td>IMSRESP</td>
<td><em>(MainView AutoOPERATOR for IMS only)</em> LOCAL returns IMS command responses to your terminal session journal.</td>
</tr>
</tbody>
</table>
A BBI-SS PAS

BBI-SS PASs on local and remote systems are linked together to provide cross-system communication for an individual terminal session through a local BBI-SS PAS to any other BBI-SS PAS.

Depending on the products that are installed, the BBI-SS PAS might also connect to a CAS on that system image.

The Figure 2 on page 138 figure provides an overview of cross-system communication.
When CASs reside in the same sysplex, CAS-to-CAS communication is established automatically through the Cross-System Coupling Facility (XCF), as shown in the figure Figure 3 on page 139.
CAS to CAS communication in a sysplex environment

**Figure 3: Cross-system communication in a sysplex environment**

**PROD SYSPLEX**

**MVS image A (SYSA)**
- z/OS PASA
- MainView for z/OS
- MainView VistaPoint

**MVS image B (SYSB)**
- z/OS PASB
- MainView for z/OS
- MainView VistaPoint

**MVS image C (SYSC)**
- z/OS PASC
- MainView for z/OS
- MainView VistaPoint

**MVS image D (SYSD)**
- z/OS PASD
- MainView for z/OS
- MainView VistaPoint

**MVS image E (SYSE)**
- z/OS PASE
- MainView for z/OS
- MainView VistaPoint

Coupling facility / CTC
- VTAM or TCP/IP link
- VTAM, TCP/IP, or XCF link
- XCF
- XCF

Chapter 3 Manual customization 139
Planning for CAS-to-CAS communication

Before CASs can request information from each other through XCF, VTAM, or TCP/IP, you might need to define or modify local and remote CAS definitions for each CAS.

**Tip**

This part of the task does not apply to MainView FOCAL POINT.

If the local CAS is to use VTAM or TCP/IP to communicate with remote CASs, a CAS definition must exist for all of the CASs.

Each CAS definition is uniquely named and contains:

- Identification information about the CAS
- An XCF group name that is used by the CAS to establish communication with CASs that reside in the same sysplex
  The default XCF group name is BBGROUP. The XCF member name always defaults to the CAS name.
- A VTAM major node name
  This name is required if the CAS uses VTAM to communicate with another CAS.
- TCP/IP information (such as a host name, IP address, and a port number)
  This information is required if the CAS uses TCP/IP to communicate with another CAS.
- Parameter member suffixes to control cross-system communication in a shared DASD environment

**Note**

If you created VTAM definitions for a previous release, you do not need to perform this step unless you are running a previous release and a new release on the same system.

To prepare for defining CASs and cross-system communication links

1. Determine whether the CAS is running in a sysplex environment.

   In a sysplex environment, the Cross-System Coupling Facility (XCF) is used to communicate automatically with CASs that are running in the same sysplex. VTAM and TCP/IP definitions are not required for this environment. If a definition does exist, it will be used as a backup.

2. Determine with which systems you want the current CAS to communicate.
For more information, see CAS communication links on page 142 and CAS cross-system customization in a shared environment on page 142.

3 In SYS1.PARMLIB member IEASYSxx, find the system name of each system with which you want the current CAS to communicate.

4 If VTAM is being used to communicate between CASs that are running in different sysplexes, perform the following actions:

   a In SYS1.VTAMLST(CDRMxx), find the VTAM Cross-Domain Resource Manager (CDRM) name for each system that is located outside the sysplex.

   b If the current CAS will communicate with CASs that are located outside the sysplex, determine a VTAM major node name for each CAS on each system.

     **Note**
     In a shared DASD environment, you can define one major node name for all of your CASs if the minor node names are unique throughout the sysplex or multisystem environment.

   c Determine a VTAM minor node name for each CAS that is not in the sysplex.

   d Ensure that your user ID is authorized to access each system and modify the system’s VTAM configuration.

5 Use the system name, when possible, as the name for each CAS.

If you cannot use the system name, determine a unique CAS name for each system.

     **Note**
     If you are running products at different release levels on the same system (for example, using a production system to run test products at a newer level than products from a previous level), a unique CAS name is required.

6 In SYS1.PARMLIB member (COUPLxx), find the sysplex name for each system that is running in sysplex mode.

7 *(CMF MONITOR only)* If you require XCF communication links and you cannot use the default XCF group name of BBGROUP, determine a valid group name for XCF communication.

**Related Information**

- “Customizing an XCF group name” on page 160
CAS communication links

CAS-to-CAS communication allows data to be shipped back and forth between systems and users to open windows and obtain data from both systems.

CAS-to-CAS communication requires customization of CAS definitions for each system. The communication link between two CASs can be provided by XCF, VTAM, or TCP/IP:

- **XCF link**
  CASs that are running on different system images within the same sysplex can communicate directly through an XCF link. An XCF communication link is established automatically between two CASs when they have the same XCF group name in the CAS definition (see Customizing an XCF group name on page 160. In an XCF configuration, a VTAM link is used only for backup when an XCF link is not available.

- **VTAM link**
  CASs that are not in the same sysplex or that are in different XCF groups can communicate through a VTAM link, as described in Define CAS-to-CAS communication on page 153.

- **TCP/IP link**
  CASs can optionally use the TCP/IP protocol for network communication. For more information, see the discussion of TCP/IP communication in the MainView Administration Guide.

**Tip**

If you have some CASs to which you want to restrict access, you can:

- Define a security definition by using the SERDEF view, as described in the MainView Security Guide
- Define a separate parameter library for the CAS that you want to protect

**Related Information**

- "Customizing an XCF group name" on page 160
- "Define CAS-to-CAS communication" on page 153

CAS cross-system customization in a shared environment

If the systems are running in a shared DASD environment, and the CASs are accessing a shared parameter library, the following conditions apply:
VTAM application major nodes and CDRM definitions that all reside in the same shared SYS1. VTAMLST library can be defined in a single member so that only one system configuration list needs to be updated.

However, if only one major node member is created, the minor node name that is defined for each CAS must be unique throughout your sysplex or multisystem environment.

If SYS1.VTAMLST is not shared, a separate VTAM major node must be created and each system configuration list must be updated.

CAS definitions do not have to be defined repeatedly in each CAS.

You need only one CAS definition parameter library member containing all definitions for all CASs that communicate with each other. Each CAS points to this library with the BBIPARM DD statement in its startup procedure. You can define one set of definitions for one CAS and use the INSTall command from the CASDEF view for all of the other CASs.

**Note**

If you have certain CASs that you do not want to communicate (such as the test-system CAS and the production-system CAS), you must have separate BBPARM libraries for those CASs.

**Related Information**

- “Defining a VTAM major node for the CAS” on page 153
- “Creating CAS cross-domain resource manager definitions” on page 156
- “Adding CAS definitions for the first time” on page 158

**CAS cross-system communication requirements**

For communication between CASs in the same sysplex, the following conditions apply:

- The default XCF group name of BBGROUP automatically provides an XCF communication link between CASs in the same sysplex.
  
  As long as the default XCF group name is not changed, CASs in different system images can communicate directly through the XCF link.

- No VTAM definition is needed.
  
  If a VTAM definition does exist, it is used for backup when the XCF link is not available.
For communication between CASs in different sysplexes when using VTAM, the following requirements apply:

- A VTAM application major node definition is required to identify the CAS to VTAM.
  When VTAM knows about a CAS, the CAS can issue instructions to VTAM to establish communication links with other CASs that other VTAMs know about.

- VTAM CDRM definitions are required to identify the other CASs that are known to other VTAMs.
  VTAM must know about all the cross-system CASs in order to establish connections between them.

- CAS definitions must exist on each system for CAS communication through VTAM, as described in Define CAS-to-CAS communication on page 153.

For communication between CASs in different sysplexes when using TCP/IP, see the discussion of TCP/IP communication in the MainView Administration Guide.

**Related Information**

- “Define CAS-to-CAS communication” on page 153

---

**Planning for BBI-SS PAS-to-BBI-SS PAS communication**

You can prepare for defining BBI-SS PASs and cross-system communication links.
Tip
This task applies for the following products:

- MainView FOCAL POINT
- MainView AutoOPERATOR
- MainView for CICS
- MainView for DB2
- MainView for DBCTL
- MainView for IMS Online
- MainView for IMS Offline
- MainView for WebSphere MQ

To prepare for defining BBI-SS PASs and cross-system communication links

1. Determine the systems that you want to communicate with the current PAS.

2. Determine a VTAM major node name and minor node name for each PAS on each system image.

3. Ensure that your user ID is authorized to access each system and modify the system’s VTAM configuration.

Adding a BBI-SS PAS

Use the following procedure to add a new BBI-SS PAS.

You can add a new BBI-SS PAS so that more than the allowable number of target systems can be managed.

You can also add a new BBI-SS PAS so that:

- A different target system can be managed
  For example, one BBI-SS PAS could be connected to a test system and another PAS could be connected to a production system.

- Unique specifications can be defined for a BBI-SS PAS (for example, for security requirements)

- Terminal sessions can be processed by one system image, and the target regions can be processed by another system image
One BBI-SS PAS can communicate with another BBI-SS PAS

**To add a new BBI-SS PAS**

1. Define the BBI-SS PAS and its associated target in BBPARM members BBIJNT00 and BBISSPx xx.

2. Define connections between the new BBI-SS PAS and any existing BBI-SS PASs that you want to communicate.

3. If the new BBI-SS PAS is connected to an active BBI-SS PAS, restart the active BBI-SS PAS to activate the connection.

**Related Information**

- “Defining BBI-SS PAS suffixes and target system parameters” on page 105
- “Define BBI-SS PAS-to-BBI-SS PAS communication” on page 147

**BBI-SS PAS communication links**

Subsystem communication allows a user from a single terminal session to manage one or more target systems that belong to multiple BBI-SS PASs.

The BBI-SS PAS processes communication between one or more terminal sessions and one or more targets when:

- A terminal session is connected to one BBI-SS PAS
- Multiple terminal sessions are connected to a single BBI-SS PAS
- Multiple target systems are assigned to a single BBI-SS PAS
- A terminal session can access data from multiple targets through its associated BBI-SS PASs
- A request from a terminal session can be automatically routed by its associated BBI-SS PAS to another BBI-SS PAS

*Define BBI-SS PAS-to-BBI-SS PAS communication on page 147* describes the VTAM parameters that you must define for single-domain and cross-domain communication. VTAM parameters are required by the VTAM system programmer to correctly define the MainView communication environment. For additional information about the VTAM parameters, see the IBM publication *Advanced Communications Function for VTAM - Installation and Resource Definition.*
All BBI-SS PAS-to-BBI-SS PAS communication is routed through VTAM, as shown in Figure 4 on page 147.

**Figure 4: BBI-SS PAS-to-PAS communication**

---

**Related Information**

- “Define BBI-SS PAS-to-BBI-SS PAS communication” on page 147

---

**Define BBI-SS PAS-to-BBI-SS PAS communication**

Use the following procedure to manually define subsystem communication for BBI-SS PASs in a single domain, and expand the definition to provide for communication with a BBI-SS PAS in another domain.
This task applies to the following products:

- MainView FOCAL POINT
- MainView AutoOPERATOR
- MainView for CICS
- MainView for DB2
- MainView for DBCTL
- MainView for IMSOnline
- MainView for IMSOffline
- MainView for WebSphere MQ

**Define a VTAM major node for the BBI-SS PAS**

Using example BBI-SS PASs named BBIA and BBIB, this topic illustrates how to create BBI-SS PAS definitions for cross-system communication.

Assume that BBIA monitors one target system, and BBIB monitors another target system. To establish communication between these two subsystems, the following requirements apply:

- Both subsystems must be defined as BBI subsystems.
- A VTAM communication link must be defined between them.

**Note**

Review the MAXDATA parameter in BBPARM member BBIXSP00 for compatibility with the VTAM MAXDATA parameter. In the network channel program's (NCP's) PCCU macro for Multisystem Networking Facility (MSNF) links, the MAXDATA parameter specifies the maximum-size buffer for VTAM. In BBIXSP00, MAXDATA specifies the maximum-size buffer that BBI passes to VTAM. However, VTAM adds a transmission header to the BBI buffer. Therefore, the BBI MAXDATA value must be 50 bytes less than the VTAM MAXDATA value.

Shared BBPARM members BBINOD00 and BBIJNT00 define subsystem names and their associated targets. BBINOD00 defines the BBI-SS PASs for subsystem communication, and BBIJNT00 defines targets and their associated BBI-SS PAS. BBINOD00 and BBIJNT00 support substituting system variables.
Example

The following table shows the parameters you would use in BBINOD00 and BBIJNT00 to define subsystems BBIA and BBIB in the node table:

<table>
<thead>
<tr>
<th>BBPARM member</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBINOD00</td>
<td>SUBSYS=BBIA,APPLID=$BBIA&lt;br&gt; SUBSYS=BBIB,APPLID=$BBIB&lt;br&gt; [QWKSTART=YES</td>
</tr>
<tr>
<td></td>
<td>■ BBIA and BBIB are the subsystem IDs that the startup JCL specifies.</td>
</tr>
<tr>
<td></td>
<td>■ $BBIA and $BBIB are the same names that the VTAM application node</td>
</tr>
<tr>
<td></td>
<td>definitions use.</td>
</tr>
<tr>
<td></td>
<td>■ QWKSTART controls the timeout value that is used for starting</td>
</tr>
<tr>
<td></td>
<td>communication links during PAS initialization. YES sets the timeout</td>
</tr>
<tr>
<td></td>
<td>value to 1 second. No (the default) sets the timeout value to 15 seconds.</td>
</tr>
<tr>
<td>BBIJNT00</td>
<td>TARGET=CICSA,TYPE=CICS,SUBSYS=BBIA&lt;br&gt; TARGET=CICSB,TYPE=CICS,SUBSYS=BBIB</td>
</tr>
</tbody>
</table>

Related Information

■ “System variables substitution in parameter library members” on page 233

Create cross-domain BBI-SS PAS resource manager definitions

The subsystem communication path is defined through VTAM. The path can be single domain or cross domain.

The single-domain path is defined as a VTAM application node. The cross-domain path is additionally defined as a VTAM cross-domain resource.
Single-domain definitions

If BBI-SS PASs BBIA and BBIB are in the same domain, as shown in Figure 5 on page 150, the following VTAM application nodes must be defined as shown:

- **SYS1.VTAMLST(BBIAPPLA)**
  
  ```
  TYPE=APPL
  VTAMLST $BBIA
  AUTH=(ACQ,NOSO,VPACE)
  ```

- **SYS1.VTAMLST(BBIAPPLB)**
  
  ```
  TYPE=APPL
  VTAMLST $BBIB
  AUTH=(ACQ,NOSO,VPACE)
  ```

**Note**
The $BBIA and $BBIB nodes can be defined within one major node, such as BBIAPPLA. You can choose names that conform to your site standards, but they must be the same as the APPLID parameters that are specified in BBPARM member BBINOD00 for the respective BBI-SS PASs. This example uses the sample member names (BBIAPPLA, BBIAPPLB, BBIXSA, BBIXSB) that are distributed by BMC in the BBSAMP data set.

---

**Figure 5: BBI-SS PAS-to-BBI-SS PAS single-domain communication**

**Development System (Domain A)**

```
BBI-SS PAS (BBIA)  VTAMLST A  BBI-SS PAS (BBIB)
```

**Target System**

Dashed lines represent one or more possible terminal session (TS) connections.
If BBI-SS PASs BBIA and BBIB are cross domain, VTAM cross-domain resources must be defined. For more information, see Figure 6 on page 152.

**Cross-domain A definitions**

Define an application node (BBIAPPLA) and a cross-domain resource (BBIXSA) in domain A as follows:

- **SYS1.VTAMLST(BBIAPPLA)**

  ```
  $BBIA    APPL    AUTH=(ACQ,NOTSO,VPACE)
  ```

- **SYS1.VTAMLST(BBIXSA)**

  ```
  $BBIB    CDRSC   CDRM=cdrmname,ISTATUS=ACTIVE
  ```

$BBIB is the cross-domain node for the BBI-SS PAS that is used in the VTAM application node definition in domain B.

*cdrmname* is the name of the CDRM major node that is defined for domain B.

**Cross-domain B definitions**

Define an application node (BBIAPPLB) and a cross-domain resource (BBIXSB) in domain B as follows:

- **SYS1.VTAMLST(BBIAPPLB)**

  ```
  $BBIB    APPL    AUTH=(ACQ,NOTSO,VPACE)
  ```
- SYS1.VTAMLST(BBXSB)

| VBUILD | TYPE=CDRSC | CDRM=cdrmname, ISTATUS=ACTIVE |

$BBIA is the cross-domain node for the BBI-SS PAS that is used in the VTAM application node definition in domain A.

cdrmname is the name of the CDRM major node that is defined for domain A.

Figure 6: BBI-SS PAS-to-BBI-SS PAS cross-domain communication

Development System (Domain A)

Development System (Domain B)
Define CAS-to-CAS communication

To establish CAS-to-CAS communication, repeat the following steps on each system:

- Defining a VTAM major node for the CAS on page 153
- Defining a default VTAM mode table entry on page 154
- Creating CAS cross-domain resource manager definitions on page 156
- Activating the VTAM major nodes on page 157

You can skip this section if:

- All CASs are running in the same sysplex
  
  CASs in the same sysplex communicate through an XCF link as long as they all have the same XCF group name. The default is BBGROUP.

- The current CAS does not need to communicate with another CAS on a different sysplex through a VTAM link

Defining a VTAM major node for the CAS

Use the following procedure to define a VTAM major node for the CAS.

To define a VTAM major node for the CAS

1. Create a new member in SYS1.VTAMLST.

   The member name that you use becomes the major node name for the CAS.

2. In the SYS1.VTAMLST member, add the following APPL statement:

   ```
   VBUILD TYPE=APPL
   name  APPL Auth = (ACQ),PARSESS=YES, MODETA = table,
   DLOGMOD = BBCDEF
   ```

   The variables are as follows:

   - `name` is the minor node name for this CAS (the name by which VTAM identifies the CAS on the system). The minor node name cannot exceed eight characters.

   In a shared DASD environment, you can define one SYS1.VTAMLST member that contains an APPL statement for each CAS; however, only one VBUILD statement should be defined, and each name that you define to an APPL statement must be unique throughout your sysplex or multisystem environment.
- **table** is the name of the mode table that governs LU-to-LU conversations.

- BBCDEF is the name of the mode table entry that governs LU-to-LU conversations.

**Note**

Depending on the limitations of the network channel programs (NCPs) in your network (if you have these limitations), you might need to create a mode table entry for your MainView products.

To define one SYS1.VTAMLST member that contains an APPL statement for each CAS in a shared DASD environment, you can create a single major node member and add APPL statements. An example follows:

```plaintext
VBUILD TYPE=APPL
CASA APPL AUTH=(ACQ),PARSESS=YES, MODETAB=BBMODET, DLOGMOD=BBCDEF
CASB APPL AUTH=(ACQ),PARSESS=YES, MODETAB=BBMODET, DLOGMOD=BBCDEF
CASC APPL AUTH=(ACQ),PARSESS=YES, MODETAB=BBMODET, DLOGMOD=BBCDEF
```

**Related Information**

- “Defining a default VTAM mode table entry” on page 154

---

**Defining a default VTAM mode table entry**

You should define a default mode table entry to control the request unit size (RUSIZES) parameter that VTAM uses to send MainView application messages to NCPs in your network.

If you do not create a default entry, VTAM could select a RUSIZES number that is too small for MainView product messages, which could result in message truncation.
To create a default mode table entry

1 Use the example in Figure 7 on page 155 to define an entry:

Figure 7: Default mode table entry for the RUSIZES parameter

```
TITLE 'modetab - MODE Table - Entries'
modetab MODETAB,                     X
   MODEENT LOGMODE=BBCDEF,             X
     FMPROF=X'13',                     X
     TPROF=X'07',                      X
     PRIPROT=X'B0',                    X
     SECProt=X'B0',                    X
     COMProt=X'50B1',                  X
     SSNDPAC=X'00',                    X
     SRCVPAc=X'00',                    X
     RUSIZES=X'F8F8',                  X
     PSNDPAC=X'00',                    X
     PSERVIC=X'06020000000000000000002300', X
     ENCR=X'00'
   MODEEND,                           X
END,                                 X
```

`modetab` is a mode table name that you supply. If you already have an entry with these defaults, use that entry as the `modetab` value in Defining a VTAM major node for the CAS on page 153.

2 Use the sample JCL in Figure 8 on page 156 to assemble the mode table source and link-edit it into SYS1.VTAMLIB on all systems where cross-system communication is to be enabled.
Note

- You can choose any name that you want for the load module. The name that you choose becomes the actual name of the mode table.

- Before you can assemble the mode table source, you must have access to the macro library that is used to assemble VTAM applications.

- Your system might have a user version of the VTAM definition library concatenated to the SYS1.VTAMLIB library, or it might use a different library altogether.

- If you have a shared DASD environment where all systems share the same SYS1.VTAMLIB, you need to perform this step only once.

**Figure 8: JCL for assembling a mode table entry**

```plaintext
//ASM     EXEC PGM=ASMBLR,PARM='OBJECT,NODECK'
//SYSLIB   DD  DSN=SYS1.SISTMAC1,DISP=SHR
//        DD  DSN=SYS1.MACLIB,DISP=SHR
//SYSUT1   DD  DSN=&&SYSUT1,UNIT=VIO,SPACE=(1700,(600,100))
//SYSTERM  DD  SYSOUT=* 
//SYSPRINT DD  SYSOUT=* 
//SYSLIN   DD  DSN=&&OBJSET,UNIT=VIO,SPACE=(80,(200,50)),
//           DISP=(MOD,PASS)
//SYSSIN   DD  DSN=datasetname(membername),DISP=SHR
//*
//LINK    EXEC PGM=IEWL,PARM=(XREF,LET,LIST,NCAL,REUS),
//        COND=(8,LT,ASM)
//SYSLIN   DD  DSN=&&OBJSET,DISP=(OLD,DELETE)
//        DD  DDNAME=SYSIN
//SYSLMOD  DD  DSNAME=SYS1.VTAMLIB(membername),DISP=SHR
//SYSUT1   DD  DSNAME=&&SYSUT1,UNIT=VIO,
//           SPACE=(1024,(50,20))
//SYSPRINT DD  SYSOUT=* 
//*
```

**Related Information**

- "Defining a VTAM major node for the CAS" on page 153

**Creating CAS cross-domain resource manager definitions**

Use the following procedure to create a CDRM definition for each cross-system CAS that you want the current CAS to communicate with.

1. Create a new member in SYS1.VTAMLST.
2 In the SYS1.VTAMLST member, insert one CDRSC statement to define each cross-
   system CAS that you want to communicate with:

```
VBUILD TYPE=CDRSC
  name  CDRSC  CDRM=cdrm
```

The variables are as follows:

- **name** is the minor node name for each cross-system CAS (as defined in Define
  a VTAM major node for the BBI-SS PAS on page 148).

- **cdrm** is the VTAM name for the remote system image.

*Note*
You should have a naming convention for each CDRM definition member.

3 Update the VTAM configuration list by performing the following:

Take the new SYS1.VTAMLST members that were created in the preceding steps
and in Defining a VTAM major node for the CAS on page 153 and add them to
the VTAM configuration list in SYS1.VTAMLST(ATCCONxx).

**ATCCONxx** lists the major nodes that VTAM activates upon initialization.

*Note*
To find ATCCONxx, look in the VTAM configuration's start list,
SYS1.VTAMLST(ATCSTRxx).

**Related Information**

- "Defining a default VTAM mode table entry" on page 154
- "Defining a VTAM major node for the CAS" on page 153

---

**Activating the VTAM major nodes**

To activate major node definitions for each updated VTAM configuration list, you
 can either restart VTAM on each system or manually activate the VTAM majo nodes.
 Use the following procedure to manually activate the nodes.

*Note*
Create cross-domain BBI-SS PAS resource manager definitions on page 149 explains
how to update the VTAM configuration lists.
To manually activate the definitions

1. Issue the following VTAM command on each VTAM system:

   VARY NET,ACT,ID=\textit{nodename}

   \textit{nodename} is the name of the major node member that was created in Defining a VTAM major node for the CAS on page 153.

2. Verify that the definition is active by issuing the following VTAM command:

   D NET,ID=\textit{nodename},E

3. Issue the following VTAM command on each VTAM system to activate new CDRM definitions:

   VARY NET,ACT,ID=\textit{cdrmname}

   \textit{cdrmname} is the name of the CDRM definition member that was created in Creating CAS cross-domain resource manager definitions on page 156.

4. Verify that the definition is active by issuing the following VTAM command:

   D NET,ID=\textit{cdrmname},E

---

**Related Information**

- “Creating CAS cross-domain resource manager definitions” on page 156
- “Defining a VTAM major node for the CAS” on page 153

---

Adding CAS definitions for the first time

Use the following procedure to add new CAS definitions.

To maintain existing CAS definitions, see the \textit{MainView Administration Guide}.

To enable cross-system communication between CASs, perform the following steps first on the local system, and then on each of the systems with which you want to communicate.

**To enable cross-system communication between CASs**

1. Start the CAS by issuing the following command:
Start a MainView terminal session.

On the MainView Selection Menu, select Option P, Plex Manager.

On the COMMAND line, type CASDEF and press Enter to display the CASDEF view.

- When you display CASDEF on a system for the first time and no CAS definitions exist, a default definition is shown with the following information in the Description field:
  
  Use CASDEF to update

  In addition, the VTAM Appl Name field has a value of *NONE*.

- If you are migrating from a previous product release, those definitions are shown when you display CASDEF. The same description appears for all definitions.

On the COMMAND line, type EDIT and press Enter to obtain an edit lock.

On the COMMAND line, type ADD casname and press Enter, where casname is the name of the system image or some other unique name.

- When defining a CAS, you should use the same name as the system name (because only one CAS should run per system image, except in test situations where a production CAS is already running on the same system).

  You can use the Add New CAS System Definition dialog to define unique cross-system communication parameters to each CAS.

In the Description field, type a description of the CAS.

- The CAS System Name and Description fields are required for a CAS definition.

- If you want to use TCP/IP for CAS-to-CAS communication, complete the IP Host Name, Port Number, and TCP Jobname fields.

When you are finished with the CAS definition, press or type END to return to the CASDEF view.
On the **COMMAND** line, type **SAVE** to save the information in the BBPARM CAS definition member.

For each cross-system CAS with which you want to establish communications, repeat Step 6 on page 159 through Step 9 on page 160.

Stop the CAS by issuing the following command:

```
P casname
```

To implement the new cross-system CAS definitions, start the CAS again by issuing the following command:

```
S casname
```

The definitions on the local CAS are now complete.

To complete the CAS definitions on the remote systems, perform one of the following tasks, depending on your CAS configuration:

- If each CAS has a separate and unique BBPARM library that it does not share with any other CAS, perform the following steps:
  
  a. Log off the current system.
  
  b. Log on to each remote system.
  
  c. Repeat Step 1 on page 158 through Step 12 on page 160.

- If the CASs operate in a shared DASD and shared BBPARM environment, perform the following steps:
  
  a. Log off the current system.
  
  b. Log on to each remote system.
  
  c. Repeat Step 1 on page 158 through Step 6 on page 159.
  
  d. On the **COMMAND** line, enter **INStall casname** to update the cross-system communication parameters for the local CAS.
  
  e. Stop the CAS (**P casname**) and restart it (**S casname**) to initialize the CAS with the new cross-system CAS definitions.

**Customizing an XCF group name**

Use the following procedure to customize the XCF group name for a CAS.
MainView products use a default XCF group name of BBGROUP, and each CAS in
the group uses a member name that is the same as the CAS name. Both of these
values are defined when you create a CAS definition for each CAS in the sysplex.

Skip this section unless:

■ You cannot use the default XCF group name of BBGROUP
■ You are running multiple CASs on the same system

To customize the XCF group name for a CAS

1 Start the CAS by issuing the following command:

   \texttt{S casname}

2 In the XCF Group Name field on the Add New CAS System Definition or Change
   CAS System Definition dialogs, define the same group name for all CASs that you
   want to join an XCF group.

3 For each CAS definition that you add or change, issue the SAVE command from
   the CASDEF view.

4 Exit the MainView window interface.

5 Issue the following command to stop the CAS:

   \texttt{P casname}

6 Issue the following command to restart the CAS:

   \texttt{S casname}

Troubleshooting CAS cross-system communication

If you have problems activating cross-system communication, see the following
table:

<table>
<thead>
<tr>
<th>If this problem occurs</th>
<th>Perform this action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message BBCSJ015W appears in the SYSLOG with sense code 800A0000.</td>
<td>See Creating a mode table entry to correct a sense code of 800A0000 on page 162</td>
</tr>
<tr>
<td>Message BBCSJ008W appears in the SYSLOG with return code 5A.</td>
<td>Ensure that you activated the APPLIDs for the CASs. See Activating the VTAM major nodes on page 157</td>
</tr>
</tbody>
</table>
If this problem occurs | Perform this action
---|---
Message BBCSJ015W appears in the SYSLOG with sense code 0857.xxxx. | Ensure that your cross-domain links are set up correctly. See Creating CAS cross-domain resource manager definitions on page 156

### Creating a mode table entry to correct a sense code of 800A0000

Use the following procedure to prevent data truncation by accommodating the buffer size of your network channel programs (NCPs).

If you receive a message in the SYSLOG that contains sense code 800A0000, at least one of the network channel programs (NCPs) in your network has a message buffer that is smaller than the MainView default buffer of 3840 bytes. As a result, some of the data that is traveling to that system is truncated. To prevent this situation, you must create an entry in the mode table to accommodate the NCP buffer size.

Perform the following steps on all systems that are connected to the NCP:

**To create a mode table entry to correct a sense code of 800A0000**

1. Add the entry shown in Figure 9 on page 162 to the mode table that you created in Defining a default VTAM mode table entry on page 154.

**Figure 9: VTAM mode table entry for NCP**

```
BBMODEA
TITLE 'BBMODEA - MODE Table - Entries'
BBMODEA
MODETAB ,
MODEENT LOGMODE=BBDEF,
FMPROF=X'13',
TSPROF=X'07',
PRIPROT=X'B0',
SECPROT=X'B0',
COMPROT=X'50B1',
SSNDPAC=X'00',
SRCVPAC=X'00',
RUSIZES=X'F8F8',
PSNDPAC=X'00',
PSERVIC=X'060200000000000000002300',
ENCR=X'00'
MODEEND ,
END ,

MODEENT LOGMODE=logmode,
FMPROF=X'13',
TSPROF=X'07',
PRIPROT=X'B0',
SECPROT=X'B0',
COMPROT=X'50B1',
SSNDPAC=X'00',
SRCVPAC=X'00',
RUSIZES=X'yyyy',
PSNDPAC=X'00'.
END ,
```

162  MainView Customization Reference
The variables are as follows:

- **logmode** is the name of the entry that you are adding.
- **RUSIZES=X'yyy'** is a hexadecimal number that is as close to 3840 bytes as possible.

2. Check the MAXDATA parameter in the NCPGEN for the NCP that is used to communicate with the failing system.

To accommodate as much data as possible, use the following table to find the number that is closest to this MAXDATA value. The number that you choose must be at least 15 bytes smaller than the MAXDATA value.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>256</td>
<td>288</td>
<td>320</td>
<td>352</td>
<td>384</td>
<td>416</td>
<td>448</td>
<td>480</td>
</tr>
<tr>
<td>6</td>
<td>512</td>
<td>576</td>
<td>640</td>
<td>704</td>
<td>768</td>
<td>832</td>
<td>896</td>
<td>960</td>
</tr>
<tr>
<td>7</td>
<td>1024</td>
<td>1152</td>
<td>1280</td>
<td>1408</td>
<td>1536</td>
<td>1664</td>
<td>1792</td>
<td>1920</td>
</tr>
<tr>
<td>8</td>
<td>2048</td>
<td>2304</td>
<td>2560</td>
<td>2816</td>
<td>3072</td>
<td>3328</td>
<td>3584</td>
<td>3840</td>
</tr>
<tr>
<td>9</td>
<td>4096</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For example, if you choose to specify 3072, determine its hexadecimal representation by finding the number's value in the A columns, which is 12 (C). Then, find its value in the B column, which is 8. So, you would specify C8C8 on the RUSIZES parameter.

**Note**

You need to repeat the digits because the RUSIZES parameter contains two bytes. The first byte (AB) reflects the size of the incoming NCP buffer; the last byte (AB) reflects the size of the outgoing buffer. Usually, these two numbers are the same.

3. Assemble the mode table source and link-edit it into SYS1.VTAMLIB on all CAS systems where cross-system communication is enabled.

You can use the sample JCL in Defining a default VTAM mode table entry on page 154 for this step.
4 In the member that contains the system VTAM definition, add the MODETAB operand to the APPL statement as shown:

```vbuild
VBUILD TYPE=APPL
  name   APPL AUTH=(ACQ),PARSESS=YES,MODETAB=module,
  DLOGMOD=BBCDEF
```

The variables are as follows:

- `name` is the name that you assigned to the system image for the CAS.
- `module` is the name of the load module that you created in Step 1 on page 162 (BBMODEA in Figure 9 on page 162).

5 Enter `VARY NET,INACT,ID=name` to deactivate the major node member.

   **Note**
   This command causes a service interruption for any active terminal sessions.

6 Enter `VARY NET,ACT,ID=name` to reactivate the member and make the MODETAB parameter accessible to VTAM.

**Related Information**

- "Defining a default VTAM mode table entry" on page 154

---

**Specifying product initialization within a BBI-SS PAS**

Use member BBISSPxx to enable your products for initialization within the BBI-SS PAS, and to review other optional parameters.
Tip
This task is required for the following products but does not apply to other products:

- MainView FOCAL POINT
- MainView AutoOPERATOR
- MainView for CICS
- MainView for DB2
- MainView for DBCTL
- MainView for IMSOnline
- MainView for IMSOffline
- MainView for WebSphere MQ

1 In BBPARM member BBISSPxx, specify the following statement for each product (placing each statement on a separate line):

```
PRODUCT= prd
```

where `prd` can be

<table>
<thead>
<tr>
<th>Prd Code</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAO</td>
<td>MainView AutoOPERATOR</td>
</tr>
<tr>
<td>CMR</td>
<td>MainView for CICS</td>
</tr>
<tr>
<td>DMR</td>
<td>MainView for DB2</td>
</tr>
<tr>
<td>MFP</td>
<td>MainView FOCAL POINT</td>
</tr>
<tr>
<td>MVDBC</td>
<td>MainView for DBCTL</td>
</tr>
<tr>
<td>MVIMS</td>
<td>MainView for IMSOnline</td>
</tr>
<tr>
<td>MVMQS</td>
<td>MainView for WebSphere MQ</td>
</tr>
<tr>
<td>MVVP</td>
<td>MainView VistaPoint</td>
</tr>
</tbody>
</table>

Related Information

- "BBI-SS PAS library members“ on page 199
Enabling Automatic Restart Manager (ARM) support

Use member BBISSPxx to enable support for the Automatic Restart Manager (ARM) if you want it to automatically restart BBI-SS PASs that have abended.

Tip
This task is optional for MainView AutoOPERATOR, and it does not apply for other products.

ARM provides support for automatically restarting batch jobs and started tasks after unexpected termination. This function is available in either single sysplex environments or multisystem sysplex environments that use coupled data sets.

To enable ARM support

1 In BBPARM member BBISSPxx, specify the following statement:

   `ARMPOLICY=[NO | YES | xxxxxxxx ]`

   Variables are as follows:
   - **NO** indicates that the BBI-SS PAS will not use the ARM facility.
   - **YES** indicates that the BBI-SS PAS will use the default ARM restart policy. The default policy is for ARM to 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 inactive.
   - **xxxxxxx** specifies the one- to eight-character name of a user-created policy for controlling BBI-SS PAS restarts.

For more information about implementing ARM support for MainView AutoOPERATOR, see the MainView AutoOPERATOR Customization Guide.

Adding the CAS ID to the BBI-SS PAS

Use your copy of member BBISSPxx to identify the CAS to the BBI-SS PAS.
Tip
This task is required for the following products but does not apply to other products:

- MainView AutoOPERATOR
- MainView for CICS
- MainView for DB2
- MainView for DBCTL
- MainView for IMS Online
- MainView for IMS Offline
- MainView for WebSphere MQ

To identify the CAS

1. In BBPARM member BBISSP xx, add a CASID statement.

2. Specify the four-character subsystem ID of the CAS to which this BBI-SS PAS should connect, as follows:

   \[ \text{CASID} = \text{ssid} \]

   The CAS subsystem ID is defined with the PROC statement in the CAS started task procedure.

Related Information

- “CAS library member” on page 173

Adding extended function data sets to the BBI-SS PAS startup procedure

Use your copy of member SSJCL to add extended function data sets to the BBI-SS PAS started task procedure.
Tip
This task is optional for the following products and does not apply to other products:

- MainView AutoOPERATOR
- MainView for CICS
- MainView for DB2
- MainView for DBCTL
- MainView for IMS Online
- MainView for IMS Offline
- MainView for WebSphere MQ

To add extended function data sets to the BBI-SS PAS startup procedure

1. Edit BBSAMP member SSJCL by adding the following statements:
   - //HISTDSnn DD for each historical data set
   - //BBACTDEF for the action and view table data set
   - //BBVDEF for the default views data set

   The following example shows DD statements that add three historical data sets for a BBI-SS PAS running on SYSA. Separate DD statements initialize the action and view table data set, and the default views data set:

   ```
   //HISTDS00 DD DSN=hlq.SYSA.HISTDS00,DISP=SHR
   //HISTDS01 DD DSN=hlq.SYSA.HISTDS01,DISP=SHR
   //HISTDS02 DD DSN=hlq.SYSA.HISTDS02,DISP=SHR
   //BBACTDEF DD DSN=hlq.BBACTDEF,DISP=SHR
   //BBVDEF DD DSN=hlq.BBVDEF,DISP=SHR
   //PARMFILE DD DSN=hlq.PARMFILE,DISP=SHR
   ```

   `hlq` is the high-level qualifier that was used to load product libraries to DASD.

2. After you install historical data sets, issue the MainView DSLIST command to view their status.

   The DSLIST view shows:

   - Name and current status of each data set
Beginning and ending time of the most recent recording interval
You can allocate another historical data set by issuing the ADD command from DLIST.

**Related Information**

- “Creating a BBI-SS PAS startup procedure” on page 121

---

**Update command processor table**

Some BMC programs and commands are either issued as a TSO/E command processor, or called as a program in the TSO/E environment. Use the following procedure to verify that your security environment allows command processors to execute under ISPF.

**Tip**
This task is optional for all products.

These programs and commands might be invoked explicitly by options on the MainView Selection Menu, or implicitly by the MainView user address space (UAS) or product address space (PAS).

Some sites have a security package (such as IBM RACF, Computer Associates CA ACF2 or CA TOP SECRET, or IBM PCF) that defines an authorized command processor table to restrict TSO command processor execution. If your security package defines this type of table, you might need to add the following programs and commands to the table:

- AOEXEC
- BALCMSG
- BBM3API
- BBM9TC21
- BBM9TC22
- BBM9TC24
- BBVJSETP
- EMTMPW
SMLOAD

TSLOAD

To add programs and commands to the command processor table, see the documentation for the security package that is used at your site.

Note

If any of these programs or commands are invoked by MainView and are not in the TSO command processor table, the following message is issued:

IKJ56500I COMMAND xxxxxxxx NOT FOUND

xxxxxxxx is the program or command that was invoked. If the program or command was invoked in background mode (by the PAS, for example), the IKJ56500I message appears in the SYSTSPRT output.

If you are invoking the MainView terminal session in an ISPF environment, or you are calling TSLOAD as a command processor, you must verify that your security environment allows command processors to execute under ISPF. Otherwise, the message SMLOAD NOT FOUND is issued.

To authorize the terminal session

1 Determine whether the ISPF TSO Command Table (ISPTCM) allows TSLOAD and SMLOAD to be issued as a TSO command processor.

   If not, see the IBM publication Interactive System Productivity (ISPF) and ISPF/Program Development Facility for MVS Installation and Customization, and add TSLOAD and SMLOAD to the ISPTCM table.

   Note

   The ISPTCM entry for TSLOAD and SMLOAD must specify a FLAG parameter with the X'40' and X'02' bits set on (that is, ISPTCM must be set to X'42').

2 If the Program Control Facility (PCF) is installed on your system, determine whether it allows the SMLOAD and TSLOAD commands to be issued as a TSO command processor.

   If not, add SMLOAD and TSLOAD to the PCF-authorized Program Table, or set the user's authorization level to zero.

3 If you have RACF or other appropriate security installed, determine whether it allows the SMLOAD command to be issued as a TSO command processor.

   If not, change the ACF2 command list to do so.
If you have checked all of the preceding requirements and you still get the `SMLOAD NOT FOUND` message, turn off all security checking and run the terminal session. If the terminal session runs, the problem is related to one of the site's security programs. If the problem persists, contact BMC Customer Support.

### Specifying product initialization within the z/OS PAS

Use member BBATSP00 to control product initialization within the z/OS PAS.

#### Tip

This task is optional for the following products and does not apply to other products:

- CMF MONITOR
- MainView for z/OS
- MainView for UNIX System Services (USS)
- MainView SYSPROG Services
- MainView VistaPoint

All of the products are initialized by default. However, you can suppress the initialization of one or more products by using BBATSP00.

#### To control product initialization within the z/OS PAS

1. Copy `hlq.BBSAMP` member BBAMVCPY to a private JCL library (where `hlq` is the high-level qualifier that is used for MainView for UNIX System Services data sets at your site).

   BBAMVCPY contains JCL to execute IEBCOPY, which will copy BBPARM member BBATSP00 to a UBBPARM library.

2. Customize the JCL in BBAMVCPY by following the instructions at the top of the member.

3. Submit the job.

4. Review the job output to verify that the IEBCOPY job was successful.
5 In UBBPARM member BBATSP00, use the following syntax to enable or disable a product:

```xml
<product>
  <name>ppppp</name>
  <server>ssssssss</server>
  <target>tttttttt</target>
  <enable | disable/>
</product>
```

The variables are specified according to the following table:

<table>
<thead>
<tr>
<th>ppppp</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMF</td>
<td>CMF MONITOR</td>
</tr>
<tr>
<td>MV390</td>
<td>MainView for z/OS</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> When enabling MainView for z/OS, ensure that MainView SYSPROG Services is also enabled to activate the SYSPROG Services component of MainView for z/OS.</td>
</tr>
<tr>
<td>MVUSS</td>
<td>MainView for UNIX System Services</td>
</tr>
<tr>
<td>MVSPS</td>
<td>MainView SYSPROG Services</td>
</tr>
<tr>
<td>MVVP</td>
<td>MainView VistaPoint</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sssssss</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>One- to eight-character job name of the z/OS PAS</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ttttttt</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>One- to eight-character name of the context for the z/OS PAS</td>
<td></td>
</tr>
</tbody>
</table>

6 To disable a product, change `<enable/>` to `<disable/>` in BBATSP00.

7 Save the member and restart the PAS.
CAS library member

BBILIB member @@YZZ022 contains statements for customizing or creating a CAS startup procedure in your system started task procedure library.

CAS started task procedure statements

The following table lists the CAS started task procedure statements.

Table 7: CAS started task procedure statements

<table>
<thead>
<tr>
<th>Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>// procname PROC</td>
<td>This statement identifies the started task procedure and its parameters.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You can override these parameters with the START command for the CAS, as described in the MainView Administration Guide.</td>
</tr>
<tr>
<td>//CAS EXEC</td>
<td>References a program named BBM9ZA00 that processes EXEC parameters and performs basic initialization tasks.</td>
</tr>
<tr>
<td></td>
<td>You can use the PARM parameter to override the default parameter values that are specified with the PROC statement.</td>
</tr>
<tr>
<td></td>
<td>TIME=1440 defines unlimited processing time for the CAS.</td>
</tr>
<tr>
<td></td>
<td>REGION=4096K specifies the size of the private region that is required by the CAS. Do not define a region smaller than 4096 K.</td>
</tr>
<tr>
<td>//STEPLIB DD</td>
<td>Defines the BBLINK load library, which contains load modules for all MainView products. The CAS requires some of these load modules; each product address space requires other modules.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This statement should be deleted if BBLINK is added to your system link list.</td>
</tr>
<tr>
<td>Statement</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| //BBSECURE DD          | Defines the data set that contains security resource definitions. These definitions are used by the CAS and PAS to determine whether security checking should occur for access to systems, products, views, and actions. The same data set name must be defined for the CAS started task procedure BBSECURE statement and the PAS started task procedure BBSECURE statement.  
  **Note:** Defining a BBSECURE data set is only part of setting up security with your external security manager (ESM). For more information, see the *MainView Security Guide*. |
| //BBACTDEF DD          | Defines the data set that contains distributed action and view tables. These tables help the CAS determine valid view names and actions. The BBACTDEF data set also contains default security parameter files and resource definition files. The CAS accesses this data set in response to window commands and actions that are associated with Plex Manager views. |
| //BBVDEF DD            | Defines the library that contains distributed views.                                                                                                                                                           |
| //BBIPARM DD           | Defines and concatenates MainView product parameter libraries. If you establish cross-system communication or create target or SSI contexts, all definitions are stored in the *hlq.UBBPARM* data set, which is created by MainView Customization.  
  *hlq.BBPARM* contains distributed parameters. |

**Related Information**

- “CAS PROC statement parameters” on page 174

---

**CAS PROC statement parameters**

This topic explains each parameter that is available for use with the PROC statement in the CAS started task procedure:
procname parameter

The required procname parameter specifies the name of the startup procedure for the CAS (for example, BBMCAS, as created by MainView Customization).

This parameter uses the following syntax:

```
procname
```

SSID parameter

The required SSID parameter (required for the CAS startup procedure, but optional for the START command) identifies the CAS subsystem ID to the UAS and PAS.

This parameter uses the following syntax:

```
SSID=ssid
```

If an SSID is not specified on the START command, the SSID on the PROC statement in the CAS startup procedure is used. The default is BBCS.
Be sure that your MainView users know the SSID value, because this value is required for the **Subsystem ID** field in the Session Control Parameters panel.

---

**Note**
For a PAS to connect to a CAS, the SSID value in the CAS startup procedure must be the same as:

- The CASID parameter in the BBISSPxx member of BBPARM
- The SSID parameter in the z/OS PAS startup procedure

---

**CAPS parameter**

The optional CAPS parameter indicates whether the CAS should provide Katakana terminal support for console messages and WTO messages that are issued from the CAS.

This parameter uses the following syntax:

\[
\text{CAPS} = \{ \text{Y} \mid \text{N} \}
\]

The default is N (No), which means that console messages and WTO messages appear in mixed case. If you initialize the CAS with CAPS=Y, console messages and WTO messages are converted to uppercase for Katakana terminals.

---

**COLD parameter**

The optional COLD parameter indicates whether the CAS startup should be a cold start.

This parameter uses the following syntax:

\[
\text{COLD} = \{ \text{Y} \mid \text{N} \}
\]

The default is N (No).

When the CAS initializes, several control blocks and load modules are placed in common storage. Most of this storage is freed when the CAS terminates. However, certain blocks are retained to permit the reuse of previously allocated system resources—in particular, system linkage indexes. In addition, two load modules are retained in common storage. COLD=N reuses those control blocks and load modules rather than building new ones.
Specify COLD=Y only when instructed to do so by BMC Customer Support in an effort to clear an error condition. COLD=Y causes the linkage indexes, control blocks, and load modules to be discarded. The linkage indexes and the common storage that is occupied by those resources are lost until the system is IPLed. Inappropriate use of COLD=Y might exhaust available system linkage indexes, at which point an IPL is required to restore normal system operation.

**COMMORDR parameter**

The optional COMMORDR parameter determines the order in which communication methods (XCF, TCP/IP, and VTAM) will be tried in attempts by the local CAS to communicate with remote CASs.

This parameter uses the following syntax:

```
COMMORDR={XTV | xxx }
```

Specify the first letter of one or more communication methods (X, T, or V) in any order. Each specified method will be tried in the order that you specify. The first method that successfully establishes communication with a remote CAS will be used.

The default is XTV, which means that the local CAS will first attempt to communicate with remote CASs by using the Cross-System Coupling Facility (XCF). If that method fails, TCP/IP will be tried and, if that fails, VTAM will be tried.

---

### Note

- Because VTAM communication will be attempted if XCF and TCP/IP communications are not successful, a V will always be included in the COMMORDR string, even if you do not specify it. For example, COMMORDR=XT is interpreted as COMMORDR=XTV.

- To use VTAM for CAS-to-CAS communication, all participating CASs must have the necessary VTAM information defined for them, as described in Define cross-system communication on page 136.

- To use XCF for CAS-to-CAS communication, all participating CASs must also include CONVXCF=Y (either explicitly or by default).

- The SPCF=Y parameter is still accepted in the CAS startup procedure but is ignored. To use the Cross-System Coupling Facility (XCF) to communicate with other CASs in a sysplex, specify CONVXCF=Y and X on the COMMORDR parameter.
CONVXCF parameter

The optional CONVXCF parameter indicates whether a CAS uses the Cross-System Coupling Facility (XCF) to communicate with other CASs in a sysplex environment.

This parameter uses the following syntax:

CONVXCF={Y | N}

The default is Y (Yes). If CONVXCF=N is specified, the CAS cannot participate in XCF communication with other CASs.

To use XCF for CAS-to-CAS communication, all participating CASs must:

- Reside in the same sysplex and XCF group
- Include X on the COMMORDR parameter (either explicitly or by default)

If all the CASs are in the same sysplex and XCF group, CONVXCF=Y allows the CAS to auto-discover and auto-connect to the other CASs through XCF.

DFLTGRP parameter

The optional DFLTGRP parameter identifies the Cross-System Coupling Facility (XCF) group name that is to be used at CAS startup.

This parameter uses the following syntax:

DFLTGRP={BBGROUP | name}

The default is BBGROUP.

DFLTGRP can be used during migration from one release of MainView to another. You can use it to start a test CAS on a system image that is already running a production CAS. By specifying different XCF group names, you can run two CASs on the same system image.

DUMP parameter

The optional DUMP parameter indicates whether system dumps (SDUMPs) are taken when a severe error is detected by a CAS.

This parameter uses the following syntax:
DUMP={Y | N | ALL | YH | ALLH}

DUMP=Y (the default) allows dumps for errors that occur in privileged code.

DUMP=N suppresses all dumps within the CAS.

DUMP=ALL allows dumps for all errors, whether the error occurs in privileged code or problem-state code. You should use ALL only as instructed by BMC Customer Support.

DUMP=YH allows dumps for errors that occur in privileged code, but limits the dump to the home address space (along with summary information from any primary or secondary address space).

DUMP=ALLH allows dumps for all errors, but limits the dump to the home address space (along with summary information from any primary or secondary address space).

**Note**

YH and ALLH are performance-saving options that can restrict BMC Customer Support’s ability to diagnose problems. These options should be used only if the address space cannot tolerate any delay, and only after consulting with Customer Support.

**EMM parameter**

The optional EMM parameter indicates whether extended message mode (EMM) is active when the CAS starts.

This parameter uses the following syntax:

EMM={Y | N}

The default is N (No).

EMM messages are a subset of the messages that are controlled by the XDM parameter. You can enable EMM messages by specifying EMM=Y without enabling XDM messages. However, EMM=Y is not recommended for normal CAS operation; use only as instructed by BMC Customer Support.
**XDM parameter**

The optional XDM parameter indicates whether the CAS should operate in extended diagnostic mode (XDM), which produces additional messages and suspends some CAS error-recovery processing.

This parameter uses the following syntax:

```
XDM={Y|N}
```

The default is N (No).

---

**WARNING**

Specify XDM=Y only when instructed to do so by BMC Customer Support. The XDM=Y parameter disables certain error-recovery processes that could severely impact your CAS subsystem.
BBILIB member BBMPAS contains statements for customizing or creating a PAS startup procedure in your system started task procedure library.

### z/OS PAS started task procedure statements

The following table lists the z/OS PAS started task procedure statements:

<table>
<thead>
<tr>
<th>JCL statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>// procname PROC</td>
<td>This statement identifies the started task procedure and its parameters. You can override these parameters with the START command for the PAS, as described in the MainView Administration Guide.</td>
</tr>
<tr>
<td>//PAS EXEC</td>
<td>This statement references a program named BBM9DA00 that processes EXEC parameters and performs basic initialization tasks. You can use the PARM parameter to override the default parameter values that are specified with the PROC statement. The REGION parameter specifies the size of the private region that is required by the PAS. TIME=1440 defines unlimited processing time for the PAS.</td>
</tr>
<tr>
<td>//STEPLIB DD</td>
<td>This statement defines the BBLINK load library, which contains load modules for all MainView products. The CAS requires some of these load modules; each product address space requires other modules. This statement should be deleted if BBLINK is added to your system link list.</td>
</tr>
</tbody>
</table>
### JCL statement

<table>
<thead>
<tr>
<th>JCL statement</th>
<th>Description</th>
</tr>
</thead>
</table>
| //CMFCPM xx DD      | *(CMF MONITOR only)* This statement specifies one or more data sets where the Extractor records data that is collected by the CPM monitor.  
Do not define this statement if you are recording CPM data to SMF or if you are specifying the data set names on the REPORT Extractor control statement.  
To manually customize the CMFCPMxx DD statements to point to your CPM data sets, remove the DUMMY parameter and add the following statement for each data set that you want defined during Extractor initialization:  
//CMFCPMxx DD      
DISP=SHR,DSN=&BAVSHLQ..&SYSID..CMFOUTxx  
xx is a unique value for the allocated data set.  
The CMFCPMxx DD statements can be defined to write simultaneously to the same data sets as the CMFIPMxx DD statements. During MainView Customization, these statements are dynamically defined, if required.  
Specifying alternate data sets is optional; however, alternate data set support is provided only if an alternate data set is specified.  
See the *CMF MONITOR Customization Guide* for more information about the CPM monitor, allocating CPM data sets, and alternate data set support. See the REPORT section of the *CMF MONITOR Batch User Guide and Reference* for more information about the REPORT Extractor control statement. |
<table>
<thead>
<tr>
<th>JCL statement</th>
<th>Description</th>
</tr>
</thead>
</table>
| //CMFIPM xx DD                | *(CMF MONITOR only)* This statement specifies one or more data sets where the Extractor records data that is collected by the IPM monitor. Do not define this statement if you are recording IPM data to SMF or if you are specifying the data set names on the REPORT Extractor control statement. To manually customize the CMFIPMxx DD statements to point to your IPM data sets, remove the DUMMY parameter and add the following statement for each data set that you want defined during Extractor initialization:  
  //CMFIPMxx DD  
  DISP=SHR,DSN=&BAVSHLQ..&SYSID..CMFOUTxx  
  xx is a unique value for the allocated data set.  
  The CMFIPMxx DD statements can be defined to write simultaneously to the same data sets as the CMFCPMxx DD statements. During MainView Customization, these statements are dynamically defined. Specifying alternate data sets is optional; however, alternate data set support is provided only if an alternate data set is specified.  
  See the CMF MONITOR Customization Guide for more information about the IPM monitor, allocating IPM data sets, and alternate data set support. See the REPORT section of the CMF MONITOR Batch User Guide and Reference for more information about the REPORT Extractor control statement. |
| //CMFCDS xx DD or             | *(CMF MONITOR only)* This statement specifies one or more data sets for DSO data from the IPM and CPM modes. If defined, all CMF MONITOR 240 records that are produced by the HEADMOVE control statement are directed to these data sets. See the DSO User Guide and Reference for more information about the HEADMOVE control statement and allocating DSO data sets.  
  These DD statements function the same way as the CMFCPMxx DD and CMFIPMxx DD statements.  
  Do not define this statement if you are recording DSO data to SMF or if you are specifying the data set names on the HEADMOVE Extractor control statement.  
  Note: HEADMOVE is not supported by version 5.8 and later of CMF MONITOR. |
<p>| //CMFIDS xx DD                |                                                                                                                                                                                                            |</p>
<table>
<thead>
<tr>
<th>JCL statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>//CMFDUMP DD</td>
<td><em>(CMF MONITOR only)</em> This statement defines a print data set for snapshot dumps. When a program fails in CMF MONITOR Online, the abend is intercepted by an ESTAE routine, and a formatted dump of the failing component is printed.</td>
</tr>
<tr>
<td>//CMFLOG DD</td>
<td><em>(CMF MONITOR only)</em> This statement defines an optional file that can direct the CMF MONITOR Control Card Log to an alternative data set. When no CMFLOG DD statement is present, the data set for the Control Card Log is dynamically allocated and deallocated as needed. The CMFLOG data set allows for immediate inspection of messages, regardless of how long an Extractor job runs. Dynamic allocation is to the MSGCLASS that is specified on the JOB statement for batch jobs or to CLASS A for started tasks.</td>
</tr>
<tr>
<td>//CMFMSG DD</td>
<td><em>(CMF MONITOR only)</em> This statement defines a print data set that contains non-initialization messages for the system. When no DD statement is present, the CMFMSG DD data set is dynamically allocated to SYSOUT class A. To print this data set, issue a MSGFREE command by using the MODIFY command. The data set is dynamically deallocated and printed, and a new allocation occurs immediately.</td>
</tr>
<tr>
<td>//CMFXDSLG DD</td>
<td><em>(CMF MONITOR only)</em> This statement defines a print data set that contains XDS initialization messages and control cards. If this DD statement is not present, XDS messages are issued as WTO.</td>
</tr>
<tr>
<td>//PARMLIB DD</td>
<td><em>(CMF MONITOR only)</em> This statement defines a partitioned data set that contains control statement members for the Extractor to read. This data set must be a fixed-block data set with a logical record length of 80, such as security hlq.UBBPARM</td>
</tr>
</tbody>
</table>
### JCL statement

<table>
<thead>
<tr>
<th>JCL statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>//BBSECURE DD</td>
<td>This statement defines the data set that contains security resource definitions. These definitions are used by the CAS and PAS to determine whether security checking should occur for access to systems, products, views, and actions. The same data set name must be defined for the CAS started task procedure BBSECURE statement and the PAS started task procedure BBSECURE statement. Defining a BBSECURE data set is only one part of setting up security to function with your external security manager (ESM). For more information about customizing security, see the <em>MainView Security Guide</em>.</td>
</tr>
<tr>
<td>//BBVDEF DD</td>
<td>This statement defines the library that contains distributed views.</td>
</tr>
<tr>
<td>//BBACTDEF DD</td>
<td>This statement defines the data set that contains distributed action and view tables.</td>
</tr>
<tr>
<td>//USSDUMP DD</td>
<td><em>(MainView for UNIX System Services only)</em> This statement is used by the OEDGTRC trace function for problem diagnosis at the request of BMC Customer Support.</td>
</tr>
<tr>
<td>//PARMFILE DD</td>
<td>This statement defines a VSAM data set that contains:</td>
</tr>
<tr>
<td></td>
<td>- Names of historical data sets for the DSLIST view</td>
</tr>
<tr>
<td></td>
<td>- User-defined workload definitions that are created through the MainView for z/OS WKLIST view</td>
</tr>
<tr>
<td></td>
<td>This data set is allocated and initially loaded during the customization process.</td>
</tr>
<tr>
<td>//HISTDS00 DD</td>
<td>This statement defines up to 100 VSAM data sets that compose the historical database.</td>
</tr>
<tr>
<td>//HISTDS01 DD</td>
<td></td>
</tr>
<tr>
<td>//HISTDS02 DD</td>
<td></td>
</tr>
</tbody>
</table>

### Related Information

- "z/OS PAS PROC statement parameters" on page 186
This topic explains each parameter that is available for use with the PROC statement in the z/OS PAS started task procedure:

**Related Information**

- “BAVSHLQ parameter” on page 187
- “BBCHILV parameter” on page 187
- “BBLINK parameter” on page 187
- “BBSOUT parameter” on page 187
- “CASWAIT parameter” on page 188
- “CMDID parameter” on page 188
- “CPM parameter” on page 189
- “CXEN parameter” on page 189
- “DC parameter” on page 190
- “EM parameter” on page 191
- “IPM parameter” on page 191
- “procname parameter” on page 192
- “RGN parameter” on page 192
- “SSID parameter” on page 192
- “SYSID parameter” on page 193
- “TZOVRLAP parameter” on page 193
- “UBBPARM parameter” on page 194
- “USEZIIP parameter” on page 194
- “XDM parameter” on page 194
- “XDS parameter” on page 195
**BAVSHLQ parameter**

The required BAVSHLQ parameter defines the high-level qualifier for the VSAM data sets that are used for historical data and internal and user-defined system workloads.

This parameter uses the following syntax:

`BAVSHLQ=hlq`

**BBCHILV parameter**

The required BBCHILV parameter defines the high-level qualifier for the product libraries.

This parameter uses the following syntax:

`BBCHILV=hlq`

**BBLINK parameter**

The optional BBLINK parameter defines the name to be assigned to BBLINK.

This parameter uses the following syntax:

`BBLINK=name`

---

**Note**

If BBLINK is in your system link list, this parameter and the STEPLIB DD statement should be removed.

---

**BBSOUT parameter**

The required BBSOUT parameter defines a valid SYSOUT class for started tasks.

This parameter uses the following syntax:

`BBSOUT=class`
CASWAIT parameter

The optional CASWAIT parameter (used for CMF MONITOR and MainView for z/OS only) specifies the amount of time that the PAS should wait for the CAS to start, both during initial PAS startup and if the CAS terminates while the PAS is executing.

This parameter uses the following syntax:

\[ \text{CASWAIT=}n\text{nn}n \]

\( n\text{nn}n \) can be any number of minutes from 0 to 9999. The default is 10 minutes. If you specify CASWAIT=0, the initialization process does not retry the connection to the CAS.

CMDID parameter

The optional CMDID parameter defines a single character that can be used in place of the MODIFY command and procname when a MODIFY command is issued to the PAS.

This parameter uses the following syntax:

\[ \text{CMDID=}s\text{ymbol} \]

You can use the CMDID value as a quick method for issuing commands. You must manually add the CMDID parameter to the PAS procedure statement to enable this support. Valid operands are as follows:

<table>
<thead>
<tr>
<th>Valid operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;</td>
</tr>
<tr>
<td>/</td>
</tr>
<tr>
<td>#</td>
</tr>
</tbody>
</table>

Certain operands might need to be enclosed within quotation marks to work in your environment. Alternatively, operands can be specified as a two-digit hexadecimal representation.

For information about using the MODIFY command to control the z/OS PAS, see the MainView Administration Guide.
CPM parameter

This topic defines the optional CPM parameter (used for CMF MONITOR and MainView for z/OS only).

The CPM parameter:

- Specifies the two-character suffix of the CMFCPMxx control statement member in the data set to which the PARMLIB DD statement points
- Indicates not to run the CPM mode samplers

This parameter uses the following syntax:

```plaintext
CPM={xx | 00 | STOP}
```

**WARNING**

CPM=STOP terminates the address space unless IPM monitoring mode is active.

The default is 00.

The control statement member contains Extractor REPORT and sampler control statements. This member controls the operation of the CPM mode samplers and defines the resources that the Extractor monitors.

CPM mode samplers execute either when a MODIFY command is issued or when the CPM parameter is used. Sample control statement members are in the `hlq.UBBPARM` data set that was created by MainView Customization.

The CPM mode control statement member that the Extractor uses can be changed while the Extractor is active by using the MODIFY command; for more information, see the MainView Administration Guide.

CXEN parameter

The optional CXEN parameter (used for CMF MONITOR only) controls whether more than one copy of the Extractor can initialize on the same system.

```plaintext
CXEN={Y | N}
```

The default is Y (Yes), which allows enqueue sampling, workload sampling, and writing to SMF data sets. You can run more than one CMF Extractor on your system at a time. However, only one Extractor can sample workload data or enqueue data.
By specifying CXEN=N for any additional Extractors, more than one copy of CMF MONITOR Extractor can execute on the same system concurrently. CXEN=N prevents the Extractor from writing to SMF data sets, sampling workload data, and sampling enqueue data.

**Note**
Because of the associated overhead, you should not run multiple Extractors unless it is required in trial situations to evaluate MainView for z/OS. CXEN=N should be defined to a trial Extractor, if necessary.

### DC parameter

The optional DC parameter (used for CMF MONITOR and MainView for z/OS only) affects the samplers that are associated with the PGDDLAY and CFDATA Extractor control statements and the z/OS PAS data collectors.

This parameter uses the following syntax:

```
DC=({START | STOP | CPM | IPM})
```

**WARNING**
DC=STATUS should not be specified at z/OS PAS initialization because the data collectors will not initialize. This command should be used only when the PAS is running.

DC=CPM causes the z/OS PAS data collectors to initialize and accept data from the Extractor in CPM monitoring mode at startup. Specifying DC=CPM is the same as specifying DC=START.

DC=IPM causes the z/OS PAS data collectors to initialize and accept data from the Extractor in IPM monitoring mode at startup. If you specify DC=IPM, you must also specify the parameter IPM=xx, where xx is the suffix of the CMFIPMxx control statement member to which the //PARMLIB DD statement points.

DC=START (the default) causes z/OS PAS data collectors to initialize at startup under CPM monitoring mode. When the data collectors are initialized, access to CMF MONITOR Online, MainView for z/OS, or both is provided. The CMF MONITOR samplers that are associated with the PGDDLAY and CFDATA Extractor control statements are enabled for sampling.

DC=STOP prevents the data collectors from initializing when the z/OS PAS initializes. You cannot access CMF MONITOR Online or MainView for z/OS. The PGDDLAY and CFDATA CMF Extractor control statements do not sample data because they gather information from the z/OS PAS data collectors.
The DC parameter can also be controlled by using a MODIFY command; MODIFY enables the data collectors to be started or stopped, or the monitoring modes to be switched without restarting the z/OS PAS and the Extractor. When the MODIFY command is issued, you can specify an additional value with the DC parameter. DC=STATUS displays the status of each data collector in the job log.

For more information about how to use the MODIFY command to change z/OS PAS operation, see the MainView Administration Guide.

**EM parameter**

The required EM parameter (used for MainView for z/OS only) specifies the two-character suffix of the PWSCPMxx control statement member that you want the MainView for z/OS Exception Monitor to use.

This parameter uses the following syntax:

\[
EM=\{ xx \mid 00 \}
\]

The default is 00.

**IPM parameter**

This topic defines the optional IPM parameter (used for CMF MONITOR and MainView for z/OS only).

The IPM parameter:

- Specifies the two-character suffix of the CMFIPMxx control statement member in the data set to which the PARMLIB DD statement points:

- Indicates not to run the IPM mode samplers

This parameter uses the following syntax:

\[
IPM=\{ xx \mid 00 \mid STOP \}
\]

The default is 00.
You should add this parameter to the PAS started task procedure only if you always want to start the IPM monitor when the CPM monitor starts. Member CMFIPM00 is built as part of the installation process. If you do not want the IPM monitor to run, specify IPM=STOP.

The control statement member contains the REPORT Extractor control statement and sampler control statements, which are described in the *CMF MONITOR User Guide and Reference*. This member controls the operation of the IPM mode samplers and defines the resources that the Extractor monitors.

IPM mode samplers execute either when a MODIFY command is issued or when the IPM parameter is used. Sample control statement members are in the `hlq.UBBPARM` data set that was created by MainView Customization.

The IPM mode control statement member that the Extractor uses can be changed while the Extractor is active by using the MODIFY command; for more information, see *MainView Administration Guide*.

**procname parameter**

The required procname parameter identifies the startup procedure for the PAS (for example, MVSPAS).

This parameter uses the following syntax:

```
procname
```

**RGN parameter**

The required RGN parameter assigns a region size of 32 megabytes.

This parameter uses the following syntax:

```
RGN=32
```

**SSID parameter**

The required SSID parameter specifies the subsystem ID of the CAS to which the z/OS PAS should connect.
This parameter uses the following syntax:

**SSID=ssid**

For a PAS to connect to a CAS, the SSID values in both the CAS and PAS startup procedures must be the same.

**Note**
The CAS is started as a separate address space from the z/OS PAS and must be initialized before the z/OS PAS is initialized.

### SYSID parameter

The required SYSID parameter specifies the SMF system ID for your system.

This parameter uses the following syntax:

**SYSID=sysid**

### TZOVRLAP parameter

The optional TZOVRLAP parameter (used by CMF MONITOR CPM and IPM sessions only) specifies the recording of CMF SMF records when the Time Zone offset is changed and an overlap exists in the local time.

This parameter uses the following syntax:

**TZOVRLAP=RECORD | DISCARD**

- **TZOVRLAP=RECORD** causes all output records to be written. Consequently, two sets of interval records cover one hour when the Time Zone offset is changed backwards (that is, an hour earlier).

- **TZOVRLAP=DISCARD** causes the CMF SMF records to be discarded for the intervals that follow the change that overlap the previously recorded intervals.

This parameter is applicable only to CMF MONITOR CPM and IPM sessions that are writing SMF records (to SMF or to the CMFCPMx or to CMFIPMx DDs). Otherwise, this parameter is ignored.

This value can be altered at any time by using a MODIFY command.
UBBPARAM parameter

The required UBBPARAM parameter defines the name to be assigned to the user parameter library.

This parameter uses the following syntax:

\[ \text{UBBPARAM} = \text{name} \]

USEZIIP parameter

The optional USEZIIP parameter (used for MainView for z/OS or CMF MONITOR only) specifies whether eligible CPU processing should be offloaded to zIIPs if zIIPs are online.

This parameter uses the following syntax:

\[ \text{USEZIIP} = (\text{Y} \mid \text{N}) \]

The default is Y (Yes).

The USEZIIP parameter can also be specified using the MODIFY command, which gives you control over whether zIIP-eligible processing should be offloaded from standard CPs to zIIPs.

XDM parameter

The optional XDM parameter indicates whether the z/OS PAS should execute in extended diagnostic mode (XDM). This mode produces additional messages and suspends some PAS error-recovery processing.

This parameter uses the following syntax:

\[ \text{XDM} = (\text{Y} \mid \text{N}) \]

**WARNING**

Specify XDM=Y only when instructed to do so by BMC Customer Support. The XDM=Y parameter disables certain error-recovery processes that could severely impact your system.

The default is N (No).
**XDS parameter**

The optional XDS parameter (used for CMF MONITOR only) controls the collection of data by the cross-system data server (XDS).

This parameter uses the following syntax:

\[ DS=\{xx \mid \text{STOP}\} \]

Requirements for sharing cross-system data are as follows:

- All systems must be in the same sysplex.
- The z/OS PAS on each system must have XDS active.
- DC=START must be specified for each z/OS PAS.
- The CAS on all systems must be in the same XCF group.

For SDSF to use XDS data, the BBLINK data set needs to be in the system link list or in the TSO logon procedure STEPLIB.

To collect XDS data, specify XDS=xx, where xx indicates the suffix of a CMFXDS xx member in the hlq.UBBPARM data set that was created by MainView Customization. You can switch to a different XDS member without stopping the z/OS PAS or XDS by using the MODIFY command.

**Example**

If you are running the z/OS PAS pointing to the CMFXDS00 member with XDS=00, you can begin using the CMFXDS01 member by issuing the following MODIFY command: \texttt{F MVSPAS,XDS=01}. To stop XDS without stopping the z/OS PAS, issue the following MODIFY command: \texttt{F MVSPAS,XDS=STOP}.

The parameters for the CMFXDSxx members are TYPE, RECORDS, and SIZE. Each parameter can be specified only once in a particular member. If a line begins with an asterisk (*), that line is ignored. Data in columns 73 through 80 is also ignored.

**TYPE**

The TYPE parameter defines which SMF record types are included in the XDS SMF record buffer for the CX10XDQY and CX10XDRC APIs.

- Specify `TYPE CMF` (the default) to use output records from CMF MONITOR itself. Use this to buffer all type 70-78 records, as well as all CMF MONITOR user record types.
Specify **TYPE SMF** to use SMF IEFU83, IEFU84, and IEFU85 dynamic exits. Use this to buffer all record types except those types that are written to CMF Extractor output data sets. If the CMF Extractor is writing to SMF, **TYPE SMF** works exactly the same as **TYPE ALL**.

Specify **TYPE ALL** to use both SMF-provided and CMF MONITOR records.

Specify **TYPE NONE** to allow API calls and CX10XDGS requests for type 79 data from this system without buffering records on this system.

### RECORDS

The RECORDS parameter defines which SMF record types and subtypes are included in the XDS buffer.

This set of records must be a subset of the records that are specified for collection by the TYPE parameter. Records that are specified with this parameter are available for CX10XDQY (XDS record query) and CX10XDRC (XDS record retrieval) API calls. In addition, cross-system snapshots of type 79 data are available from the CX10XDGS (XDS data-gathering service) API call.

Specify the records that you want to include in the buffer by listing those types in a list that follows the RECORDS parameter. For example, if you want to buffer type 70 and 72 records only, specify `RECORDS 70,72`. Subtypes can be specified by placing a hyphen (`-`) after a type, followed immediately by the subtype. For example, to buffer only subtype 3 of type 72 records, specify `RECORDS 72-3`.

Ranges of types and subtypes can be specified by using a colon (`:`) (to separate the start and end of a range) or by using an `X` character (to indicate all digits in a range). For example, to buffer record types 70 through 79, you can specify either `RECORDS 70:79` or `RECORDS 7X`.

If you do not want to buffer any records, specify `RECORDS NONE`. If you want to buffer all record types that are included in the TYPE parameter, specify `RECORDS ALL`. ALL is the default if you do not include a RECORDS parameter.

### SIZE

The SIZE parameter defines how much SMF data should be buffered before records are lost because of buffer wraparound.

Specify a size in the format `SIZE nnnu`, where `nnn` is a one- to nine-digit number and `u` is one of the following values:

- blank (bytes)
- K (kilobytes, or $2^{10}$ bytes)
- M (megabytes, or $2^{20}$ bytes)
- G (gigabytes, or $2^{30}$ bytes)

For example, for a buffer size of 3 megabytes, specify `SIZE 3M`. If you do not specify the SIZE parameter, the default buffer size is $32M$.

You can also specify the word PURGE with the SIZE parameter to indicate that the existing buffer should be discarded at the time of a MODIFY command, rather than when the new buffer fills up.

**Example**

If you want to change a buffer size from 3 megabytes to 5 megabytes and immediately discard the 3-megabyte buffer, specify `SIZE 5M,PURGE` in the member that is referred to by the MODIFY command.
BBI-SS PAS library members

This appendix describes some of the library members and the parameters in them that affect the BBI-SS PAS product environment. The following members and their parameters are discussed:

BBIBBCFG parameters

BBPARM member BBIBBCFG is a model for UBBPARM members to identify suffixes of BBPARM configuration members to the BBI-SS PAS.

The following products use the BBIBBCFG member:

- MainView AutoOPERATOR
- MainView FOCAL POINT
- MainView for CICS
- MainView for DB2
- MainView for DBCTL
- MainView for IMS Online
- MainView for IMS Offline
- MainView for WebSphere MQ

By using the BBCFG facility, multiple BBI-SS PASs can process different BBPARM configuration members, even when the PASs are sharing the same BBPARM data set.

The suffix for a BBPARM configuration member can be any two characters that are valid for a partitioned data set member name.

The following parameters are used in the BBIBBCFG member:
"AAOALS parameter" on page 200

"AAOARP parameter" on page 201

"AAOEXP parameter" on page 201

"AAOGME parameter" on page 201

"AAOMQL parameter" on page 201

"AAOPRM parameter" on page 202

"AAOTSP parameter" on page 202

"BBIISP parameter" on page 202

"BBIISSP parameter" on page 203

"BBKMON parameter" on page 203

"CMRBEX parameter" on page 203

"CMRDTL parameter" on page 203

"CMRPRB parameter" on page 204

"CMRSOP parameter" on page 204

"CMRTTH parameter" on page 204

---

AAOALS parameter

(MainView AutoOPERATOR only) AAOALSxx members contain information that controls how much storage MainView AutoOPERATOR ALERTs use.

This parameter uses the following syntax:

AAOALS={ xx | 00 }

The default is 00.
AAOARP parameter

(MainView AutoOPERATOR only) AAOARPx members contain parameters that control how the MainView AutoOPERATOR Automation Reporter application operates.

This parameter uses the following syntax:

\[
\text{AAOARP} = \{ xx \mid 00 \}
\]

The default is 00.

AAOEXP parameter

(MainView AutoOPERATOR only) AAOEXPxx members contain parameters and thresholds that determine how efficiently MainView AutoOPERATOR EXECs execute on your system.

This parameter uses the following syntax:

\[
\text{AAOEXP} = \{ xx \mid 00 \}
\]

The default is 00.

AAOGME parameter

(MainView AutoOPERATOR only) AAOGMEx members contain parameters that MainView AutoOPERATOR uses to create General Messages Exchange (GME) connections to allow applications to communicate.

This parameter uses the following syntax:

\[
\text{AAOGME} = \{ xx \mid 00 \}
\]

The default is 00.

AAOMQL parameter

(MainView AutoOPERATOR only) AAOMQLxx members contain parameters that MainView AutoOPERATOR for WebSphere MQ uses to specify which MQ queues are listened to by MainView AutoOPERATOR for automation.
This parameter uses the following syntax:

\[ \text{AAOMQL} = \{xx | 00\} \]

The default is 00.

**AAOPRM parameter**

*(MainView AutoOPERATOR only)* This value is the BBPARM member suffix that is used by the BBI-SS PAS.

This parameter uses the following syntax:

\[ \text{AAOPRM} = \{xx | 00\} \]

The default is 00.

**AAOTSP parameter**

*(MainView AutoOPERATOR only)* AAOTSPxx members contain parameters that control how the MainView AutoOPERATOR TapeSHARE application operates.

This parameter uses the following syntax:

\[ \text{AAOTSP} = \{xx | 00\} \]

The default is 00.

**BBIISP parameter**

BBIISPxx members contain the BBI-SS PAS Interval Services parameters.

The suffix can be any two characters that are valid for a partitioned data set member name.

This parameter uses the following syntax:

\[ \text{BBIISP} = \{xx | 00\} \]

The default is 00.
**BBISSP parameter**

BBISSPxx members contain the BBI-SS PAS subsystem control parameters.

This parameter uses the following syntax:

\[ \text{BBISSP} = \{ xx \mid 00 \} \]

The default is 00.

**BBKMON parameter**

(MainView for CICS only) BBKMONxx members contain windows-based monitor definitions.

This parameter uses the following syntax:

\[ \text{BBKMON} = \{ xx \mid 00 \} \]

The default is 00.

**CMRBEX parameter**

(MainView for CICS only) CMRBEXxx members contain application trace default value definitions.

This parameter uses the following syntax:

\[ \text{CMRBEX} = \{ xx \mid 00 \} \]

The default is 00.

**CMRDTL parameter**

(MainView for CICS only) CMRDTLxx members contain transaction performance history file definitions.

This parameter uses the following syntax:

\[ \text{CMRDTL} = \{ xx \mid 00 \} \]
The default is 00.

**CMRPRB parameter**

(MainView for CICS only) CMRPRBxx members contain problem table threshold definitions.

This parameter uses the following syntax:

```
CMRPRB={ xx | 00 }
```

The default is 00.

**CMRSOP parameter**

(MainView for CICS only) CMRSOPxx members contain options table definitions.

This parameter uses the following syntax:

```
CMRSOP={ xx | 00 }
```

The default is 00.

**CMRTTH parameter**

(MainView for CICS only) CMRTTHxx members contain Task Kill Exit threshold definitions.

This parameter uses the following syntax:

```
CMRTTH={ xx | 00 }
```

The default is 00.

**BBIISPxx parameters**

BBPARM member BBIISPxx contains parameters that describe the initialization of timer-driven services.
Member BBIISPxx supports the substitution of system variables.

The following topics explain each parameter that is used in the BBIISPxx member. For other product specific parameters, see the comment field in BBIISPxx.

**Related Information**

- “MAXREQ parameter” on page 205
- “INTERVAL parameter” on page 205
- “IRRI parameter” on page 206
- “AUTOID parameter” on page 207
- “TARGET BLK and USRID parameters” on page 207
- “TRDIR and SUBSYS parameters” on page 208
- “System variables substitution in parameter library members” on page 233

### MAXREQ parameter

The optional MAXREQ parameter specifies the maximum number of concurrent timer-driven requests.

This parameter uses the following syntax:

```
MAXREQ={nnnn | 500}
```

The default is 500. The maximum value is 3000.

### INTERVAL parameter

The optional INTERVAL parameter specifies the default timer interval when an interval is not supplied by the request itself.

This parameter uses the following syntax:

```
INTERVAL={hh:mm:ss | 00:01:00}
```

The default is one minute (00:01:00).
**IRRI parameter**

The optional IRRI parameter specifies a recording interval in hours and minutes when records are to be written to a historical data set.

This parameter uses the following syntax:

```
IRRI={ hh:mm | 00:15 }
```

The default is 15 minutes (00:15).

**IRSYNCH parameter**

The optional IRSYNCH parameter specifies how many minutes into the hour to offset the recording interval set by the IRRI parameter.

This parameter uses the following syntax:

```
IRSYNCH=[ 0 | nn | QTR | HALF | HOUR ]
```

Possible values are as follows:

- 0 (the default, which records data as specified by the IRRI parameter)
- `nn` (any value between 1 and 60 minutes)
- QTR (15 minutes)
- HALF (30 minutes)
- HOUR (60 minutes)

For example, assume that IRSYNCH=8 and IRRI=15. Data recording begins 8 minutes after the hour and recurs every 15 minutes, as follows (where `xx` represents the hour):

- `xx:08:00`
- `xx:23:00`
- `xx:38:00`
- `xx:53:00`
These parameter settings can impact all products running in the BBI-SS PAS that process interval record data. Use the BBI Control command .D P (Display Products) to display the list of active products in a BBI-SS PAS.

**Note**
You can specify different IRSYNCH values for different BBI-SS PASs. When you retrieve historical interval data from these PASs, the display contains data from different interval boundaries for the same MVI TIME interval.

### AUTOID parameter

The optional AUTOID parameter specifies the logon ID of a user to be associated with automatically started timer requests.

This parameter uses the following syntax:

```
AUTOID=xxxxxxxx
```

This value is used if USRID is not specified. If USRID is specified, the USRID value supersedes the AUTOID value. AUTOID is written on a separate line from the TARGET statement.

### TARGET BLK and USRID parameters

The required TARGET and BLK parameters and the optional USRID parameter are used by all products.

These parameters use the following syntax:

```
TARGET=xxxx,
BLK=xxxxxxxx,
[USRID=xxxxxxxx]
```

- TARGET specifies the target system that is to receive requests for BLK. The value can be the target system ID or the subsystem ID. TARGET is written on the same line as BLK and can be repeated.

- BLK specifies the name of a member that contains multiple timer-driven requests to be invoked at BBI-SS PAS startup. This member must exist in the BBPARM data set. BLK is written on the same line as TARGET.
USRID specifies the logon ID of a user to be associated with the timer requests in the member that is specified with the BLK parameter. USRID is written on the same line as TARGET and BLK. If AUTOID is also specified, USRID supersedes AUTOID.

TRDIR and SUBSYS parameters

The optional TRDIR and SUBSYS parameters are used for MainView for CICS, MainView for IMS Online, MainView for DBCTL, and MainView for DB2 only.

These parameters use the following syntax:

TRDIR=xxxxxxxx
SUBSYS=ssid

- TRDIR specifies the name of the data set that is used for a trace directory (required for trace logging).
- SUBSYS specifies the subsystem name to which this trace directory applies. This parameter allows multiple BBI-SS PASs to share the same BBIISPxx member.

BBIISPxx parameters for the BBI-SS PAS environment

This topic explains each parameter that is used in the BBIISPxx member. BBIISPxx can be used by all products that run in the BBI-SS PAS:
**CASID parameter**

The required CASID parameter specifies the one- to four-character ID of the MainView coordinating address space (CAS).

This parameter uses the following syntax:

```
CASID=xxxx
```
CMDCHAR parameter

The optional CMDCHAR parameter defines the command characters to be used for issuing control commands from a system console to this subsystem.

This parameter uses the following syntax:

\[ \text{CMDCHAR}=x \]

If CMDCHAR is not specified, control commands can still be issued from a console with the MODIFY command. This parameter is optional, but if CMDCHAR is specified, it must not be a blank (meaning CMDCHAR= is not valid). This parameter is mutually exclusive with the CMDPREF parameter.

CMDPREF parameter

The optional CMDPREF parameter defines the MVS command prefix to be used for issuing control commands from an MVS console to the subsystem.

Without CMDPREF, control commands from a console must be issued by using the MVS MODIFY command.

This parameter uses the following syntax:

\[ \text{CMDPREF}=xxxxxxxx \]

The value is 1 to 8 alphanumeric or special characters.

The value must not be a duplicate, subset, or superset of preexisting MVS command prefixes in the MVS LPAR.

This parameter is optional and is mutually exclusive with CMDCHAR parameter.

DAE parameter

The optional DAE parameter specifies whether dumps with duplicate symptom strings are suppressed when DUMPS=SDUMP is specified.

This parameter uses the following syntax:

\[ \text{DAE}=[\text{YES} \mid \text{NO}] \]
Specify DAE=NO only in controlled diagnostic situations when instructed to do so by BMC Customer Support.

**DUMPCLAS parameter**

The optional DUMPCLAS parameter specifies the output class for formatted dumps.

This parameter uses the following syntax:

\[ \text{DUMPCLAS} = \{ x | A \} \]

The default value is A.

**DUMPS parameter**

The optional DUMPS parameter specifies whether formatted dumps are to be taken in case of a failure.

This parameter uses the following syntax:

\[ \text{DUMPS} = \{ \text{YES} | \text{NO} | \text{SDUMP} | \text{ALL} \} \]

- If YES is specified, a BBI-formatted dump is created.
- If NO is specified, only LOGREC records are written, and no SVC dump or BBI-formatted dump is created.
- If SDUMP (the default) is specified, SVC dumps are created and DAE=YES is used to suppress duplicate SVC dumps. For dumps that are related to MainView AutoOPERATOR, you should use DUMPS=SDUMP.
- If ALL is specified, both an SVC dump and a BBI-formatted dump are created.

**JESCNCT parameter**

The optional JESCNCT parameter specifies whether the BBI-SS PAS should attempt to connect to the job entry system (JES).

This parameter uses the following syntax:

\[ \text{JESCNCT} = \{ \text{YES} | \text{NO} \} \]
This parameter is valid only when the PAS is started:

- During system IPL, before JES is started
- Outside the control of JES, using SUB=MSTR on the START command

If YES is specified, the BBI-SS PAS attempts to connect to JES as soon as JES is available. If the connection is successful, the BBI-SS PAS is given:

- A JES job ID
- A JES job log
- The ability to use JES-dependent services, such as SUBMIT processing or SYSOUT=x for the allocation of subsystem-specific data sets (if MainView AutoOPERATOR is present)

If NO (the default) is specified, no attempt is made to connect to JES, and no JES services are available to the BBI-SS PAS.

**JOURNAL parameter**

The optional JOURNAL parameter determines whether subsets of the journal can be displayed.

This parameter uses the following syntax:

```
JOURNAL={STD | STANDARD | ENHANCED}
```

If STD or STANDARD (the default) is specified, the entire journal is always shown in the Log Display application.

If ENHANCED is specified, you can use Log Display PROFILE specifications to display subsets of the journal.

**JRNLCMSG parameter**

The optional JRNLCMSG parameter specifies whether split BBI-SS PAS journal log messages are to be identified in the BBIPRINT journal log print data set.

This parameter uses the following syntax:

```
JRNLCMSG={YES | NO}
```
The default is YES. Split messages are identified by a one-character prefix as follows:

1 (first line of a split message)

+ (continuation line of a split message)

- (ending line of a split message)

**JRNLSTC parameter**

The optional JRNLSTC parameter specifies the name of the started task that is used to dump and reset the BBI-SS PAS journal logs.

This parameter uses the following syntax:

```
JRNLSTC={xxxxxxxx | BBIDLOG}
```

The name can be from 1 to 8 characters long. The default is BBIDLOG.

**LOGGING parameter**

The optional LOGGING parameter specifies whether BBI-SS PAS journaling is to be performed.

This parameter uses the following syntax:

```
LOGGING={YES | NO}
```

The default is YES.

**MAXPQE parameter**

The optional MAXPQE parameter specifies the maximum number of PQEs that are to be used concurrently in CSA between the target and the BBI-SS PAS.

This parameter uses the following syntax:

```
MAXPQE={nnnn | 2000}
```

The default is 2000. The maximum number of concurrent PQEs is 99999.
MAXUSER parameter

The optional MAXUSER parameter specifies the maximum number of concurrent terminal sessions that can access this BBI-SS PAS.

This parameter uses the following syntax:

```
MAXUSER={nn | 99}
```

The default is 99. Valid values are 1 to 500.

MVLOG parameter

The optional MVLOG parameter controls the logging of BBI-SS PAS journal messages to MainView Logger.

This parameter uses the following syntax:

```
MVLOG={JRNL | YES | NO}
```

The default is NO. BBI-SS PAS journal records are not written to MainView Logger. Journal records are still written to the BBI-SS PAS journal log.

If YES is specified, BBI-SS PAS journal records are written to MainView Logger and the journal log. These messages are viewed by using the LOGMSG view.

If JRNL is specified, BBI-SS PAS journal messages are written to MainView Logger. These message are viewed by using the LOGJRNL view, which filters the BBI-SS messages and provides additional information compared to the LOGMSG view. BMC recommends using the JRNL option rather than the YES option.

---

**Note**

- Before you direct journal records to MainView Logger, you should review the size of the log data sets to ensure that they are large enough. For information about specifying the size of MainView Logger log data sets, see Creating MainView Logger LOGGERxx PARMLIB member on page 88.

- The MainView Logger started task must be running in order for journal records to be written to its log data sets. MainView Logger does not start automatically when you specify MVLOG=YES. For information about starting MainView Logger, see Creating the MainView Logger started task procedure on page 90.
**MVLOGSS parameter**

When MVLOG=YES is specified, the optional MVLOGSS parameter identifies the subsystem ID of the MainView Logger instance to which records will be written.

This parameter uses the following syntax:

```
MVLOGSS=ssid
```

If MVLOGSS is not specified when MVLOG=YES is specified, journal records are written to the default instance of MainView Logger. The default logger for a system image is defined in BBPARM member LOGGER00.

---

**PRODUCT parameter**

The required PRODUCT parameter specifies the code of a product to be initialized within this BBI-SS PAS.

This parameter uses the following syntax:

```
PRODUCT=xxx
```

Valid values are AAO, CMR, DMR, IIZ, MFP, MVDBC, MVIMS, MVMQS, and MVVP.

---

**STALL parameter**

The optional STALL parameter specifies the maximum number of seconds to complete the initialization process.

This parameter uses the following syntax:

```
STALL={nnn | 180}
```

If this value is exceeded, the BBI-SS PAS fails to initialize and issues a STALL DETECTED message. The default is 180. A value of 0 sets the stall limit to 8 hours.
This topic explains each parameter that is used in the BBISSPxx member for MainView AutoOPERATOR only:

**Related Information**

- “ALRTRCVE parameter” on page 217
- “ARMPOLCY parameter” on page 217
- “AUDIT parameter” on page 218
- “CMDCON parameter” on page 218
- “CMDWAIT parameter” on page 219
- “CNSLEXEC parameter” on page 219
- “CONSOLES parameter” on page 219
- “HILITE parameter” on page 220
- “JESCHAR parameter” on page 220
- “JES3CHAR parameter” on page 220
- “JES3CMD parameter” on page 221
- “JESFLTR parameter” on page 221
- “JESTYPE parameter” on page 222
- “MCTSIZE parameter” on page 222
- “MIGCONS parameter” on page 222
- “NETVIEW parameter” on page 223
- “PROFUSED parameter” on page 223
- “XCFGROUP parameter” on page 223
**ALRTRCVE parameter**

The optional ALRTRCVE parameter specifies whether this BBI-SS PAS is to be considered a product alert receiver.

This parameter uses the following syntax:

```
ALRTRCVE={YES | NO}
```

If YES is specified, the PAS can receive product alerts and alarm messages from other BMC products.

If NO (the default) is specified, the PAS cannot receive alerts from other products. The PAS can, however, process alerts that are created by MainView AutoOPERATOR EXECs, Rules, or AOAnywhere calls in this BBI-SS PAS.

To determine whether a PAS is an alert receiver, issue the following BBI control command:

```
.D XCF
```

If message AB3006I is displayed, the PAS is an alert receiver.

**ARMPOLCY parameter**

The optional ARMPOLCY parameter indicates whether the Automatic Restart Manager (ARM) function is specified.

This parameter uses the following syntax:

```
ARMPOLCY{YES | NO | xxxxxxxxx}
```

- If YES is specified, the BBI-SS PAS will use the ARM default restart policy. The default ARM restart policy is that ARM restarts an abended address space up to three times within 5 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.

- If NO (the default) is specified, the BBI-SS PAS will not use ARM.

- xxxxxxxxx specifies the one- to eight-character name of a user-created policy for controlling BBI-SS PAS restarts.

For more information about implementing ARM support for MainView AutoOPERATOR, see the *MainView AutoOPERATOR Customization Guide*. 
AUDIT parameter

This topic defines the optional AUDIT parameter.

The optional AUDIT parameter indicates:

- Whether Rule Set updates and time-initiated setups and purges are recorded in the journal
- Whether a request to delete an alert on an Alert Management view should be recorded in the journal

This parameter uses the following syntax:

```
AUDIT={YES | NO}
```

The default is NO.

The following processes are not monitored with AUDIT=YES:

- Time-initiated EXEC SETs and PRGs that are issued from the TI application
- SETs and PRGs that are issued from the TI application

CMDCON parameter

The optional CMDCON parameter identifies an active, valid console that is available to MainView AutoOPERATOR and is not being used by another address space.

This parameter uses the following syntax:

```
CMDCON={id | MCSname | 0}
```

Specify one of the following values:

- A two-byte console ID (from 1 to 99)
- A two- to eight-character multiple console support (MCS) console name

MCS console names are defined in SYS1.PARMLIB member CONSOL xx and are generally more meaningful in a sysplex environment.

The default is console 0.
CMDWAIT parameter

The optional CMDWAIT parameter defines the number of seconds to wait for a response to a system command that was issued by a terminal session user.

This parameter uses the following syntax:

\[
\text{CMDWAIT=}\{xx \mid 3\}
\]

The default is 3.

CNSLEXEC parameter

The optional CNSLEXEC parameter indicates whether operators can invoke EXECs through the console interface (MODIFY command or subsystem character command).

This parameter uses the following syntax:

\[
\text{CNSLEXEC=}\{\text{YES} \mid \text{NO}\}
\]

The default is YES.

CONSOLES parameter

The optional CONSOLES parameter (used for MainView AutoOPERATOR for z/OS only) represents the number of X-MCS consoles (without MIGID) to be acquired by MainView AutoOPERATOR.

This parameter uses the following syntax:

\[
\text{CONSOLES=}\{nn \mid 4\}
\]

Valid values are 0 to 99. The default is 4. The number represents the number of concurrent commands with response that can execute. Additionally, one non-response console will always be acquired.
**HILITE parameter**

The optional HILITE parameter indicates whether IMS messages, beginning with a `/`, should be highlighted in the BBI-SS PAS journal.

This parameter uses the following syntax:

```
HILITE={YES | NO}
```

The default is YES.

**JESCHAR parameter**

The optional JESCHAR parameter (used for MainView AutoOPERATOR for z/OS only) specifies the character to recognize as the JES2 command character.

This parameter uses the following syntax:

```
JESCHAR={x | $}
```

The default is $.

**JES3CHAR parameter**

The optional JES3CHAR parameter (used for MainView AutoOPERATOR for z/OS only) specifies the character to recognize as the JES3 command character.

This parameter uses the following syntax:

```
JES3CHAR={x | *}
```

The default is *.

---

**Note**

This parameter is ignored when JES3CMD=N is specified because the SVC34 interface is used.
**JES3CMD parameter**

The optional JES3CMD parameter (used for MainView AutoOPERATOR for z/OS only) specifies whether the subsystem interface should be used to issue JES3 commands.

This parameter uses the following syntax:

\[
\text{JES3CMD} = \{ Y \mid N \}
\]

The default is Y.

*Note*

This parameter should be set to N for JES 3 Release 2.2.1 and later.

---

**JESFLTR parameter**

The required JESFLTR parameter specifies what subsystem request functions to filter before JES processes a function.

This parameter uses the following syntax:

\[
\text{JESFLTR} = \{ \text{CMD} \mid \text{WTO}.[\text{NOSSI}] \}
\]

- If CMD is specified, Rules can suppress JES commands.
- If WTO is specified, messages to a JES3 console can be suppressed.

WTO activates the filter only if the BBI-SS PAS is started *after* JES. If the PAS is started before JES, you can use the `.T JESFLTR=WTO BBI` control command to activate the filter after JES initializes.

If you specify the WTO option, the NOSSI option is also available for use. When the NOSSI option is specified, any MSG rules that have an action of `DISPLAY AT DEST = NO` will suppress the WTO broadcast to most or all other subsystems in the SSI chain in addition to suppressing the WTO from the console.

*Note*

Normally all WTOs are broadcast to all subsystems and the NOSSI option can cause issues for any subsystems that depend on seeing the suppressed WTOs.

You can combine the values in parentheses, for example:

\[
\text{JESFLTR} = (\text{CMD}, \text{WTO})
\]
JESFLTR=(WTO,CMD,NOSSI)

For information about removing the JESFLTR setting from a PAS, see the *MainView AutoOPERATOR Customization Guide*.

**JESTYPE parameter**

The optional JESTYPE parameter specifies the type of JES that starts after the BBI-SS PAS is active.

This parameter uses the following syntax:

\[
\text{JESTYPE} = \{ \text{JES2} \mid \text{JES3} \}
\]

*Note*

- If JES2 is specified on a JES3 system, event types EOS, EOM, and TLM will not function correctly and will not initiate any rules of these types.
- JESTYPE is needed only when starting the BBI-SS PAS *before* JES is started.

**MCTSIZE parameter**

The optional MCTSIZE parameter specifies how many unique message identifiers to track for the Message Statistics application.

This parameter uses the following syntax:

\[
\text{MCTSIZE} = \{ nnnn \mid 510 \}
\]

The valid range is 0 to 9999. The default is 510. Specifying a value of zero (0) completely disables the gathering of message statistics.

**MIGCONS parameter**

The optional MIGCONS parameter (used for MainView AutoOPERATOR for z/OS only) represents the number of additional X-MCS consoles to be acquired by MainView AutoOPERATOR.

This parameter uses the following syntax:
MIGCONS=\{nn | 1\}

Valid values are 0 to 99. The default is 1. Migration IDs are one-byte console IDs. Non-sysplex users can force MainView AutoOPERATOR to use X-MCS consoles by using this parameter.

NETVIEW parameter

The required NETVIEW parameter (used for MainView AutoOPERATOR for Access NV only) specifies the one- to eight-character name of the NetView subsystem to which the MainView AutoOPERATOR subsystem is to connect.

This parameter uses the following syntax:

NETVIEW=\textit{name}

PROFUSED parameter

The optional PROFUSED parameter specifies the percentage to be used for the Profile Pool (BBIVARS) data set usage threshold. When the specified threshold is reached, message CS1023W is issued, stating that the threshold has been exceeded.

This parameter uses the following syntax:

PROFUSED=\{nn | 85\}

Valid values are 5 to 99. The default is 85.

XCFGROUP parameter

The optional XCFGROUP parameter specifies the one- to eight-character name of a cross-system coupling facility (XCF) group that this BBI-SS PAS is to join for the purpose of AOAnywhere communication.

This parameter uses the following syntax:

XCFGROUP=\{name | BMCAB\}

The default is BMCAB.
Note

- This parameter is used only for AOAnywhere communication; it is not a replacement for normal VTAM communication between PASs.

- The XCF group that you specify should be used only for BBI-SS PASs; otherwise, communication with other subsystems will be adversely affected.

- Only BBI-SS PASs that belong to the same XCF group and reside in the same sysplex can communicate with each other. To determine the XCF group to which a PAS belongs, issue the following BBI control command:

  .D XCF

BBISSPxx parameter for MainView AutoOPERATOR for WebSphere MQ and MainView for WebSphere MQ

The optional BBISSPxx GTS parameter (used for MainView AutoOPERATOR for WebSphere MQ and MainView for WebSphere MQ only) controls whether the Generic TCP/IP Server (GTS) is started during BBI-SS PAS initialization.

This parameter uses the following syntax:

$$GTS=\text{NO} | \text{nn} | \text{00}$$

- If NO is specified, the GTS is not started.

- If nn is specified, the GTS is started. The nn variable identifies the 2-character suffix of a BBPARM member (BBTCPnn) that should be used to start the GTS.

The default is 00, which means that BBTCP00 is used to start the GTS.
The required BBISSPxx ACONCICS parameter (used for MainView AutoOPERATOR for CICS and MainView for CICS only) determines the BBI-SS PAS to which CICS.

This parameter uses the following syntax:

ACONCICS={NO | YES | ONLY}

Do not change the default until you read the information about BBI-SS PAS and CICS target connections in the MainView for CICS Customization Guide.

The default is NO.

The optional BBISSPxx IMAGSTC parameter (used for MainView for CICS, MainView for DB2, MainView for DBCTL, and MainView for IMS Online only) specifies the one- to eight-character name of the started task that is used to print the image log.

This parameter uses the following syntax:

IMAGSTC={xxxxxxxx | BBIILOG}

The default is BBIILOG.

BBIVARxx parameters

BBPARM member BBIVARxx contains any number of variables that are to be set at BBI-SS PAS startup when VPOOL=RESET is specified with the startup parameters.

This parameter uses the following syntax:
**Note**

- VPOOL=RESET requires a COLD start.

- Member BBIVARxx supports the substitution of system variables.

- These SHARED variables are set before the MainView AutoOPERATOR Rules processor is initialized. Therefore, Rules for the event type VAR will not match when these SHARED variables are initialized. However, these variables are available to be used in all other Rules.

You can create more than one BBIVARxx member (where xx is the suffix for member).

**Related Information**

- “System variables substitution in parameter library members” on page 233

### BBIVARxx processing rules

The following rules apply to determine which BBIVARxx member is processed:

- You can request multiple BBIVARxx members with the BBPARM member BBICFG.
  
  The name of the member is constructed from the literal BBIVAR and a suffix. The default suffix is 00. An alternative suffix member can be specified through configuration of the BBI-SS PAS with a BBCFG DD statement in the PAS JCL:
  
  — If a BBCFG DD statement is present and it contains a valid BBIVAR=xx statement, the BBIVAR=xx statement is processed.
  
  — If no valid BBIVAR=xx statement is found in the BBCFG configuration member, a suffix of 00 is assumed.
  
  — An example of specifying a specific BBIVARxx member in the BBCFG member is BBIVAR=(99), where 99 is the suffix for BBPARM member BBIVAR99.

  To specify multiple BBIVARxx members, use BBIVAR=(99,23,12,15), where 99, 23, 12 and 15 are suffixes for BBPARM members BBIVAR99, BBIVAR12, BBIVAR12 and BBIVAR15.
When the BBCFG configuration member has multiple BBIVAR statements specified, the system starts with the last one. For example, if the BBCFG member contains the following statements, the system starts with $BBIVAR=(99,12,13)$:

$BBIVAR=(99,12)$
$BBIVAR=(99,12,13)$

■ Error or warning messages are issued for the following possible situations:

— BBCFG contains a valid BBIVAR=xx statement but the member was not found.
— BBCFG contains syntax errors. Syntax errors in the configuration member might cause the default BBIVAR00 to be processed.
— The BBIVARxx member contains syntax errors.

**BBIVARxx syntax rules**

The following syntax rules apply to processing within the BBIVARxx member:

■ Any number of entries can be specified.
■ Each line is restricted to 72 characters, and entries cannot be continued over multiple lines.
■ Variables names can have a maximum length of 32 characters.
■ An asterisk in column one indicates a comment line and causes the line to be ignored.
■ An assignment statement can be in either of the following formats:
  
  VARIABLE = 'VALUE'
  VARIABLE = value

  The first format must be used if the variable value contains blanks; otherwise, blank spaces are ignored. An example follows:

  MYVAR = 'A B C' (MYVAR is set to A B C)
  MYVAR = A B C (MYVAR is set to A)

■ Variable names are always translated to uppercase letters.
■ Variable names must conform to TSO/E IBM REXX syntax.
Variables are created in the variable pool exactly as specified. REXX stem variables do not experience any form of substitution. For example, the following specification creates a variable called A and a variable called I.A, not I.0:

\[
\begin{align*}
A &= 0 \\
I.A &= 5
\end{align*}
\]

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 variables is to contain double quotation marks, enclose the string in single quotation marks (').

If any of the preceding conditions are not met, an error or warning message is issued, accompanied by an additional message that identifies the invalid line. Invalid entries are skipped, and processing continues.

**Example**

Valid entries are as follows:

\[
\begin{align*}
TEST1 &= 25 \\
test2 &= 15 \\
end_of_day &= \text{TRUE} \\
COMMENT &= \text{\'SHIFT CYCLE\'} \\
TEST1.A &= 5
\end{align*}
\]

Invalid entries are as follows:

\[
\begin{align*}
\text{\'TEST1\'} &= 25 \\
TEST1* &= 255 \\
COMMENT &= \text{\'SHIFT CYCLE} (\text{End quotation mark is missing.}) \\
QSMID &= \text{SYSB} (\text{Q-variables are reserved and read-only.})
\end{align*}
\]

**BBTCP00 parameters**

The BBTCP00 member contains parameters that control whether a BBI-SS PAS supports the BMC TCP/IP interface.

The BBTCP00 member is copied from the BBSAMP library to the BBPARM library and modified for use by the following products:

- MainView AutoOPERATOR
- MainView AutoOPERATOR *for WebSphere MQ*
MainView for WebSphere MQ

MainView host server

The following parameters are used in the BBTTCP00 member:

- TCPNAME
- INFOMSG
- DEBUG

**Related Information**

- "TCPNAME parameter" on page 229
- "INFOMSG parameter" on page 229
- "DEBUG parameter" on page 230

---

**TCPNAME parameter**

The required TCPNAME parameter specifies the name of the TCP/IP address space.

This parameter uses the following syntax:

\[ \text{TCPNAME} = \text{name} \]

For TCP/IP Release 3.2 and later, you can omit the TCPNAME parameter and the BBI-SS PAS will attempt to determine which TCP/IP address space to use. If more than one address space is found, the first address space that is found with the latest release is used.

**INFOMSG parameter**

The optional INFOMSG parameter determines whether TCP/IP interface startup messages are displayed on the console.

This parameter uses the following syntax:

\[ \text{INFOMSG} = \{ \text{YES} \mid \text{NO} \} \]
The default is YES.

**DEBUG parameter**

The optional DEBUG parameter activates debug mode and writes diagnostic messages to the operator console.

This parameter uses the following syntax:

\[
\text{DEBUG=\{ALL | ERROR | PARM | } \text{xxxxxx} \}
\]

---

**Note**

Specify this parameter only at the request of BMC Customer Support because it causes many messages to be written to the operator console.

---

You can specify multiple DEBUG statements, but the statements must not conflict with each other.

- If ALL is specified, a message is produced for each socket call.
- If ERROR is specified, messages are produced only for unsuccessful socket calls.
- If PARM is specified, a message is produced with parameters 4 through 9 for each socket call.
- \text{xxxxxx} produces messages only for the specified task ID. Each MainView product has one or more task IDs. The main TCP/IP task ID is Global. For product-specific task IDs, see your MainView product documentation.

Debug messages are documented in the MainView online message system. All debug messages have a suffix of D. Some MainView products might display these messages even when debug mode is not active. Those messages will have a suffix of I or E. For those products, if debug mode is activated, some messages will be issued twice, once with the I or E suffix, and once with the D suffix.

**BBPROF data set members**

The BBPROF data set contains profile information and cycle refresh definitions for a terminal session.

Other BBPROF members are created dynamically. These members should not be edited or deleted because they contain important profile information for BBI-SS PAS applications.
The following topics describe the BBPROF members that affect BBI-SS PAS operation:

**Related Information**

- “xxxCYC or CYCnn member” on page 231
- “BBICMDS member” on page 231
- “BBIPFK member” on page 232
- “BBITSP00 member” on page 232
- “LOGPFK member” on page 232
- “xxxPFK member” on page 233

### xxxCYC or CYCnn member

The xxxCYC members (where xxx is an application prefix) contain sample definitions of cycles for refreshing services.

The CYCnn members can be added to BBPROF by using a naming convention of CYCnn or any valid member name. (The definition is activated with the full member name.)

A CYCnn definition can be activated in the CYCLE application with a SET nn command, where nn specifies an alphanumeric suffix.

### BBICMDS member

The BBICMDS member contains the BBI control command table.

For a description of the BBI control commands, see the *MainView Administration Guide*. 
**BBIPFK member**

The BBIPFK member contains default program function (PF) key definitions that are used if a xxxxPFK member is not defined for a given product application (for example, IMSPFK).

**BBITSP00 member**

The BBITSP00 member contains profile information for a terminal session.

BBITSP00 defines the primary product application (PRIMAPPL) to display when the terminal session starts if multiple application types, target names, and associated menus are specified as follows:

\[ \text{TGRT=\text{tgtn}} \text{, \text{TYPE}=\text{appltype}} [, \text{PRIMMENU}=	ext{name}] \]

The application type can be AO, IMS, CICS, or DB2, depending on the product that is installed.

Other parameters define:

- Whether local logging (terminal session journaling) is performed (LOGGING=YES)
  
  The LOGGING parameter is set to NO if BBIJRNL1 DD is not allocated in the TSCLIST (for example, JOURNAL(YES) in TSCLIST).
- The default screen refresh interval (INTERVAL)
- Screen characteristics (SCREEN)
- Characters (CAPS)
- Colors (COLORS)
- Reverse video (REVERSE)
- Formatted dump output (DUMPS and DUMPCLAS)
- ISPF split-screen (ISPF)

**LOGPFK member**

The LOGPFK member contains default program function key definitions for the General Services LOG DISPLAY option.
xxxPFK member

The xxxPFK member contains program function key definitions that are application specific. The member name prefix (xxxx) can be CICS, DB2, IMS, or MVS.

The PF key definitions are displayed when the KEYS option is selected from the Primary Option Menu or when the KEYS command is entered. For more information, see the MainView User Guide.

If the member contains the MASTER keyword, the online changes cannot be stored in the xxxxPFK member when the SAVE command is entered from the COMMAND line.

If the MASTER keyword is removed and a user BBPROF data set does not exist, subsequent online PF key changes are saved in the PF key member of the site's BBPROF data set.

System variables substitution in parameter library members

The following MainView parameter library members support system variables (or symbolic) substitution:

- BBIISPxx
- BBIJNT00
- BBINOD00
- BBISSPxx
- BBIVARxx
- BBIXSP00

With symbolic substitution, you can reduce the number of parameter libraries that you maintain. When the member is read, the value of the symbol is substituted for the symbol. The symbol must begin with an ampersand followed by one to eight characters and, optionally, a period. You can define your own symbols by placing them in the IBM PARMLIB member IEASYMxx.
**Symbolic substitution example**

Symbolic names can be concatenated with a string, at the beginning, middle, or end of the string, as shown in the following example:

\[ DSN=VAM3.&SYSPLEX..&SYSNAME..OFFLOAD \]

In the example, if &SYSPLEX=BMC1 and &SYSNAME=SJSD, the string would convert to the following statement:

\[ DSN=VAM3.BMC1.SJSD.OFFLOAD \]

**Note**

A period at the end of a symbolic name is assumed to be a delimiter and is omitted in the substitution. If you want to generate the string BMC1SJSD, you would specify &SYSPLEX&SYSNAME.

For more information about symbolic substitution, refer to the discussion of system symbols and symbolic substitution in the *IBM Initialization and Tuning Reference*.

**Controlling symbolic substitution**

You can control symbolic substitution on a member-by-member basis by adding a PROCESS statement control card to the beginning of each member.

The statement uses the following syntax:

\[ \text{PROCESS \ [SYM=\{YES | NO\}, \ \{LIST=\{YES | NO\}\]} \]

If present, the PROCESS statement must be the first statement in the member and must begin in column 1. The keywords must be separated by a comma. An example follows:

\[ \text{PROCESS SYM=\text{YES}, LIST=\text{NO}} \]

- **SYM** indicates whether symbolic substitution is allowed:
  - YES (the default) indicates that symbols are to be replaced by their value, if the calling program has enabled symbol substitution.
  - NO indicates that symbolic substitution should be suppressed for this member, even if the calling program has enabled substitution.

Substitution will not be performed on comments or anything beyond column 71.
LIST indicates whether the contents of the member being processed should be written to the job log (displayed as WTO messages), one statement at a time as the member is read:

— YES causes the entire contents of the member, except the PROCESS statement, to be written to the job log.

— NO (the default) indicates that the contents of the member should not be written to the job log.

If substitution is performed, two WTO messages are displayed for each statement on which substitution was performed. The first message contains the statement with the symbol, and the second message contains the same statement with the substituted value of the symbol. The second statement is also prefixed with <> to indicate that the statement was modified.
MainView Logger library members

BBSAMP members MVLOGGER and LSLOGSTC contain statements for customizing or creating started task procedures. MVLOGGER is the MainView Logger primary logspace, and LSLOGSTC is a product-specific logspace.

MainView Logger started task procedure statements

The following table lists the MainView Logger started task procedure statements.
### Table 9: MVLOGGER started task procedure statements

<table>
<thead>
<tr>
<th>JCL statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>// procname PROC</code></td>
<td>The PROC statement parameters are as follows:</td>
</tr>
<tr>
<td></td>
<td>■ `SYSP={xx</td>
</tr>
<tr>
<td></td>
<td>At least one suffix must be specified. The MVLOGGER PROC statement is distributed with SYSP=00, which calls parameter member LOGGER00.</td>
</tr>
<tr>
<td></td>
<td>If you specify more than one suffix, each LOGGERxx member is associated with a single task, or logspace, within the MainView Logger address space. Logspaces are started serially, in the order specified on the SYSP parameter. The first logspace with a LOGSPACE name that is equal to the LOGSSID (or with no LOGSPACE name at all) is considered the primary logspace.</td>
</tr>
<tr>
<td></td>
<td>All other logspaces are considered product-specific logspaces.</td>
</tr>
<tr>
<td></td>
<td>The SYSP value can be overridden on a START command from the console. For example, to start MainView Logger manually with the parameters in member LOGGER01, you would specify:</td>
</tr>
<tr>
<td></td>
<td><code>S MVLOGGER,SYSP=01</code></td>
</tr>
<tr>
<td></td>
<td>■ NEWREG enables the NEWREG parameter, which can be used on a START command to allocate a new register data set. NEWREG should have a null value on the PROC statement.</td>
</tr>
<tr>
<td></td>
<td>■ RESET enables the RESET parameter, which can be used on a START command to reinitialize the data-in-virtual (DIV) data set and start a new log file. RESET should have a null value on the PROC statement.</td>
</tr>
<tr>
<td></td>
<td>■ RESETA enables the RESETA parameter, which can be used on a START command to force an emergency restart. RESETA should have a null value on the PROC statement.</td>
</tr>
</tbody>
</table>
### MainView Logger initialization parameters

BBPARM members LOGGER00 and LSLOGGER contain the initialization parameters for creating the following MainView Logger logspaces:

- **LOGGER00** for the MainView Logger primary logspace
- **LSLOGGER** for a product-specific logspace

The following topics explain the parameters LOGGER00 and LSLOGGER members use:

<table>
<thead>
<tr>
<th>JCL statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>//MVL1 EXEC</td>
<td>This statement specifies the MainView Logger program name (PGM=BBM9ZL00). The PARM subparameters are all symbolic and use the same values that were defined for the corresponding parameter in the PROC statement.</td>
</tr>
</tbody>
</table>
| //STEPLIB DD        | This statement defines a partitioned data set that contains the MainView Logger load modules. The partitioned data set must be an authorized library, such as hlq.BBLINK.  
  **Note:** This statement should be deleted if BBLINK was added to your system link list concatenation. |
| //BBIPARM DD        | This statement identifies the user parameter library, such as hlq.UBBPARM.                                                                      |
MVSSYSTEM parameter

The required MVSSYSTEM parameter specifies the system to which all subsequent parameters apply.

This parameter uses the following syntax:
MVSSYSTEM=sysname

MainView Logger reads the parameters between this MVSSYSTEM statement and the next MVSSYSTEM statement, or to the end of the LOGGERxx member.

The sysname value must match the current system name in CVTSNAME. Alternatively, you can specify the system variable &SYSNAME, which enables a single set of parameters to start MainView Logger on multiple system images. You can also use &SYSNAME to define default parameters, and then specify additional MVSSYSTEM statements in the same member to define different parameters for certain systems.

LOGCYLINDERS or LOGTRACKS parameter

The required LOGCYLINDERS or LOGTRACKS parameter specifies the size of the allocated log file data sets.

These parameters use the following syntax:

LOGCYLINDERS=nn

LOGTRACKS=nnn

Note
Use either LOGCYLINDERS or LOGTRACKS, but not both.

Valid values per logspace are 5 to 50 cylinders or 75 to 750 tracks. A product-specific logspace can use 5 to 150 cylinders.

Depending on your logger activity, you can adjust this value so that the recommended one to three log file data sets are created in a single day. Up to 4080 log file data sets can be created in a single day. If the maximum number of log file data sets is exceeded, a message is issued and MainView Logger shuts down. You will have to manually restart MainView Logger.

This parameter can be temporarily modified dynamically for an active MainView Logger. To change the value, issue one of the following MODIFY commands:

F mvloggerid,LOGCYLINDERS=nn

F mvloggerid,LOGTRACKS=nnn
**LOGPREFIX parameter**

The required LOGPREFIX parameter specifies the data set prefix for the allocated log file data sets.

This parameter uses the following syntax:

```
LOGPREFIX=prefix
```

MainView Logger supplies the low-level qualifier in the form of `.xyyddnn`, where `x` is an alphabetic qualifier of A through I and S through Y; `yy` is the year; `ddd` is the Julian day; and `nn` is a hexadecimal numerical value from 01 to FF.

This parameter can be modified dynamically for an active MainView Logger. To change the value, issue the following MODIFY command:

```
F mvloggerid,LOGPREFIX=prefix
```  

**REGISTERDSN parameter**

The required REGISTERDSN parameter identifies the register data set that you allocated to hold log file data set names and statistics.

This parameter uses the following syntax:

```
REGISTERDSN=dsname
```

The contents of this data set can be displayed in the LOGLIST view. This data set must be unique for each instance of MainView Logger and must be allocated prior to starting the logger task. See Verification of MainView Logger data sets on page 85.

**DIVDSN parameter**

The required DIVDSN parameter identifies the data-in-virtual (DIV) data set that you allocated for buffer log requests and for writing log records to DASD.

This parameter uses the following syntax:

```
DIVDSN=dsname
```

This data set must be unique for each instance of MainView Logger and must be allocated prior to starting the logger task. See Verification of MainView Logger data sets on page 85.
**COMPRESSION parameter**

The optional COMPRESSION parameter specifies whether data compression should be used for the data portion of records that are written to MainView Logger log files.

This parameter uses the following syntax:

```
COMPRESSION={NO \textbf{ | } YES}
```

If data compression is turned on, approximately 20 percent less space is used. The default is YES.

This parameter can be modified dynamically for an active MainView Logger. To change the value, issue the following MODIFY command:

```
F mvloggerid,COMPRESSION={NO \textbf{ | } YES}
```

**DEFAULTLOGGER parameter**

The optional DEFAULTLOGGER parameter (ignored for product-specific logspaces) specifies whether this instance of MainView Logger is considered the default logger (if multiple instances of the logger are running on a system image).

This parameter uses the following syntax:

```
DEFAULTLOGGER={NO \textbf{ | } YES}
```

Only one default MainView Logger can be specified per system image. The default is YES.

\textit{Note}

Only the primary logspace can be considered the default logger.

**DIVBLOCKS parameter**

The optional DIVBLOCKS parameter specifies the number of blocks to be used by the DIV data set.

This parameter uses the following syntax:

```
DIVBLOCKS={n\text{nnn} \textbf{ | } 180}
```
This value should normally be the same as the number of blocks that are allocated for the data set. The default is 180 (1 cylinder on a 3390 equals 180 blocks). The maximum value is 9000 blocks.

**Note**
For product-specific logspaces, use the value that is recommended by the BMC product that is using the logspace.

**DUMPCOUNTMAX parameter**

The optional DUMPCOUNTMAX parameter specifies the maximum number of SVC dumps that MainView Logger can take during any one execution.

This parameter uses the following syntax:

\[
\text{DUMPCOUNTMAX} = \{ n \mid 1 \}
\]

Valid values are 0 to 9. The default is 1.

This value is decremented each time a dump is taken. When MainView Logger is restarted, the value is reset to the value specified in LOGGERxx.

This parameter can be modified dynamically for an active MainView Logger. To change the value, issue the following MODIFY command:

\[
\text{F \_\_\_\_loggerid, DUMPCOUNTMAX=\_\_\_\_n}
\]

**LOGSSID parameter**

The optional LOGSSID parameter specifies the one- to four-character subsystem ID that is to be used by the MainView Logger started task procedure.

This parameter uses the following syntax:

\[
\text{LOGSSID} = \{ ssid \mid \text{MVL1} \}
\]

The default is MVL1.
**LOGSPACE parameter**

The optional LOGSPACE parameter specifies a three- to eight-character name for the primary logspace to which this LOGGERxx member applies. Secondary LOGSPACE names are five- to eight-characters in length.

This parameter uses the following syntax:

```
LOGSPACE=xxxxxxxx
```

LOGSPACE can be allowed to default to the LOGSSID value. If more than one logspace is identified on the SYSP parameter of the PROC statement, the first logspace with a LOGSPACE name that is equal to the LOGSSID (or with no LOGSPACE name at all) is considered the primary logspace. The primary logspace is:

- The only logspace that can capture WTO messages (WTOLOG=Y)
- The default target for many of the MainView Logger MODIFY commands, if no `target` parameter is specified

**MANAGEMENTCLASS parameter**

The optional MANAGEMENTCLASS parameter specifies the one- to eight-character management class value to be used when allocating log files.

This parameter uses the following syntax:

```
MANAGEMENTCLASS=xxxxxxxx
```

If you specify a MANAGEMENTCLASS value, you must also specify a STORAGECLASS value; otherwise, MANAGEMENTCLASS is ignored.

---

**Note**

If you specify a blank (MANAGEMENTCLASS= ), no management class is used on dynamic allocation.

This parameter can be modified dynamically for an active MainView Logger. To change the value, issue the following MODIFY command:

```
F mvloggerid,MANAGEMENTCLASS=xxxxxxxx
```
MAXPERSEC parameter

The optional MAXPERSEC parameter specifies the number of log records per second that, if sustained by any given client or ASID for one minute, would cause the client's MainView Logger log requests to be temporarily suppressed.

This parameter uses the following syntax:

\[ \text{MAXPERSEC} = \{nn \mid 3 \mid 0\} \]

Suppression is invoked until the log rate drops below this value and remains there for about 8 seconds.

The rate is a numeric value from 0 to 99 records per second. The default value for the primary logspace is 3. For product-specific logspaces, the default is 0, which means that log records are never suppressed.

This parameter can be modified dynamically for an active MainView Logger. To change the value, issue the following MODIFY command:

\[ F \text{mvloggerid,MAXPERSEC}=nn \]

MSGFLT parameter

The optional MSGFLT parameter (ignored for product-specific logspaces) specifies the two-digit alphanumeric suffix of an optional message filter member.

This parameter uses the following syntax:

\[ \text{MSGFLT}=xx \]

A message filter member (MSGFLTxx) is used to filter out WTO messages that are written to the MainView Logger log files. WTO messages that match one of the prefixes in the MSGFLTxx member are not logged by the WTO processor.

MULTIREC parameter

The optional MULTIREC parameter (ignored for product-specific logspaces) specifies whether multiline WTO messages should be combined into one MainView Logger record.

This parameter uses the following syntax:
MULTIREC={NO | YES}

MULTIREC=YES (the default) saves log file space by combining multiline WTOs into a single log record. Once combined, however, the complete WTO message might not be visible in the LOGMSG view. To view the complete WTO message, select the record and display the Log Message Detail dialog.

If you specify MULTIREC=NO, each WTO record produces a separate log record and a separate line in the LOGMSG view.

**Note**
If WTOLOG=NO is specified (or defaulted to), the MULTIREC parameter is ignored.

This parameter can be modified dynamically for an active MainView Logger. To change the value, issue the following MODIFY command:

F mvloggerid,MULTIREC={NO | YES}

**STORAGECLASS parameter**

The optional STORAGECLASS parameter specifies the one- to eight-character storage class value to be used when allocating log files.

This parameter uses the following syntax:

STORAGECLASS=xxxxxxxx

**Note**
If you specify a blank (STORAGECLASS= ), no STORCLAS value is used on dynamic allocation.

This parameter can be modified dynamically for an active MainView Logger. To change the value, issue the following MODIFY command:

F mvloggerid,STORAGECLASS=xxxxxxxx

**VOLSER parameter**

The optional VOLSER parameter specifies a one- to six-character volume serial number to be used when allocating log files.

This parameter uses the following syntax:
VOLSER=xxxxxx

If allocation fails with the specified VOLSER number, MainView Logger tries to allocate log files by using the STORAGECLASS value. If STORAGECLASS is not specified, or that value also fails, log files are allocated without a specific VOLSER or STORAGECLASS value.

**WTOLOG or LOGWTO parameter**

The optional WTOLOG or LOGWTO parameter (ignored for product-specific logspaces) specifies whether MainView Logger should activate the subsystem interface to capture WTO and WTOR messages and write them to the log files.

These parameters use the following syntax:

- WTOLOG={YES | NO}
- LOGWTO={YES | NO}

This parameter can be modified dynamically for an active MainView Logger. To change the value, issue the following MODIFY command:

```fortran
F mvloggerid,WTOLOG={YES | NO}
```

**Note**

LOGWTO is an alias for WTOLOG. You can specify either parameter name.
MainView Alarm Manager version 2.1 PAS library member

BBSAMP member BBHPAS contains statements for customizing or creating a MainView Alarm Manager version 2.1 PAS startup procedure in your system started task procedure library.

Table 10 on page 249 describes the MainView Alarm Manager version 2.1 PAS startup JCL statements. These statements can be found in BBSAMP member BBHPAS.

Table 11 on page 251 describes the parameters for the MainView Alarm Manager version 2.1 PAS PROC statement.

Tip
MainView Alarm Management (the new version of MainView Alarm Manager) is integrated into the CAS. Alarm Management no longer requires its own PAS. For more information, see the MainView Alarm Management Guide.

Table 10: MainView Alarm Manager version 2.1 PAS started task procedure statements

<table>
<thead>
<tr>
<th>JCL statement</th>
<th>Description</th>
</tr>
</thead>
</table>
| // procname PROC  | This statement identifies the started task procedure and its parameters. See Table 11 on page 251 for an explanation of each PROC statement parameter.  

**Note:** You can override these parameters with the START command for the PAS, as described in the MainView Administration Guide.  |
| //PAS EXEC        | This statement specifies the program name as PGM=SSLOAD.  
The PARM subparameters are all symbolic and use the same values that were defined for the corresponding parameter in the PROC statement.  
For a detailed explanation of each parameter, see Table 11 on page 251. |
<table>
<thead>
<tr>
<th>JCL statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>//STEPLIB DD</td>
<td>This statement defines a partitioned data set that contains the MainView Alarm Manager version 2.1 load modules. The partitioned data set must be an authorized library, such as hlq.BBLINK. <strong>Note:</strong> This statement should be deleted if BBLINK was added to your system link list concatenation.</td>
</tr>
<tr>
<td>//BBACTDEF DD</td>
<td>This statement defines the library containing all action and view tables for MainView Alarm Manager version 2.1.</td>
</tr>
<tr>
<td>//BBIPARM DD</td>
<td>This statement contains the BBHTMN.xx members, where xx is the ID for the group. Each member contains information for all alarm definitions in the group.</td>
</tr>
<tr>
<td>//BBSECURE DD</td>
<td>This statement defines the data set that contains MainView Alarm Manager version 2.1 security resource definitions. These definitions are used by the CAS and PAS to determine whether security checking should occur for access to systems, products, views, and actions. The same data set name must be defined for the CAS started task procedure BBSECURE statement and the PAS started task procedure BBSECURE statement. <strong>Note:</strong> Defining a BBSECURE data set is only one part of setting up security to function with your external security manager (ESM). For more information, see the MainView Security Guide.</td>
</tr>
<tr>
<td>//BBVDEF DD</td>
<td>This statement defines the view library that contains all default views for MainView Alarm Manager version 2.1.</td>
</tr>
<tr>
<td>//BBIJRNL1 DD</td>
<td>This statement defines the journal log data sets that MainView Alarm Manager version 2.1 uses to record:</td>
</tr>
<tr>
<td>//BBIJRNL2 DD</td>
<td>■ Commands and responses that are issued by the Alert Management component</td>
</tr>
<tr>
<td></td>
<td>■ Product alerts, warnings, and messages</td>
</tr>
<tr>
<td>//BBIDIV DD</td>
<td>This statement defines the data-in-virtual (DIV) data set that is used to store binary large objects and data across MainView Alarm Manager version 2.1 restarts.</td>
</tr>
</tbody>
</table>
Table 11: MainView Alarm Manager version 2.1 PAS PROC parameters

<table>
<thead>
<tr>
<th>Proc parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVALSS=ssid</td>
<td>This parameter specifies a one- to four-character subsystem ID for the MainView Alarm Manager version 2.1 started task procedure. This ID must be unique within the MVS image where MainView Alarm Manager version 2.1 will be started.</td>
</tr>
<tr>
<td>START=WARM</td>
<td>COLD</td>
</tr>
</tbody>
</table>
| MVAJRNL=dsname | This parameter identifies the MainView Alarm Manager version 2.1 journal log data sets. The data set names are derived by adding a suffix of 1 or 2 to the value of this parameter.  
**Example:** If you specify MVAJRNL='BMC.MVAJRNL', the started task procedure looks for two data sets called ‘BMC.MVAJRNL1’ and ‘BMC.MVAJRNL2’.  
**Note:** The data sets that you identify must be allocated by using the BBIJRNL1 and BBIJRNL2 DD statements in the PAS startup JCL. |
| MVADIV=dsname | This parameter identifies a data set to be used by MainView Alarm Manager version 2.1 for storing data across product restarts.  
**Note:** The data set that you identify must be allocated by using the BBIDIV DD statement in the PAS startup JCL. |
| BBLINK= | *(optional)* This parameter defines the name to be assigned to BBLINK.  
**Note:** If BBLINK is in your system link list, this parameter and the STEPLIB DD statement should be removed. |
<p>| BBCHILV= | This parameter defines the high-level qualifier for the product libraries. |</p>
<table>
<thead>
<tr>
<th>Proc parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVAPARM=</td>
<td>This parameter defines the name to be assigned to the MainView Alarm Manager version 2.1 user parameter library.</td>
</tr>
</tbody>
</table>
Migration of MainView Infrastructure and MainView products

This appendix contains procedures for migrating to a new version of MainView Infrastructure and for migrating MainView products to a new system image.

Migrating to a new version of MainView Infrastructure

Use the following procedures to migrate to a new version of MainView Infrastructure.

Version 6.0.00 is the current version of MainView Infrastructure.

This topic applies to users who:

- Have MainView products that are running earlier versions of MainView Infrastructure
- Are migrating to MainView Infrastructure version 6.0.00

The MainView Infrastructure version that is installed at your site is indicated by \( V v.r.mm \) in a CAS startup message, where \( v \) is the version number, \( r \) is the release number, and \( mm \) is the modification level:

```
BMZA001I CAS(SYSB) SSID(BBCS) INITIALIZATION COMPLETE - Vv.r.mm
(BPY xxxx)
```
Migrating CAS definitions

When you upgrade to a new version of MainView Infrastructure, you can migrate your existing CAS definitions to the new system.

CAS definitions are stored in a member of the UBBPARM data set. Copying the CAS definitions forward enables the new CAS to know about your existing CASs.

**Note**

During the installation and customization process, you select whether to use runtime libraries or SMP/E libraries. If you use runtime libraries, your customized library names will start with BMC. If you use SMP/E libraries, your customized library names will start with BB, UBB, or TOSZ. A few library names start with IBB or SBB for runtime or SMP/E libraries.

Throughout the MainView documentation, the BB*, UBB*, and TOSZ* data set names are used even if the actual library names start with BMC*.

**Related Information**

- “MainView product libraries” on page 21
- “Migrating BBMTYB00 (CAS definitions)” on page 84

Running mixed versions of MainView Infrastructure

For products that run with MainView Infrastructure, the user address space (UAS)—where the terminal session resides—and the product address space (PAS) must be running the same version as the CAS to which they connect.

Mixed versions of MainView Infrastructure can run in a sysplex. If the version on one target system is different from the version on another system, some functions on the different systems might not be available.

Each new MainView product version provides its own commands and features plus the commands and features of earlier versions. The earlier versions, however, do not support all of the commands and features of the latest version.

This topic describes how to run mixed versions of MainView Infrastructure in a sysplex, either on different system images or on the same system image.
Note
During the installation and customization process, you select whether to use runtime libraries or SMP/E libraries. If you use runtime libraries, your customized library names will start with BMC. If you use SMP/E libraries, your customized library names will start with BB, UBB, or TOSZ. A few library names start with IBB or SBB for runtime or SMP/E libraries.
Throughout the MainView documentation, the BB*, UBB*, and TOSZ* data set names are used even if the actual library names start with BMC*.

Related Information

■ “MainView product libraries” on page 21

Running mixed versions on different system images

Use the following procedure to migrate to a new version of MainView Infrastructure on one system image at a time. You will then have mixed versions of the infrastructure on different systems in a sysplex.

To implement a new version on one system at a time

1 Shut down the existing CAS and all connected PASs on the system that is being migrated.

2 (optional) Copy member BBMTYB00 from your existing UBBPARM data set to the new UBBPARM data set that was allocated during the installation of MainView Infrastructure.

This step makes all of your existing CAS definitions available to the new CAS that you are about to start.

3 Start the CAS that is running the new version of MainView Infrastructure.

The following messages are issued at CAS startup:

```
BBMZA096I CAS(ssid) IS VERSION v.r.mm (modid) AND PRIOR START WAS VERSION v.r.mm (modid)
BBMZA098I CAS(ssid) AUTOMATIC COLD START IN PROGRESS FOR THIS STARTUP
```

These informational messages indicate that the new CAS is at a different release level than the previous CAS. The messages are expected because you are migrating from one version of MainView Infrastructure to another version.
Note

The new CAS attempts to establish communication with any remote CASs that are defined to it, even though those CASs are not yet running the new version of MainView Infrastructure. The new CAS will also automatically connect to any CAS that it can locate through XCF within the SYSPLEX. If you do not want the new CAS to connect to the remote CASs, you must not provide a BBMTYB00 member and you must temporarily override the XCF group name by using the DFLTGRP parameter at CAS startup.

4 Start one or more PASs that are running the new version of MainView Infrastructure.

A message is issued to indicate that a PAS successfully connected to the new CAS.

Example

BBDD10291I MVS PAS(cas_ssid) INITIALIZATION COMPLETE -(modid)

5 Start one or more UASs that are running the new version of MainView Infrastructure.

If you attempt to connect a UAS that is running one version of MainView Infrastructure to a CAS that is running a different version, the following messages are issued:

BBMZA1031I CAS(ssid) IS VERSION v.r.mm (modid) AND CLIST VERSION IS v.r.mm (modid)

BBMZA1021I SELECT A CAS THAT MATCHES YOUR CLIST RELEASE OR EXIT CLIST AND RETRY

These messages indicate that the CLIST that is being used for the terminal session tried to connect to a CAS that is running a version of MainView Infrastructure that is different from the CLIST libraries. Remember that the UAS must be running the same version as the CAS to which it connects.

Running mixed versions on the same system image

If you are testing new MainView product releases on the same system image as existing products, you might have mixed versions of MainView Infrastructure on the same system. Use the following procedure to implement a new version on the same system as a previous version.

Ideally, a separate system image should be used for testing. Sometimes, however, you might want to test a new version on a system that currently runs a CAS, and you might want to keep that CAS available to end users. You can perform this task without disturbing your current environment by defining a separate CAS, PAS, and terminal session for each version of MainView Infrastructure.
If you have a single system image with mixed versions of MainView Infrastructure for testing and production, the following conditions must be met:

- The test system and the production system must have different terminal sessions, PASs, and CASs.
- Each CAS must have a unique subsystem ID (SSID).
- The MainView CLIST from the previous version of MainView Infrastructure and the current MainView CLIST must have different names.

**To implement a new version on the same system as a previous version**

1. *(optional)* Copy member BBMTYB00 from your existing UBBPARM data set to the new UBBPARM data set that was allocated during the installation of MainView Infrastructure.

   This step makes all of your existing CAS definitions available to the new CAS that you are about to start.

2. Start the CAS that is running the new version of MainView Infrastructure.

   Be sure to specify:

   - A unique CAS name on the SSID parameter
     By default, the CAS name is eight characters long, consisting of the system name followed by the SSID.
   - A unique XCF group name by using the DFLTGRP parameter

   For complete information about the CAS startup parameters, see the *MainView Administration Guide*.

3. Start one or more PASs that are running the new version of MainView Infrastructure.

   A message is issued to indicate that a PAS successfully connected to the new CAS.

   **Example**
   
   BBDDIO291 MVS PAS(cas_ssid) INITIALIZATION COMPLETE -(modid)

4. Start one or more UASs that are running the new version of MainView Infrastructure.

   The CLIST for the new version and the CLISTs from previous versions of MainView Infrastructure must have different names and allocate different user...
libraries. To access products at different version levels on the same system, you can exit and reenter ISPF and then run the appropriate MainView CLIST.

5 Use the Plex Manager CASDEF view to add one of the following identifiers to the definition of the CAS that you just started:

- A VTAM APPLID for VTAM communications
- A TCP/IP address and port number for TCP/IP communications

After you stop and restart the new CAS, the new communication parameters will be used.

Migrating MainView products to a new system image

This topic provides guidelines for migrating MainView products from one system image, or LPAR, to another system image.

For example, you might want to migrate a new release of MainView Infrastructure or some other MainView product from a test system to a production system.

Migrating MainView products involves:

- Migrating the MainView environment (see Migrating the MainView environment on page 258)
- Migrating specific products (see Migrating specific products on page 264.)

Migrating the MainView environment

Use the following procedure to migrate the MainView environment to a new system image.

How you migrate the MainView environment depends on how you customized the MainView environment during the product installation process.

If you customized the MainView environment to execute from runtime data sets, see “To use RTE to migrate the MainView environment to a new system image” on page 259. If you customized to execute from SMP/E target-zone data sets, see “To manually migrate the MainView environment to a new system image” on page 259.
During the installation and customization process, you select whether to use runtime libraries or SMP/E libraries. If you use runtime libraries, your customized library names will start with BMC. If you use SMP/E libraries, your customized library names will start with BB, UBB, or TOSZ. A few library names start with IBB or SBB for runtime or SMP/E libraries. For a list of data set names, see the product library customization summary information in MainView product libraries on page 21. Throughout the MainView documentation, the BB*, UBB*, and TOSZ* data set names are used even if the actual library names start with BMC*.

To use RTE to migrate the MainView environment to a new system image

During the installation process, you can set up and deploy your MainView environment to every system on which you will run MainView products.

1 Perform one of the following actions:

   Note

   If you defined the system during the installation process and deployed the MainView environment to it, no further action is required to migrate the MainView environment.

   - If you defined the system during the installation process, but did not deploy the MainView environment to it, deploy it now. Your migration is complete. For information about product deployment, see the Installation System User Guide.

   - If you did not define the system during the installation process, you can use the Installation System now to define your system and deploy the MainView environment to it. For information about defining the system and deploying the MainView environment, see the Installation System User Guide.

To manually migrate the MainView environment to a new system image

1 Copy the following target libraries from the existing system image to all system images where MainView will run:

   - hlq.BB* or hlq.BMC*
   - hlq.SAS*
   - hlq.BMCPWD
   - hlq.TOSZRTCS
   - hlq.TOSZLINK
hlq is the high-level qualifier for MainView product libraries at your site. The asterisk (*) is a generic qualifier that represents all data sets that start with the same high-level qualifier.

**Tip**
The MainView environment requires various target, site, and user libraries. If your site uses shared DASD, determine whether the original system image and the new system image to which you are migrating MainView are part of the same shared DASD complex. If they are, you might want to share the same MainView libraries rather than making copies of them on the new system image.

2 Copy the following user and site libraries from the existing system image to all system images where MainView will run:

- BBSECURE
- SBBCDEF
- SBBCUST
- SBBPROF
- SBBSDEF
- SBBVDEF
- UBBPARAM
For products that use a BBI-SS PAS, you can:

- Create a set of user libraries (UBBxxxx) for each PAS on the system
- Share one set of user libraries among all PASs on the system

3 Perform the following steps for the Runtime Component System (RTCS):

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

a Share or create the following data set:

```hlq.REGISTRY```

b Copy OSZINI00 from your logical SYS1.PARMLIB concatenation and make the appropriate changes if any data sets have different names.

c Add `hlq.TOSZLINK` to your Authorized Program Facility (APF) list.

d Add `hlq.TOSZLINK` to your link list data sets.

e Add RTCS entries to your Program Properties Table (PPT).

f Add the RTCS entry to your Subsystem Name Table (SSN).

g Verify that the target system has the necessary Linkage Index (NSYSLX) and Common Data Area Spaces (MAXCAD) values required by RTCS.

4 On each new system image, perform the following steps:

a Modify the required MainView started task procedures and place them in a JES PROCLIB.

b Modify the MainView CLIST and place it in a JES CLIST library.

c Authorize the BBLINK library by using APF and, if appropriate, add it to the system link list.
If BBLINK is not in the system link list, verify that all MainView started task procedures have a STEPLIB to BBLINK defined.

d Modify IKJTSOnn in SYS1.PARMLIB to add SYSPROG and PMGLAUTH to AUTHTSF, as necessary.

e Make any necessary ESM security changes for started task or user access.

f Allocate or share the MainView Infrastructure (MVI) common registry (previously known as the Alarm Management file system).

g Update your BPXPRM member to mount the directory that contains the MVI common registry.

---

**Note**

When you start the CAS for the first time, connect to PLEXMGR and use the HFSPATH view to add the new MVI common registry.

If you are sharing this data set, you might have to use HFSPATH to specify the data set name as shared. For more information about using the HFSPATH view, see the *MainView Administration Guide*.

---

5 If you run one or more of the following products (which use a z/OS PAS), complete this step:

- CMF MONITOR
- MainView for z/OS
- MainView for UNIX System Services
- MainView SYSPROG Services

a Allocate the z/OS PAS history files by using UBBSAMP member @@YZZ051 (@@YZZ051 is copied from BBILIB).

b Allocate the z/OS PAS PARMFILE by using UBBSAMP member @@YZZ061 (@@YZZ061 is copied from BBILIB).

6 If you run one or more of the following products (which use a BBI-SS PAS), complete this step:

- MainView AutoOPERATOR
- MainView for CICS
- MainView for DB2
- MainView for DBCTL
MainView for IMS Online

MainView for WebSphere MQ

MainView VistaPoint

a Allocate the BBI-SS PAS history files for each PAS by using UBBSAMP member @@YZZ051 (@@YZZ051 is copied from BBILIB).

Copy the BBI-SS PAS VTAM definitions (as specified in BBINOD00) to the VTAMLST library on the new system image and modify the node names, if necessary.

b Allocate the BBI-SS PAS PARMFILE for each PAS UBBSAMP member @@YZZ061 (@@YZZ061 is copied from BBILIB).

c Allocate the BBI-SS PAS trace directory for each PAS by using UBBSAMP(JXT001).

d Allocate the BBIDIV data-in-virtual data set for each PAS by using UBBSAMP(DIVDEF).

e Copy SSLOG from BBSAMP to UBBSAMP and use the job to allocate the BBI-SS image and journal logs for each PAS.

MainView Customization dynamically allocates these logs on the system where you originally installed the products.

f Modify the following UBBPARM members for use on the new system image:

- BBIISP00
- BBIJNT00
- BBINOD00
- BBISSP00
- BBVLAS00
- BBVTAS00
- CFG ssidA

g Copy the BBI-SS PAS VTAM definitions (as specified in BBINOD00) to the VTAMLST library on the new system image and modify the node names, if necessary.
7 If you use MainView Alarm Manager version 2.1, perform the following steps on each new system image:

a Allocate the BBIJRNL1 and BBIJRNL2 journal logs by using UBBsamp(SSLOG).

b Allocate the BBIDIV data-in-virtual data set by using UBBsamp(DIVDEF).

c Modify BBPARM member MVAPRM00 for use on the new system image.

8 If you use MainView Alternate Access, perform the following steps on each new system image:

a Modify UBBsamp member BBVINIT for use on the new system image.

b Copy the MainView Alternate Access VTAM definitions to the VTAMLST library on the new system image and modify the node names, if necessary.

Migrating specific products

After you have migrated the MainView environment to a new system image, you can complete the migration of specific MainView products. This topic identifies additional tasks that must be completed for certain MainView products.

For each MainView product, perform the tasks on each new image.

**Note**

- For complete information about migrating any MainView product, refer to that product’s customization guide.

- During the installation and customization process, you select whether to use runtime libraries or SMP/E libraries. If you use runtime libraries, your customized library names will start with BMC. If you use SMP/E libraries, your customized library names will start with BB, UBB, or TOSZ. A few library names start with IBB or SBB for runtime or SMP/E libraries.

Throughout the MainView documentation, the BB*, UBB*, and TOSZ* data set names are used even if the actual library names start with BMC*.

**Related Information**

- “MainView product libraries” on page 21
Migrating MainView AutoOPERATOR

Use the following procedure to migrate to MainView AutoOperator.

1. Allocate the BBIVARS profile variable pool by using UBBSAMPS(SLOG).

2. Modify UBBPARM member AAOPRM00 for use on the new system image and copy any necessary rule sets to UBBPARM.

   For more information, see the MainView AutoOPERATOR Customization Guide.

Migrating MainView for CICS

The following topic describes migration considerations for MainView for CICS.

For users who are new to MainView for CICS, use the information in the standard implementation procedures and (optional) implementation procedures chapters in the MainView for CICS Customization Guide to make the modifications needed for the CICS regions to support MainView for CICS functionality.

For existing MainView for CICS users that already have CICS and BBI-SS PAS connections defined, continue to use your existing UBBPARM for the following MainView for CICS related members if they exist:

- BBMTXPxx
- BBKMONxx
- BBKTWKxx
- CMRBEXxx
- CMRDTLxx
- CMRPRBxx
- CMRSOPxx
- CMRTTHxx
- BMCICSxx
- BBIJNTxx
- BBIISPxx
- BBISSPxx
CMRBLKxx

Additional important migration information is provided in the migration considerations chapter of the MainView for CICS Customization Guide and in the product release notes when appropriate.

**Migrating MainView for DB2**

Use the following procedure to migrate to MainView for DB2.

1. Grant access to the DB2 IFI.
   
   **Example**
   
   'Grant trace, monitor1, monitor2, display to bbissracfid;'

2. In the BBI-SS PAS JCL, modify the DB2 DSNLOAD library for Call Attach initiation.

3. In the BBI-SS PAS STEPLIB concatenation, modify the load library for access to Catalog Manager and the Data Collector task.

4. In BBCLIB, modify the DMRACT and DOMC CLISTS with the appropriate high-level qualifiers for the Catalog Manager and Data Collector task libraries.

**Migrating MainView for IMS Online**

Use the following procedure to migrate to MainView for IMS online.

1. Customize the IMS system parameters by performing the following steps:
   
   a. Perform one of the following actions:
      
      ■ If the UBBPARM data set is shared by multiple IMS systems, create a copy of the IMFSYS00 member and name it *imsidSYS*, where *imsid* is the IMS identification code.
      
      ■ If the UBBPARM data set is *not* shared, use the default member name IMFSYS00.

   b. Change the SUBSYS parameter to the subsystem ID of the BBI-SS PAS that will be monitoring the IMS system.

   2. Customize the Event Collector parameters if you want them to be different for this IMS system:
a If the UBBPARM data set is shared by multiple IMS systems, create a copy of the IMFECP00 member and name it imsidECP, where imsid is the IMS identification code.

**Note**

If the UBBPARM data set is *not* shared, use the default member name IMFECP00.

b Change the appropriate Event Collector options.

3 Update the JCL for each IMS control region:

a Add an IMFPARM DD statement with two data sets concatenated:
   `hilevel.UBBPARM` followed by `hilevel.BBPARM`.

b Add BBLINK to the STEPLIB concatenation.

c If the Product Authorization password table is not included in BBLINK or the LINKLIST, perform one of the following actions:

- Add the password table data set to the STEPLIB concatenation.
- Add a BMCPSWD DD statement.

4 Update the JCL for each IMS DLISAS by adding BBLINK to the STEPLIB concatenation.

For more information, see the *MainView for IMS Online Customization Guide*.

**Migrating MainView for UNIX System Services**

Use the following procedure to migrate to MainView for UNIX System Service.

1 To access UNIX System Services data, define a user ID for the MainView for UNIX System Services PAS and grant the user ID superuser authority.

The user ID must have an OMVS segment with a UID of 0 and a home of `/`.

**Example**

```
UID = 0000000000
HOME = /
PROGRAM = /bin/false
```

2 Authorize the UNIX Systems Services CSSLIB data set and place it in the system link list or in the STEPLIB concatenation of the PAS startup procedure.
Migrating MainView for WebSphere MQ

Use the following procedure to migrate to MainView for WebSphere MQ.

1. Grant security privileges to the BBI-SS PAS for MainView for WebSphere MQ access.
   Optionally, you can alter QMPROF to pass the user ID instead of the PAS ID to MainView for WebSphere MQ for security checking.

2. For each Queue Manager that is to be monitored, perform the following steps:
   a. In the Queue Manager master JCL, add the MainView for WebSphere MQ Extensions step before the Queue Manager initialization step.
   b. Modify BBLINK in the Queue Manager initialization STEPLIB concatenation.
   c. Modify the SCSQAUTH library in the BBI-SS PAS STEPLIB concatenation.

For more information, see the MainView for WebSphere MQ User Guide.
Index

@@YZZ001 member; BBILIB data set
   @@YZZ001 member 58
@@YZZ011 member 60
@@YZZ022 member 69, 173
@@YZZ051 member 98
@@YZZ052 member 98
@@YZZ061 member 101, 103
@@YZZ061 member; BBILIB data set
   @@YZZ061 member 101

A

ACONCICS parameter 225
action table
   CAS 173
   MainView Alarm Manager (version 2.1) 250
   z/OS PAS 185
adding
   BBI-SS PAS 145
   BBLINK library to link list 64
   CAS definitions 158
Alarm Manager. See MainView Alarm Manager (version 2.1)
ALIAS parameter 110
ALRTRCVE parameter 217
AOEXEC program, authorizing 169
APF-authorizing link library 63
ARM, enabling 166, 217
ARMPOLCY parameter 217
ASMLKED member 130, 131
ATTACH parameter 111
AUDIT parameter 218
Authorization Utility, Product 60
authorized command processor table 169
authorizing
   AOEXEC program 169
   BALCMSG program 169
   BBI-SS PAS 122
   BBLINK library 63
   BBM3API program 169
   BBM9TC21 program 169
   BBM9TC22 program 169
   BBM9TC24 program 169
   BBVJSETP program 169
   EMTMPW program 169
   products to run under ISPF 68
   SMLOAD program 169
   TSLOAD program 169
   TSO command processor 170
AUTHTSF list 68
AUTOID parameter 207
Automatic Restart Manager, enabling 166, 217

B

BAIAPF member 63
BAIMAINV member 65
BALCMSG program, authorizing 169
BAVSHLQ parameter 187
BB library names 24–26, 28, 30, 32, 33, 37, 45
BBACTDEF data set
   CAS 173
   MainView Alarm Manager (version 2.1) 250
   security definitions 62
   z/OS PAS 185
BBAIVP member 57
BBATSP00 member 171
BBCHILV parameter
   MainView Alarm Manager (version 2.1) 251
   z/OS PAS 187
BBCLIB data set 22
BBGROUP name 160
BBHPAS member 83
BBI architecture 16
BBIBBCFG member
parameters 199
using 112

BBICMDS member 136, 231

BBIDIV data set
BBILOG started task
IMAGSTC parameter 225

BBISPxx member
historical recording interval 117
parameters 204
timer-driven services 114

BBIJNT00 member
BBI-SS PAS targets 105, 148
dynamic targets 105
MainView Alarm Manager (version 2.1) 77
TYPE parameter 109

BBIJRNLD DD statements 250

BBILIB data set
:BBMPAS member 181
@@YZZ022 member 69, 173
@@YZZ051 member 98
@@YZZ052 member 98
@@YZZ061 member 101, 103
@@YZZ062 member 103
BAIAPF member 63
BBINOD00 member 148

BBIPARM DD statement
CAS 173
MainView Alarm Manager (version 2.1) 250
MainView Logger 239

BBIPFK member 232
BBIRESP parameter 136
BBISSP parameter 202
BBISSPxx 77

BBISSPxx member
ARM support 166
BBI-SS PAS log handling 128
BBI-SS PAS product initialization 165
CAS ID 167
PRODUCT parameter 77, 165
specifying 202

BBISSPxx parameters 224
MainView (for CICS) 225
MainView (for DBCTL) 225
MainView (for IMS Online) 225
MainView AutoOPERATOR 216, 225
product environment 208

BBITSP00 member description 232

BBIUSR01 user exit 130
BBIUSR02 user exit 131

BBIVARS data set
allocating 118

BBIVARxx member 225
BBIXSP00 member 148

BBLINK load library
adding to link list 64
APF authorization 63
CAS 173
MainView Alarm Manager (version 2.1) 250
MainView Logger 239
TSF authorization 68
z/OS PAS 181, 187

BBM3API program, authorizing 169
BBM9TC21 program, authorizing 169
BBM9TC22 program, authorizing 169
BBM9TC24 program, authorizing 169
BBMCAS procedure 69, 175
BBMLIB data set 22
BBMPAS member 102, 181
BBMTYB00 member 85
BBMXPJCL member 72
BBPAML data set
  BBATSP00 member 171
  BBIBBCFG member 112, 199
  BBISPx member 114, 204
  BBIXNT00 member 77, 105, 148
  BBINOD00 member 148
  BBIVARxx member 225
  BBIXSP00 member 148
  BBTTCP00 member 74, 228
  BKDKDBTW member 115
  BLKDMRW member 115
  CMRBLKn members 115
  considerations 21
  LSLOGDRG member 92
BBPLIL data set 23
BBPROF data set
  BBICMDS member 231
  BBIPFK member 232
  BBITSP00 member 232
  considerations 21
  CYCnn members 231
  LOGPFK member 232
BBSAMP data set
  ASMLKED member 130
  BBAIVP member 57
  BBHPAS member 83
  BBMXPJCL member 72
  considerations 21
  DLOGJG member 129
  HFSALLOC member 75
  ILOGJG member 130
  LOGRNDIV member 87
  LSALCDIV member 92
  LSALCREG member 91
  LSLOGSTC member 94
  MVLOGG member 237
  SSJCL member 122, 168
  SSLOG member 80, 118
  TOMALLOC member 120
  TOMJGCL member 127
BBSDEF data set
  screen definition library 34
BBSECURE data set
  CAS 173
  MainView Alarm Manager (version 2.1) 250
  z/OS PAS 185
BBSOUT parameter 187
BBTLIB data set 23
BBTTCP00 member 74, 228
BBVDEF data set
  CAS 173
  MainView Alarm Manager (version 2.1) 250
  z/OS PAS 185
BBVJSETP program, authorizing 169
BLK parameter 207
BLKDBTW member 115
BLKDMRW member 115
BLKIMFW member 115
BMC library names 24–26, 30, 32, 33, 37, 45

C

CA ACF2
  command processor table 169
CAPS parameter 232
CAS. See coordinating address space (CAS)
CASDEF view 158
CASID parameter 209
CASPERM registry data set
  Allocating 71
CASWAIT parameter 188
CDRM definitions 142
CFGssidA member
  defining 112
  parameters 199
CICSMR menu 135
CICSMVA menu 135
CLIST
  library 22
  MainView 22, 65
CMDCHAR parameter 210
CMDCON parameter 218
CMDID parameter 188
CMDPREF parameter 210
CMDFWAIT parameter 219
CMF Extractor 189
CMF MONITOR
  started task 182
  XCF communication 141
CMF SMF record recording 193
CMFCDSxx DD statement 183
CMFDDUMP DD statement 184
CMFIDSxx DD statement 183
CMFLOG DD statement 184
CMFMMSG DD statement 184

Index 271
CMFXDSLG DD statement 184
CMRBEX parameter 203
CMRBLKn members 115
CMRPRB parameter 204
CMRSOP parameter 204
CMRTTH parameter 204
CNSLEXEC parameter 219
cold start
  BBI-SS PAS 123
  CAS 176
COLORS parameter 232
command authorization table 169
command processor table 169
COMMNDxx member 102
common registry 75
COMMORDR parameter 177
communication links
  BBI-SS PAS 136
  CAS 140
  CASs 142
COMPRESSION parameter 243
configuration
  BBI-SS PAS 112
CONSOLES parameter 219
conventions, documentation 14
CONVXCF parameter 178
coordinating address space (CAS)
  adding ID to BBI-SS PAS 167
  adding system definition 158
  communication 136
  cross-system considerations 140
  defining SSID 175
description 16
dispatching priority 69
migrating definitions 85, 254
START command 69
START command parameters 174
started task procedure 69, 173
TCP/IP communication 142, 159
CPM monitor
  DD statement 182
  mode samplers 189
cross-system communication
  BBI-SS PAS 145
  CAS 140, 153
  messages 161
  overview 136
  troubleshooting 161
Cross-System Coupling Facility
  CAS communication 140
  COMMORDR parameter 177
  CONVXCF parameter 178
  DFLTGRP parameter 178
  SPCF parameter 177
customization
  IMS parameters 27
  manual 45
  summary 38
tasks 45
  UBBPARM parameter library 27
CXEN parameter 189
cycle refresh 230
CYCnn members 231

D
data sets
  high-level qualifier 56
  historical 95
  PARMFILE 101, 103
  product support 21
  security 62
  site 34
  user 35
data-in-virtual (DIV) data set
  MainView Alarm Manager (version 2.1) 81, 250
  MainView AutoOPERATOR 121, 122
  MainView Logger 87, 92
  DB2@MVMR menu 135
  DB2@RXnn menu 135
DD statements
  BBIJRNL 250
  BBIPARM 173, 239, 250
  CMFCDSxx 183
  CMFDUMP 184
  CMFIDSxx 183
  CMFLOG 184
  CMFMSG 184
  CMFXDSLG 184
  CPM monitor 182
  HISTDSnn 185
  PARMFILE 185
  PARMLIB 184
  DEBUG parameter 230
  DEFAULTLOGGER parameter 243
  DESC parameter 109
  DFLTGRP parameter 178
Discovery Agent, host server 72
dispatching priority
  BBI-SS PAS 122
  CAS 69
  MainView Alarm Manager (version 2.1) 83
  z/OS PAS 102
DIV data set. See data-in-virtual (DIV) data set
DIVBLOCKS parameter 243
DIVDSN parameter data-in-virtual (DIV) data set
  MainView Logger 242
DLOGJCL member 129
documentation information 13
DUMP parameter (CAS) 178
DUMPCLAS parameter 211, 232
DUMPCTOUNTMAX parameter 244
DUMPS parameter (BBI-SS PAS) 211, 232
dynamic targets 105, 148

E
electronic documentation 13
EMM parameter 179
EMTMPW program, authorizing 169
Exception Monitor 191
Explorer. See MainView Explorer
extended diagnostic mode (XDM)
  CAS 180
  PAS 194
extended message mode (EMM) 179

G
GDDM/PGF support 65
graphical user interface (GUI) 18
graphing support 65
GTS parameter 224

H
Help
  online 13
Help library
  online Help library 22
HFSALLOC member 75
high-level qualifier for data sets 56
hilevel. See high-level qualifier
HILITE parameter 220
HISTDSnn DD statements 185
historical data set
  BBI-SS PAS 95, 117
  z/OS PAS 95, 187
hlq. See high-level qualifier
HOLD statement 60
host server 72

I
IBB library names 24–26, 28, 30, 32, 33, 37, 45
IEASYMxx member 233
IEASYSxx member 141
ILOGJCL member 130
image identifier 56
image log
  BBI-SS PAS 37, 118
  printing 128
IMFECPO0 member, customizing 27
IMFSYS00 member, customizing 27
IMS parameters 27
IMS parameters, customizing 27
IMS@MVA menu 135
IMS@MVMR menu 135
IMSID parameter 110
IMSPRESP parameter 136
IMSTYPE parameter 109
initializing
  CMF Extractor 189
  z/OS PAS data collectors 190
installation verification program 57
INTERVAL parameter 205, 232
IPM monitor 191
IRRI parameter 206
IRSYNCH parameter 206
ISPF parameter 135, 232

J
JES3CHAR parameter 220
JES3CMD parameter 221
JESCHAR parameter 220
JESCNCT parameter 211
JESFLTR parameter 221
JESTYPE parameter 222
jobcards 56
log switching 128
LOGCYLINDERS parameter 241
LOGGER00 member 88
LOGGING parameter 213, 232
LOGPFK member 232
LOGPREFIX parameter 242
LOGRDIV member 87
LOGRREG member 86
LOGSPACE parameter 245
LOGSSID parameter 244
LOGTRACKS parameter 241
LOGWTO parameter 247, 248
LSALCDIV member 92
LSALCREG member 91
LSLOGGER member 92
LSLOGSTC member 94
LU conversations 154

M
MainView 72
Data Collector load library 122
migrating products 253
MainView (for CICS)
BBISSPxx parameter 225
defining targets 105
initialization 165
timer-driven services 114
trace directory 114
MainView (for DB2)
Data Collector load library 126
initialization 165
timer-driven services 115
trace directory 114
MainView (for DBCTL)
BBISSPxx parameter 225
customizing 27
initialization 165
timer-driven services 115
trace directory 114
MainView (for IMS Online)
BBISSPxx parameter 225
customizing 27
initialization 165
timer-driven services 115
trace directory 114
MainView (for UNIX System) Services
started task statements 185
MainView (for WebSphere MQ)
- BBISSPxx parameter 224
- initialization 165

MainView (for z/OS)
- started task PROC parameters 190, 191
- started task statements 181, 185

MainView Alarm Management (version 5.0)
- allocate file system 75
- MainView Infrastructure common registry, allocating 75

MainView Alarm Manager (version 2.1)
- :product initialization 77
- BBIDIV data set 81
- dispatching priority 83
- initialization parameters 77
- journal logs 80
- PAS startup procedure 83
- security requirements; security:MainView Alarm Manager (version 2.1) 84
- START command 83
- started task procedure 83

MainView AutoOPERATOR 224
- AutoRule 56
- BBISSPxx parameters 216
- BBIVARxx parameters 225
- EXECs 32, 122
- initialization 165
- procedure library 32
- rule sets 56

MainView CLIST, creating 65

MainView Customization
- CLIST library 22
- parameter library 27
- profile library
- MainView Customization 31
- sample library 29

MainView customization methods 15

MainView event management utility 56

MainView Explorer 18, 72
- BBTTCP00 parameters 228
- host server startup 72

MainView Infrastructure
- architecture 16
- running mixed versions 254

MainView library names 28, 30, 32, 33, 37, 45

MainView Logger
- BBISSPxx parameters 214
- data compression 243
- DIV data set 87, 92
- initialization parameters 88
- MainView Logger 90
- message filtering 246
- overview 85
- register data set 86, 91
- started task procedure 90, 237

MainView products 173

MainView Total Object Manager (TOM)
- allocate data sets 120
- started task procedure 127

MANAGEMENTCLASS parameter 245
manual customization
description 45
MAXDATA parameter 148, 163
MAXPERSEC parameter 246
MAXPQE parameter 213
MAXREQ parameter 205
MAXUSER parameter 214
MCS consoles 218
MCTSIZE parameter 222
messages
cross-system communication 161
journal log 212
libraries 22
STALL DETECTED 215
statistics 222
MIGCONS parameter 222
migrating
CAS definitions 85, 254
MainView environment 258
MainView products 253
to new system 258
VTAM definitions 140
mixed versions, running
different system images 255
same system image 256
mode table, VTAM 154
MODEENT table 154
MSGFLT parameter 246
multiple console support (MCS) 218
MULTIREC parameter 246
MVADIV parameter 251
MVAJRNLP parameter 251
MVALARM. See MainView Alarm Manager (version 2.1)
MVALSS parameter 251
MVAPARM parameter 252
MVAPRM00 member 77
MVLOG parameter 214
MVLOGGER member 237
MVLOGSS parameter 215
MVSSYSTEM parameter 240

N
NETVIEW parameter 223
Network Channel Program (NCP) 162
NEWREG parameter 238

O
online Help 13
operational defaults 56

P
panel
  library 23
parameter library
    CAS 173
    description 23, 26
    MainView Alarm Manager (version 2.1) 252
    MainView Customization 27
    MainView Logger 239
    manual customization 27
    using system variables 233
    z/OS PAS 194
PARMFILE data set
  BBI-SS PAS 103
  z/OS PAS 101
PARMFILE DD statement 185
PARMLIB DD statement 184
PF keys
  BBIPFK member 232
  xxxPFK members 233
Plex Manager 16
PRIMAPPL parameter 134
Primary Option Menu panels 135
PRIMMENUS parameter 135
printing image and journal logs 130
PROC statement
  CAS 174
  MainView Alarm Manager (version 2.1) 249
  MainView Logger 238
  z/OS PAS 186
procedure library
  description 23
  using 32
PROCESS statement 234
product
  authorization 60
  installation verification 57
product address space (PAS), overview 17
Product Authorization Utility 60
product libraries
  configuration 38
  site-customized 21
  types 21
PRODUCT parameter
  BBI-SS PAS 165, 215
  MainView Alarm Manager (version 2.1) 77
profile library
  user 21, 23
  using 30
profile variable pool, BBIVARS 118
PROFUSED parameter 223
PTFs 60
publications, related 13

Q
qualifier, for data sets 56

R
RACF
  BBI-SS PAS 122
  command processor table 169
  recording CMF SMF records 193
  recording interval, historical 117
  region size, BBI-SS PAS 122
  register data set, MainView Logger 91
  REGISTERDSN parameter 242
related publications 13
RELEASE parameter 109
RESET parameter 238
RESETA parameter 238
REVERSE parameter 232
RTCS
   description 69
Runtime Component 69
Runtime Component System
description 69
Runtime Component System (RTCS)
description 19
RUSIZES parameter 154, 163

S

sample library
description 23
MainView Customization 29
using 28
SBB library names 24–26, 28, 30, 32, 33, 37, 45
SBBCDEF data set
description configuration library 34
SBBHELP data set 60
SBBPROF data set
considerations 21
creating 60
SBBSDEF data set
creating 60
description 34
SBBVDEF data set
creating 60
description 34
screen definition library 23
SCREEN parameter 232
security
   BBI-SS PAS 122
   BBSECURE data set 173, 185, 250
   command processor table 169
   cross-system communication 142
default definitions 62
   MainView Alarm Manager (version 2.1) 250
required data set 62
site data sets, allocating 60
site libraries 21, 34
SMLOAD program, authorizing 169
SPCF parameter 177
SSID
   BBI-SS PAS procedure 109
   CAS procedure 69, 175
   logs 118, 128
   migration 256
   PARMFILE data set 103
   z/OS PAS procedure 102, 192
   SSJCL member 122, 168
   SSLOG member 80, 180
   STALL parameter 215
START command
   CAS 69
   z/OS PAS 102
started task procedure
   BBI-SS PAS 122
   CAS 173
   log handling 128
   MainView Alarm Manager (version 2.1) 83, 249
   MainView Total Object Manager 127
   MainViewLoggerR 237
   z/OS PAS 102, 181
startup procedure
   BBI-SS PAS 122
   CAS 175
   MainView 83
   MainView Alarm Manager (version 2.1) 249
   MainView Explorer host server 72
   MainView Logger 90, 237
   MainView Total Object Manager 127
   z/OS PAS 102, 181
STORAGECLASS parameter 247
SUBSYS parameter 109, 208
subsystem
   communication 146, 148
subsystem ID
   BBI-SS PAS 103, 109
   CAS 167, 175
   IMS control region 110
   MainView Alarm Manager (version 2.1) 251
   MainView Logger 244
symbolic substitution 233
SYS1.PARMLIB
   APF authorization 63
   IEAAPFxx member 63
   IEASYMxx member 233
   IEASYSxx member 141
   PROGxx member 63
   TSF authorization 68
SYS1.VTAMLIB 155, 163
SYS1.VTAMLST
   BBI-SS PAS definitions 149
   CAS definitions 156
   shared 142
SYSID, defining 56
SYOUT class 187
SYSP parameter 238
sysplex communication 140
system link list 64
system parameters
  BBI-SS PAS 199
  CAS 174
  z/OS PAS 186
system variables substitution 233
systems, migrating 258

T

table
  library 23
target libraries 21
TARGET parameter
  BBIISPxx 207
  MainView Alarm Manager (version 2.1) 77
target system
  attachment 111
  BBI-SS PAS 108
dynamic 105
  jobname 105
  parameters 105, 108
terminal session 133
TCP/IP interface
  BBIISPxx member 224
  BBTTCP00 member 229
  MainView Explorer 229
TCPNAME parameter 229
terminal session
  parameters 133
  TSO command processor 170
terminal session, description 18
timer-driven services 114
TOM. See MainView Total Object Manager (TOM)
TOMALLOC member 127
TOSZ library names 24–26, 28, 30, 32, 33, 37, 45
TRANSfer command 68
TRDIR parameter 208
TSCLIST member 65
TSF authorization 68
TSLOAD program, authorizing 169
tso command processor 170
TYPE parameter
  BBIJNT00 109
  BBITSP00 134
terminal session 134
TZOVRLAP parameter 193

U

UBB library names 24–26, 28, 30, 32, 33, 37, 45
UBBFARM data set
  considerations 21
  creating 60
  z/OS PAS 194
UBBSAMP data set
  considerations 21
  creating 60
user address space (UAS) 18
user exits
  BBIUSR01 130
user libraries 35
user profile library 21, 23
USRID parameter 207

V

verifying product installation 57
versions
  running mixed 254
view library, distributed
  CAS 173
  description 23
  MainView Alarm Manager (version 2.1) 250
  z/OS PAS 185
view table
  CAS 173
  MainView Alarm Manager (version 2.1) 250
  z/OS PAS 185
VTAM
  activating major nodes 157
  BBI-SS PAS definition 149
  CAS definition 153
  MAXDATA parameter 148
  resource manager definitions 149, 156
  RUSIZES parameter 163
  subsystem communication 146
  updating configuration list 157

W

warm start, BBI-SS PAS 123
web browser 18
WTOLOG parameter 247, 248
X

X-MCS consoles 219
XCF
  communication link 140
  customizing group name 160
XCF enabling for CAS 178
XCFGROUP parameter 223
XDM parameter
  CAS 180
  PAS 194

Z

z/OS PAS 95
  created by MainView Customization 102
  data collectors 190
  defining SSID 192
  dispatching priority 102
  PARMFILE data set 101
See also z/OS PAS
ZFSALLOC member 75

XDS data 195