MainView for CICS
Customization Guide

Supporting

Version 6.7 of MainView for CICS

December 2015
Contacting BMC Software

Several methods are available for contacting BMC Software.

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

United States and Canada

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

Outside United States and Canada

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

© Copyright 1987-2015 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, IBM, IBM z, IMS, Language Environment, MVS, RACF, VTAM, WebSphere, and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

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

SAP is a trademark or registered trademark of SAP AG in Germany and in several other countries.

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](http://www.bmc.com/support). From this website, you can:

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

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

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

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

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

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

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

### Chapter 1  Migration considerations
- Release-specific considerations for version 6.7 .......................... 17
  - Supported versions of CICS ....................................................... 17
  - MainView AutoOPERATOR support requirements ...................... 18
  - Changes to reporting programs for CMRDETL (T6E) Data ............ 18
  - Task kill exit supports new dual thresholds (kill and warning) .... 18
  - New options in the CMRSOPT administrative views .................. 19
  - Changes to views ..................................................................... 19
- Release-specific considerations for version 6.6 ............................. 22
  - Changes to the default CMRSOPT options ................................. 22
  - Enhancement to components of response time (CORT) .............. 23
  - Changes to MainView for CICS workloads to support CICS applications .... 23
  - Support for generic target names ............................................. 24
  - Enhancement to security options for CICS actions .................... 25
  - DUMP display changes ........................................................... 26
  - Security considerations for the DUMP view ............................... 26
  - Changes to the task kill exit .................................................... 27
  - Changes to the CMRDATA data segment .................................. 28
  - Obsolete views ....................................................................... 28
  - Changes to views ..................................................................... 28
- Release-specific considerations for version 6.5 ............................. 32
  - Changes in the CREVIEW family of views ................................. 32
  - Changed default fields for summary views ............................... 32
  - New views to replace the CREGDB2*,CREGMQS* and CREGSYS* views ... 33
  - Changes to the task kill exit .................................................... 33
  - Changes to command messages ............................................. 34
  - Changes to the CMRSTATS program ....................................... 35
  - Changes to the CMRCMPWN program ...................................... 36
  - Changes to the T6E fields ....................................................... 36
- Release-specific considerations for version 6.4 ............................. 37
  - Audit data for parameter updates .......................................... 37
New record types in CMRSTATS .................................................................38
Monitor message changes ........................................................................39
Obsolete views ....................................................................................... 40
DUMP display changes ...........................................................................40
Considerations for any new release ............................................................41
Report program considerations .................................................................41
CMRCUEX considerations .......................................................................41
CICS Transaction Gateway support ..........................................................41
Updated views might issue messages .........................................................42
New views not downward–compatible .......................................................42
Updating or switching the BBI-SS PAS to different releases ..................43

Chapter 2 How product libraries should be used 45

Chapter 3 Auto maintenance facility 47
Using the auto maintenance facility ..........................................................47
  Initiating auto maintenance .................................................................49
Auto maintenance resources and messages .............................................49
  CSD definitions .................................................................................49
  Managed resources ...........................................................................50
  Messages ...........................................................................................57
  Auto maintenance BBSAMP members .................................................57

Chapter 4 BBI-SS PAS and CICS target connection 59
Connection facility ................................................................................59
  ACONCICS=NO (default) ...................................................................60
  ACONCICS=YES ............................................................................60
  ACONCICS=ONLY ..........................................................................61
Using static BBIJNT00 entries .................................................................61
Implementing dynamic target definitions .................................................62
  Removing static target definitions ......................................................62
  Enabling dynamic target support .......................................................63
  Creating target definitions .................................................................63
  Special considerations ......................................................................63
Providing a BBI-SS PAS directive to a target CICS region .....................64
AutoConnect .........................................................................................65
  AutoConnect BBI-SS PAS definitions .................................................66
Implementing AutoConnect ..................................................................68
  Initializing AutoConnect on the CICS region .....................................68
  Initializing AutoConnect on the BBI–SS PAS ..................................69
  Terminated AutoConnect targets ......................................................69
AutoConnect affinity factors .................................................................70
Affinity factor examples ........................................................................72

Chapter 5 Using CMRDETL data sets ................................................. 77
Using a history data space ...................................................................79
Restrictions .........................................................................................80
Creating task performance detail history file definitions .....................80
Using the CMRDETL view ..................................................................80
History file specification dialog fields ...............................................84
History file allocation dialog fields ....................................................87
System variables ..............................................................................88
Automatic data set creation ...............................................................89
Manual data set creation ..................................................................89
Detail data-collection scenarios .........................................................91
Controlling data set switching with CONTINUE=YES ......................91
Controlling data set switching with CONTINUE=NO .........................93
CMRDETL data set switch exit .........................................................94
Dual data set considerations .............................................................95
Single data set considerations ..........................................................95
Archiving a CMRDETL data set .....................................................96
Archiving a single CMRDETL data set ..............................................96
Archiving dual CMRDETL data sets ...............................................99
Using CMRDJCL ..............................................................................100
Using another archive utility ..........................................................101
CMRLDTL reload utility ....................................................................101

Chapter 6 Standard implementation procedures ................................ 105
General considerations ....................................................................106
Temporary storage queue conflicts ..................................................106
Enqueue name conflicts ...................................................................106
Processing CICS statistical records ..................................................107
Site-specific security .........................................................................107
OLTCNTL parameter ACTNSEC .....................................................109
Completing implementation ............................................................110
Step 1: Modify CICS startup JCL ......................................................112
Step 2: Update the CSD .................................................................112
Step 3: Create EXCI connection and session definitions ..................114
Step 4: Modify the PLT .................................................................114
Step 5: Modify the SIT .................................................................117
Step 6: Modify the MCT ...............................................................118
Step 7: Implement security ............................................................122
Chapter 7 Optional implementation procedures 125

Setting monitoring conditions (CMRSOPT) ................................. 125
CMRSOPT parameters ................................................................ 127
Modifying the CMRSOPT with the macro ............................... 145
TYPE=INITIAL statement (default conditions) ....................... 146
TYPE=resource statement (optional) ........................................ 146
TYPE=FINAL statement (required) ............................................ 149
Reseting the CMRSOPT ............................................................. 149

Setting problem thresholds (CMRPRBT) ............................... 150
Modifying the CMRPRBT macro ................................................ 151
Reseting the CMRPRBT macro ...................................................... 155

Implementing the task kill function ........................................ 155
Task kill modes ........................................................................ 155
Thresholds and options for task kill function ....................... 156
Task kill limitations and considerations ................................ 157
Transactions that are excluded from task kill processing ........ 159
Starting the task kill function .................................................... 160
Stopping the task kill function ................................................... 160
Setting the task kill parameters ............................................... 160
Reloading changed task kill parameters .................................. 161

Implementing program timing ....................................................... 161
Activating program timing ......................................................... 162
Notes on Program Timing operations ................................... 164

Monitoring sessions and connections ........................................ 165
Session monitoring ................................................................... 165
Connection monitoring ............................................................. 166
Starting and stopping session/connection monitoring ............ 168

Monitoring transient data (TD) queues ....................................... 169
Starting and stopping transient data (TD) queue monitoring .... 169

Monitoring temporary storage (TS) queues ............................... 170
Starting and stopping temporary storage (TS) queue monitoring ... 171

Monitoring the CICS Transaction Gateway ................................ 171
MainView for CICS modifications .............................................. 172
CICS Transaction Gateway modifications ......................... 176
Starting and resetting CTG monitoring ................................ 180

Defining history parameters in BBIPROF .................................. 182

Monitoring SAP transactions ......................................................... 184
Modifying CMRSOPT ................................................................. 184
CMRCUEX rules and conventions .................................................................257
  Restrictions ..........................................................................................258
Sample uses for CMRCUEX .......................................................................258
  Initiating a database call .................................................................259
  Adding data to the user data field ....................................................260
  Replacing program names ...............................................................260
  Replacing netnames ........................................................................261
  Replacing transaction names ........................................................261

Appendix H  FST2 transaction .................................................................263

Appendix I  CREGAGT view commands ...............................................267
Commands available with the CREGAGT view ......................................268

Appendix J  Implementing OLTHCICS to control PLT processing ..........271
Implementing the OLTHCICS program .............................................272
About this book

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

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

Note

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

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

Related publications

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

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

- View 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.
Tip
You can access the BMC Support Central site at http://www.bmc.com/support.

Conventions

This document uses the following special conventions:

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

- Variable text in path names, system messages, or syntax is displayed in italic text: `testsy-instance/fileNam`e

- 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.

Syntax statements

This topic explains conventions for showing syntax statements.

A sample statement follows:

```
COMMAND KEYWORD1 [KEYWORD2 | KEYWORD3] KEYWORD4={YES | NO} fileName...
```

The following table explains conventions for syntax statements and provides examples:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Example</th>
</tr>
</thead>
</table>
| Items in italic type represent variables that you must replace with a name or value. If a variable is represented by two or more words, initial capitals distinguish the second and subsequent words. | `alias`  
`databaseDirectory`  
`serverHostName` |
| Brackets indicate optional items. Do not type the brackets when you enter the option. A comma means that you can choose one or more of the listed options. You must use a comma to separate the options if you choose more than one option. | `[tableName, columnName, field]`  
`[-full, -incremental, -level]` |
<table>
<thead>
<tr>
<th>Convention</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Braces indicate that at least one of the enclosed items is required. Do</td>
<td>`{DBDName</td>
</tr>
<tr>
<td>not type the braces when you enter the item.</td>
<td>`UNLOAD device=(disk</td>
</tr>
<tr>
<td></td>
<td>`{-a</td>
</tr>
<tr>
<td>A vertical bar means that you can choose only one of the listed items. In</td>
<td>`{commit</td>
</tr>
<tr>
<td>the example, you would choose either commit or cancel.</td>
<td></td>
</tr>
<tr>
<td>An ellipsis indicates that you can repeat the previous item or items as</td>
<td><code>columnName...</code></td>
</tr>
<tr>
<td>many times as necessary.</td>
<td></td>
</tr>
</tbody>
</table>
Migration considerations

This topic describes:

- Changes in specific releases of MainView for CICS that might require you to take some action before migrating to the new release
- Issues that you should consider when migrating to any new release of the product

Be sure to review these migration topics and the latest MainView for CICS release notes before upgrading to the new release.

Release-specific considerations for version 6.7

This topic describes migration considerations that are specific to version 6.7 of the MainView for CICS product.

Supported versions of CICS

Version 6.7 of the MainView for CICS product supports the following releases of the IBM CICS system:

- CICS Transaction Server Version 4.1
- CICS Transaction Server Version 4.2
- CICS Transaction Server Version 5.1
- CICS Transaction Server Version 5.2
- CICS Transaction Server Version 5.3

If you continue to use CICS releases that are no longer supported, you must exclude those CICS regions from the new MainView for CICS. You can continue to use earlier supported releases of MainView for CICS to manage unsupported regions.
MainView AutoOPERATOR support requirements

If you use the MainView AutoOPERATOR for CICS option and require that it supports Version 5.3 of CICS Transaction Server (TS), you must be running MainView AutoOPERATOR version 7.4.01 with PTF BQO1711.

Changes to reporting programs for CMRDETL (T6E) Data

Version 6.7 of MainView for CICS will produce T6E records only in the F7 format. This is because version 6.7 of the MainView for CICS product does not support CICS TS (CTS) version 3.1, which was the last CICS release that needed the earlier F6 record format.

The F7 format includes expanded time fields. If you are running version 4.1 (or later) of CTS (and you have not yet done so), you should update your JCL and user-written batch reporting programs (Cobol, PRL, SAS and assembler) to use the following specifications:

- **CMRPRLN** (instead of **CMRPRL**) - reporting program
- **CMRFREDN** (instead of **CMRFRED**) - reformats CMRDETL data into COBOL-compatible format
- **CMRNIFUE** (instead of **CMRSIFUE**) - used in SAS programs
- **CMRSUMDN** (instead of **CMRSUMD**) - summarization program
- **CMRCMPRN** (instead of **CMRCMPR**) - callable decompression program
- **CMRCMPWN** (instead of **CMRCMPW**) - batch decompression program
- **CMRDETLN** (instead of **CMRDETL**) - CMRPRL dictionary
- **CMRCOBCN** (instead of **CMRCOBCP**) - COBOL copybook
- **CMRQT6e** (instead of **FACTRECM**) - assembly DSECT

**WARNING**

Continuing to use the earlier programs (such as CMRPRL) with F7 format records might result in increased CPU overhead usage because the record format is translated from F7 to F6 format (which has less granularity in time fields). In additions, MainView for CICS will withdraw support for the earlier programs, dictionary, copybook, and DSECT in a future release. For more information about the programs, dictionary, copybook and DSECT, see the MainView for CICS Performance Reporter User Guide.

Task kill exit supports new dual thresholds (kill and warning)

The task kill exit in version 6.7 of MainView for CICS supports processing of dual (kill and warning) thresholds for each transaction in KILL mode. The family of
CMRTTHR administration views are also enhanced to set these thresholds. The earlier versions of the CMRTTHnn member are compatible with the new release but as soon as you modify an earlier member with version 6.7 CMRTTHR views, the member can only be administered with version 6.7 MainView for CICS BBI-SS PASs.

You can use the version 6.7 CMRTTHnn members with earlier (supported) releases of MainView for CICS but the earlier versions ignore the new version 6.7 parameters. For more information about dual thresholds, see the section "Implementing the task kill function".

New options in the CMRSOPT administrative views

You can set two additional options with the CMRSOPT administration views:

- Include System Transactions option which determines if the CICS system transactions CSNC, CSNE and CSSY are monitored.
- Shutdown Delay option which determines if MainView for CICS delays its agents shutdown during CICS termination.

These new options are not available with the CMRSOPT macro. The earlier versions of the CMRSOP nn member are compatible with the new release but as soon as you modify an earlier member with version 6.7 CMRSOPT views, the member can only be administered with version 6.7 MainView for CICS BBI-SS PASs.

You can use the version 6.7 CMRSOPnn members with earlier (supported) releases of MainView for CICS but the earlier versions ignore the new version 6.7 parameters.

Changes to views

This version includes changes and enhancements to views. These changes include adding new data to support the new release of CICS, enhancements to the data collection, removing obsolete data, adding or changing hyperlinks to other views, and reformatting to make the views more useful.

The following table lists all of the views that have been added or modified in this release. You should review the table to determine:

- If you have customized any of these views in an earlier release and if so, re-customize the views in the new release.
- If you have created customized views with names that might conflict with any of the new views and if so, rename your customized views.
- If you have defined alarms for any of these views and if so, assess if you need to modify the alarms for the new release.
## Modified and new views in version 6.7 of MainView for CICS

<table>
<thead>
<tr>
<th>$DZSCIPR</th>
<th>$DZSCITR</th>
<th>C#DSAV</th>
<th>C#DSIZ</th>
<th>C#DSTO</th>
</tr>
</thead>
<tbody>
<tr>
<td>C#ESA</td>
<td>C#ESIZ</td>
<td>C#ESTO</td>
<td>C#PROC</td>
<td>C@CMXT</td>
</tr>
<tr>
<td>C@DB2N</td>
<td>C@ELAP</td>
<td>C@ELTM</td>
<td>C@GLBD</td>
<td>C@GLBE</td>
</tr>
<tr>
<td>C@GLBM</td>
<td>C@GLBP</td>
<td>C@GLBT</td>
<td>C@GLBU</td>
<td>C@GLBX</td>
</tr>
<tr>
<td>C@INPQ</td>
<td>C@MONI</td>
<td>C@PICT</td>
<td>C@PRB1</td>
<td>C@PRB2</td>
</tr>
<tr>
<td>C@PRB3</td>
<td>C@PRB4</td>
<td>C@RESP</td>
<td>C@RSTM</td>
<td>C@SVCT</td>
</tr>
<tr>
<td>C@TDBU</td>
<td>C@TDBW</td>
<td>C@TDQL</td>
<td>C@TDQT</td>
<td>C@TDSU</td>
</tr>
<tr>
<td>C@TDSW</td>
<td>C@TSBU</td>
<td>C@TSBW</td>
<td>C@TSA</td>
<td>C@TSKB</td>
</tr>
<tr>
<td>C@TSKC</td>
<td>C@TSKD</td>
<td>C@TSF</td>
<td>C@TSK</td>
<td>C@TSSU</td>
</tr>
<tr>
<td>C@TSSW</td>
<td>C@WMQN</td>
<td>CABEND</td>
<td>CABENDD</td>
<td>CABENDD2</td>
</tr>
<tr>
<td>CABENDQ</td>
<td>CABENDZ</td>
<td>CAPPL</td>
<td>CAPPLD</td>
<td>CAPPLQ</td>
</tr>
<tr>
<td>CAPPLZ</td>
<td>CAPPLZQ</td>
<td>CBUNDLE</td>
<td>CBUNLED</td>
<td>CBUNDLER</td>
</tr>
<tr>
<td>CBUNLEZ</td>
<td>CCAPSPC</td>
<td>CCAPSPCZ</td>
<td>CCONN</td>
<td>CCONN</td>
</tr>
<tr>
<td>CCSUT</td>
<td>CDB2CONR</td>
<td>CDB2ENT</td>
<td>CDB2ENTD</td>
<td>CDB2ENTR</td>
</tr>
<tr>
<td>CDB2ENTZ</td>
<td>CDB2TRNR</td>
<td>CDELAY</td>
<td>CDLYACT</td>
<td>CDLYDEF</td>
</tr>
<tr>
<td>CDOCTMPR</td>
<td>CDSUT</td>
<td>CECSUT</td>
<td>CENQC</td>
<td>CENQMDLR</td>
</tr>
<tr>
<td>CENQW</td>
<td>CEVTBIN</td>
<td>CEVTBINQ</td>
<td>CEVTBINR</td>
<td>CEVTPRC</td>
</tr>
<tr>
<td>CEVTPRCQ</td>
<td>CEXITSD</td>
<td>CEXITSDQ</td>
<td>CEXITSZ</td>
<td></td>
</tr>
<tr>
<td>CFIELD</td>
<td>CFIELD</td>
<td>CFILER</td>
<td>CFILRES</td>
<td>CFLOW</td>
</tr>
<tr>
<td>CFLOWS</td>
<td>CGLBO</td>
<td>CGLBO</td>
<td>CHINTDEX</td>
<td>CHINTDJV</td>
</tr>
<tr>
<td>CHINTDS1</td>
<td>CHINTDTAP</td>
<td>CHINTDTP</td>
<td>CHINTDTQQ</td>
<td>CHINTQ</td>
</tr>
<tr>
<td>CHISTD</td>
<td>CHISTDAP</td>
<td>CHISTDBT</td>
<td>CHISTDC</td>
<td>CHISTDCC</td>
</tr>
<tr>
<td>CHISTDCD</td>
<td>CHISTDCI</td>
<td>CHISTDCN</td>
<td>CHISTDCO</td>
<td>CHISTDD</td>
</tr>
<tr>
<td>CHISTDD1</td>
<td>CHISTDD2</td>
<td>CHISTDD3</td>
<td>CHISTDD4</td>
<td>CHISTDD5</td>
</tr>
<tr>
<td>CHISTDE</td>
<td>CHISTDEX</td>
<td>CHISTDF</td>
<td>CHISTDFP</td>
<td>CHISTDF1</td>
</tr>
<tr>
<td>CHISTDF2</td>
<td>CHISTDF3</td>
<td>CHISTDF4</td>
<td>CHISTDF5</td>
<td>CHISTDF1</td>
</tr>
<tr>
<td>CHISTDF5</td>
<td>CHISTDF1</td>
<td>CHISTDF2</td>
<td>CHISTDF3</td>
<td>CHISTDF4</td>
</tr>
<tr>
<td>CHISTDF4</td>
<td>CHISTDF5</td>
<td>CHISTDF6</td>
<td>CHISTDF7</td>
<td>CHISTDF8</td>
</tr>
<tr>
<td>CHISTDF3</td>
<td>CHISTDF4</td>
<td>CHISTDF5</td>
<td>CHISTDF6</td>
<td>CHISTDF7</td>
</tr>
<tr>
<td>CHISTDF2</td>
<td>CHISTDF3</td>
<td>CHISTDF4</td>
<td>CHISTDF5</td>
<td>CHISTDF6</td>
</tr>
<tr>
<td>CHISTDF1</td>
<td>CHISTDF2</td>
<td>CHISTDF3</td>
<td>CHISTDF4</td>
<td>CHISTDF5</td>
</tr>
<tr>
<td>CHISTDF0</td>
<td>CHISTDF1</td>
<td>CHISTDF2</td>
<td>CHISTDF3</td>
<td>CHISTDF4</td>
</tr>
<tr>
<td>CHISTDF5</td>
<td>CHISTDF6</td>
<td>CHISTDF7</td>
<td>CHISTDF8</td>
<td>CHISTDF9</td>
</tr>
<tr>
<td>CHISTDF4</td>
<td>CHISTDF5</td>
<td>CHISTDF6</td>
<td>CHISTDF7</td>
<td>CHISTDF8</td>
</tr>
<tr>
<td>CHISTDF3</td>
<td>CHISTDF4</td>
<td>CHISTDF5</td>
<td>CHISTDF6</td>
<td>CHISTDF7</td>
</tr>
<tr>
<td>CHISTDF2</td>
<td>CHISTDF3</td>
<td>CHISTDF4</td>
<td>CHISTDF5</td>
<td>CHISTDF6</td>
</tr>
<tr>
<td>CHISTDF1</td>
<td>CHISTDF2</td>
<td>CHISTDF3</td>
<td>CHISTDF4</td>
<td>CHISTDF5</td>
</tr>
<tr>
<td>CHISTDF0</td>
<td>CHISTDF1</td>
<td>CHISTDF2</td>
<td>CHISTDF3</td>
<td>CHISTDF4</td>
</tr>
<tr>
<td>CHISTDN5</td>
<td>CHISTDO1</td>
<td>CHISTDO2</td>
<td>CHISTDO3</td>
<td>CHISTDO4</td>
</tr>
<tr>
<td>CHISTDN4</td>
<td>CHISTDO3</td>
<td>CHISTDO4</td>
<td>CHISTDO5</td>
<td>CHISTDO6</td>
</tr>
<tr>
<td>CHISTDN3</td>
<td>CHISTDO5</td>
<td>CHISTDO6</td>
<td>CHISTDO7</td>
<td>CHISTDO8</td>
</tr>
<tr>
<td>CHISTDN2</td>
<td>CHISTDO7</td>
<td>CHISTDO8</td>
<td>CHISTDO9</td>
<td>CHISTD0</td>
</tr>
<tr>
<td>CHISTDN1</td>
<td>CHISTDO9</td>
<td>CHISTD0</td>
<td>CHISTD1</td>
<td>CHISTD2</td>
</tr>
<tr>
<td>CHISTDN</td>
<td>CHISTD1</td>
<td>CHISTD2</td>
<td>CHISTD3</td>
<td>CHISTD4</td>
</tr>
<tr>
<td>CHISTDN5</td>
<td>CHISTD2</td>
<td>CHISTD3</td>
<td>CHISTD4</td>
<td>CHISTD5</td>
</tr>
<tr>
<td>CHISTDN4</td>
<td>CHISTD3</td>
<td>CHISTD4</td>
<td>CHISTD5</td>
<td>CHISTD6</td>
</tr>
<tr>
<td>CHISTDN3</td>
<td>CHISTD4</td>
<td>CHISTD5</td>
<td>CHISTD6</td>
<td>CHISTD7</td>
</tr>
<tr>
<td>CHISTDN2</td>
<td>CHISTD5</td>
<td>CHISTD6</td>
<td>CHISTD7</td>
<td>CHISTD8</td>
</tr>
<tr>
<td>CHISTDN1</td>
<td>CHISTD6</td>
<td>CHISTD7</td>
<td>CHISTD8</td>
<td>CHISTD9</td>
</tr>
<tr>
<td>CHISTDN</td>
<td>CHISTD7</td>
<td>CHISTD8</td>
<td>CHISTD9</td>
<td>CHISTD10</td>
</tr>
<tr>
<td>CHISTDN5</td>
<td>CHISTD8</td>
<td>CHISTD9</td>
<td>CHISTD10</td>
<td>CHISTD11</td>
</tr>
<tr>
<td>CHISTDN4</td>
<td>CHISTD9</td>
<td>CHISTD10</td>
<td>CHISTD11</td>
<td>CHISTD12</td>
</tr>
<tr>
<td>CHISTDN3</td>
<td>CHISTD10</td>
<td>CHISTD11</td>
<td>CHISTD12</td>
<td>CHISTD13</td>
</tr>
<tr>
<td>CHISTDN2</td>
<td>CHISTD11</td>
<td>CHISTD12</td>
<td>CHISTD13</td>
<td>CHISTD14</td>
</tr>
<tr>
<td>CHISTDN1</td>
<td>CHISTD12</td>
<td>CHISTD13</td>
<td>CHISTD14</td>
<td>CHISTD15</td>
</tr>
<tr>
<td>CHISTDN</td>
<td>CHISTD13</td>
<td>CHISTD14</td>
<td>CHISTD15</td>
<td>CHISTD16</td>
</tr>
<tr>
<td>CHISTDN5</td>
<td>CHISTD14</td>
<td>CHISTD15</td>
<td>CHISTD16</td>
<td>CHISTD17</td>
</tr>
<tr>
<td>CHISTDN4</td>
<td>CHISTD15</td>
<td>CHISTD16</td>
<td>CHISTD17</td>
<td>CHISTD18</td>
</tr>
<tr>
<td>CHISTDN3</td>
<td>CHISTD16</td>
<td>CHISTD17</td>
<td>CHISTD18</td>
<td>CHISTD19</td>
</tr>
<tr>
<td>CHISTDN2</td>
<td>CHISTD17</td>
<td>CHISTD18</td>
<td>CHISTD19</td>
<td>CHISTD20</td>
</tr>
<tr>
<td>CHISTDN1</td>
<td>CHISTD18</td>
<td>CHISTD19</td>
<td>CHISTD20</td>
<td>CHISTD21</td>
</tr>
<tr>
<td>CHISTDN</td>
<td>CHISTD19</td>
<td>CHISTD20</td>
<td>CHISTD21</td>
<td>CHISTD22</td>
</tr>
<tr>
<td>CHISTDN5</td>
<td>CHISTD20</td>
<td>CHISTD21</td>
<td>CHISTD22</td>
<td>CHISTD23</td>
</tr>
<tr>
<td>CHISTDN4</td>
<td>CHISTD21</td>
<td>CHISTD22</td>
<td>CHISTD23</td>
<td>CHISTD24</td>
</tr>
<tr>
<td>CHISTDN3</td>
<td>CHISTD22</td>
<td>CHISTD23</td>
<td>CHISTD24</td>
<td>CHISTD25</td>
</tr>
<tr>
<td>CHISTDN2</td>
<td>CHISTD23</td>
<td>CHISTD24</td>
<td>CHISTD25</td>
<td>CHISTD26</td>
</tr>
<tr>
<td>CHISTDN1</td>
<td>CHISTD24</td>
<td>CHISTD25</td>
<td>CHISTD26</td>
<td>CHISTD27</td>
</tr>
<tr>
<td>CHISTDN</td>
<td>CHISTD25</td>
<td>CHISTD26</td>
<td>CHISTD27</td>
<td>CHISTD28</td>
</tr>
<tr>
<td>CHISTDN5</td>
<td>CHISTD26</td>
<td>CHISTD27</td>
<td>CHISTD28</td>
<td>CHISTD29</td>
</tr>
<tr>
<td>CHISTDN4</td>
<td>CHISTD27</td>
<td>CHISTD28</td>
<td>CHISTD29</td>
<td>CHISTD30</td>
</tr>
<tr>
<td>CHISTDN3</td>
<td>CHISTD28</td>
<td>CHISTD29</td>
<td>CHISTD30</td>
<td>CHISTD31</td>
</tr>
<tr>
<td>CHISTDN2</td>
<td>CHISTD29</td>
<td>CHISTD30</td>
<td>CHISTD31</td>
<td>CHISTD32</td>
</tr>
<tr>
<td>CHISTDN1</td>
<td>CHISTD30</td>
<td>CHISTD31</td>
<td>CHISTD32</td>
<td>CHISTD33</td>
</tr>
<tr>
<td>CHISTDN</td>
<td>CHISTD31</td>
<td>CHISTD32</td>
<td>CHISTD33</td>
<td>CHISTD34</td>
</tr>
<tr>
<td>CHISTDN5</td>
<td>CHISTD32</td>
<td>CHISTD33</td>
<td>CHISTD34</td>
<td>CHISTD35</td>
</tr>
<tr>
<td>CHISTDN4</td>
<td>CHISTD33</td>
<td>CHISTD34</td>
<td>CHISTD35</td>
<td>CHISTD36</td>
</tr>
<tr>
<td>CHISTDN3</td>
<td>CHISTD34</td>
<td>CHISTD35</td>
<td>CHISTD36</td>
<td>CHISTD37</td>
</tr>
<tr>
<td>CHISTDN2</td>
<td>CHISTD35</td>
<td>CHISTD36</td>
<td>CHISTD37</td>
<td>CHISTD38</td>
</tr>
<tr>
<td>CHISTDN1</td>
<td>CHISTD36</td>
<td>CHISTD37</td>
<td>CHISTD38</td>
<td>CHISTD39</td>
</tr>
<tr>
<td>CHISTDN</td>
<td>CHISTD37</td>
<td>CHISTD38</td>
<td>CHISTD39</td>
<td>CHISTD40</td>
</tr>
<tr>
<td>Modified and new views in version 6.7 of MainView for CICS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHISTDQ4</td>
<td>CHISTDQ5</td>
<td>CHISTDPG</td>
<td>CHISTDQ</td>
<td>CHISTDQ1</td>
</tr>
<tr>
<td>CHISTDQ2</td>
<td>CHISTDQ3</td>
<td>CHISTDQ4</td>
<td>CHISTDQ5</td>
<td>CHISTDS</td>
</tr>
<tr>
<td>CHISTDSA</td>
<td>CHISTDSB</td>
<td>CHISTDSC</td>
<td>CHISTDSD</td>
<td>CHISTDST</td>
</tr>
<tr>
<td>CHISTDS1</td>
<td>CHISTDS2</td>
<td>CHISTDS3</td>
<td>CHISTDS4</td>
<td>CHISTDS6</td>
</tr>
<tr>
<td>CHISTDS7</td>
<td>CHISTDS8</td>
<td>CHISTDS9</td>
<td>CHISTDTC</td>
<td>CHISTDTM</td>
</tr>
<tr>
<td>CHISTDTTP</td>
<td>CHISTDTQ</td>
<td>CHISTDBW</td>
<td>CHISTTPQ</td>
<td>CHISTQ</td>
</tr>
<tr>
<td>CHISTUOW</td>
<td>CHISTZ</td>
<td>CHPROG</td>
<td>CIAID</td>
<td>CIDCT</td>
</tr>
<tr>
<td>CIICE</td>
<td>CIPCONN</td>
<td>CJVMSRV</td>
<td>CJVMSRVR</td>
<td>CLIBRD</td>
</tr>
<tr>
<td>CLIBRR</td>
<td>CLSRL</td>
<td>CLSRS</td>
<td>CLSRW</td>
<td>CMODNAM</td>
</tr>
<tr>
<td>CMRPRBTQ</td>
<td>CMRSOPT</td>
<td>CMRSOPTQ</td>
<td>CMRTTHR</td>
<td>CMRTTHRE</td>
</tr>
<tr>
<td>CMRTTHRT</td>
<td>CMRWTHR</td>
<td>CMRWTHRE</td>
<td>CMRWTHRT</td>
<td>CMXTC</td>
</tr>
<tr>
<td>CORTD</td>
<td>CORTDS1</td>
<td>CORTIQ</td>
<td>CORTDS1</td>
<td>CORTSQ</td>
</tr>
<tr>
<td>COSGBDLQ</td>
<td>COSGBDLQ</td>
<td>COSGBDLR</td>
<td>COSGBDLZ</td>
<td>COSGSRV</td>
</tr>
<tr>
<td>COSGSRVQ</td>
<td>COSGSRVQ</td>
<td>COSGSRVZ</td>
<td>CPGMS</td>
<td>CPIPLNED</td>
</tr>
<tr>
<td>CPIPLNER</td>
<td>CPIROG</td>
<td>CPIROGD</td>
<td>CPIROGR</td>
<td>CRDELAY</td>
</tr>
<tr>
<td>CREGAGT</td>
<td>CREGDBC</td>
<td>CREGDETL</td>
<td>CREGDET4</td>
<td>CREGDET5</td>
</tr>
<tr>
<td>CREGDET6</td>
<td>CREGDSA</td>
<td>CREGDSAS</td>
<td>CREGHR</td>
<td>CREGHRDZ</td>
</tr>
<tr>
<td>CREGHRHZ</td>
<td>CREGHRZ</td>
<td>CREGION</td>
<td>CREGIONR</td>
<td>CREGIONZ</td>
</tr>
<tr>
<td>CREGMSGR</td>
<td>CREGOVER</td>
<td>CREGOVR</td>
<td>CREGPRB</td>
<td>CREGSTG</td>
</tr>
<tr>
<td>CREGSTGZ</td>
<td>CREGSYS</td>
<td>CREGTDQ</td>
<td>CREGTDQD</td>
<td>CREGTSP</td>
</tr>
<tr>
<td>CREGTSPD</td>
<td>CREVDF</td>
<td>CREVDH</td>
<td>CREVD1</td>
<td>CREVDX</td>
</tr>
<tr>
<td>CREVDY</td>
<td>CREVDZ</td>
<td>CREVD4</td>
<td>CREVD8</td>
<td>CREVD9</td>
</tr>
<tr>
<td>CREVIEW</td>
<td>CRGNDASH</td>
<td>CSDCT</td>
<td>CSESS</td>
<td>CTASKD</td>
</tr>
<tr>
<td>CTASKDD</td>
<td>CTASKDEX</td>
<td>CTASKDV</td>
<td>CTASKDS1</td>
<td>CTASKDTP</td>
</tr>
<tr>
<td>CTASKDTQ</td>
<td>CTASKQ</td>
<td>CTCBS</td>
<td>CTCBSD</td>
<td>CTCBSQ</td>
</tr>
<tr>
<td>CTCBSZ</td>
<td>CTCPSRV</td>
<td>CTCPSRVQ</td>
<td>CTCPSRVR</td>
<td>CTDQUED</td>
</tr>
<tr>
<td>CTDQUER</td>
<td>CTERM</td>
<td>CTGCICS</td>
<td>CTGCICSD</td>
<td>CTGHCIC</td>
</tr>
<tr>
<td>CTGHIST</td>
<td>CTGLSRV</td>
<td>CTGSPRV</td>
<td>CTGSPRVD</td>
<td>CTGSPRVL</td>
</tr>
<tr>
<td>CTGSPRVD</td>
<td>CTGSPRVD</td>
<td>CTIOT</td>
<td>CTRAN</td>
<td>CTRAND</td>
</tr>
<tr>
<td>CTRAND1</td>
<td>CTRANQ</td>
<td>CTRANR</td>
<td>CTRANSV</td>
<td>CTRANZ</td>
</tr>
<tr>
<td>CTRNCLR</td>
<td>CTSMODLR</td>
<td>CTSTE</td>
<td>CURIMAP</td>
<td>CURIMAPA</td>
</tr>
<tr>
<td>CURIMAPC</td>
<td>CURIMAPD</td>
<td>CURIMAPJ</td>
<td>CURIMAPP</td>
<td>CURIMAPR</td>
</tr>
</tbody>
</table>
Modified and new views in version 6.7 of MainView for CICS

<table>
<thead>
<tr>
<th>CURIMAPU</th>
<th>CURIMAPZ</th>
<th>CWEBSRVR</th>
<th>CWKLDDET</th>
<th>CWOVER</th>
</tr>
</thead>
<tbody>
<tr>
<td>CWOVERS</td>
<td>DTRAC</td>
<td>EZCFAST</td>
<td>EZCICS</td>
<td>EZCMPROG</td>
</tr>
<tr>
<td>EZCMREGN</td>
<td>EZCMSTOR</td>
<td>EZCMTASK</td>
<td>EZCREGN</td>
<td>EZCSMON</td>
</tr>
<tr>
<td>EZCSOPS</td>
<td>STRAC</td>
<td>SYSOVER</td>
<td>TASKCON</td>
<td>TASKD</td>
</tr>
<tr>
<td>TASKDSA</td>
<td>TASKFIL</td>
<td>TASKLCK</td>
<td>TASKPGM</td>
<td>TASKQ</td>
</tr>
<tr>
<td>TASKSES</td>
<td>TASKTDQ</td>
<td>TASKTRD</td>
<td>TASKTSQ</td>
<td>THINTDEX</td>
</tr>
<tr>
<td>THINTDGV</td>
<td>THINTDS1</td>
<td>THINTDTP</td>
<td>THINTDTQ</td>
<td>THINTQ</td>
</tr>
<tr>
<td>TSQUECF</td>
<td>TSQUECFZ</td>
<td>TSQUEUE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Release-specific considerations for version 6.6

This topic describes migration considerations that are specific to version 6.6 of the MainView for CICS product.

Changes to the default CMRSOPT options

This version has changed some default settings for CMRSOPT parameters, which affect various data collection settings for the CICS region.

Specifically, this version has made the following changes to the default CMRSOPT settings, the CMRSOPT macro, or the distributed CMRUSOPT sample member:

- Earlier versions turned on 13 seldom used SMF parameters out of 23 -- enabling the FIC2 transaction to write SMF statistics data.
  This version leaves the parameters turned off by default.

- Earlier versions specified the default setting of DYNAMIC=104, which turned on DB2 PLAN display data collection.
  This version specifies the default is DYNAMIC=0, turning collection off. Data collection is no longer necessary because the CDB2ENT and CDB2TRN views have replaced the PLAN display.

  With this default, MainView sends the message **FT161I DB2 PLAN Table collection will be suppressed** to the CICS joblog when the extractor is activated.
Earlier versions specified that the default assembled CMRSOPT turned on all resource data collections for full-screen graph displays for all transactions and files and factious names for the other resources.

In this version, the default CMRSOPT does not turn on data collections for full-screen graph displays.

All of these changes affect only the distributed CMRSOPT and users who create new CMRSOPxx elements with the ADD command in CMRSOPT views. Existing customized CMRSOPT module and CMRSOPxx elements will continue to work as they did in earlier releases.

Enhancement to components of response time (CORT)

This version of MainView for CICS includes enhancements and changes to CORT views.

MainView for CICS workloads track the performance of transactions matching the criteria in the workload definitions. This data is significantly enhanced to provide all facets of transaction response time data and is now available in a set of CORT* views that are similar to the TASK, CHIST, and CHINT families of views. You can access the new views through the original workload-related (CORT) views (CFLOW, CFLOWS, CWOVER, and CWOVERS). For more information about the CORT* views, see the MainView for CICS Online Services Reference Manual.

Note

The original CORT views are updated. These views and the new CORT* views are not compatible between this version and earlier versions of MainView for CICS. BMC recommends that you initialize new MVI history files for this version to avoid problems with earlier versions of CORT records in the interval recorder data. In addition, the CWOVERD and CWOVERDS views are no longer available in version 6.6 of the MainView for CICS product. Customized CFLOW, CFLOWS, CWOVER, and CWOVERS views will not work in version 6.6.

Changes to MainView for CICS workloads to support CICS applications

This version of MainView for CICS includes improved support for monitoring the performance of your CICS applications.

An enhancement to the MainView for CICS workload monitor definitions (CWKLDDEF) now allows you to collect and aggregate response time data for transactions executing as part of a CICS application or platform.
Also, you can now access the definition dialogs in either MainView Explorer or the 3270 interface.

MainView for CICS workloads that monitor transactions executing as part of a CICS application will now collect and aggregate both the average response time for transactions and all of the components of response time (CORT).

**Note**

- Existing workload definitions are compatible with version 6.6 of the MainView for CICS product.
- You can share CWKLDDEF (BBKTWK00) with definitions for platforms or applications with earlier versions of MainView for CICS. These definitions are listed in CWKLDDEF of the earlier versions as "workload" with no criteria available for them and the @RSTM is not started for them.
- If you attempt to change a workload definition for an application or platform with an earlier version of MainView for CICS, the results are unpredictable.

**Support for generic target names**

This version allows you to specify generic target names when specifying monitors to be automatically started with the TARGET and BLK statements in BBPARM member BBIISPxx.

You can specify the wildcard character asterisk (*) or plus sign (+) when specifying targets that use AUTOBLK statements to automatically start low-level monitors when their targets initialize.

To implement this new feature, you must specify GENERIC=YES before any TARGET and BLK entries in BBPARM member BBIISPxx:

Also, you must specify a target type keyword on individual entries because TARGET and BLK entries are shared by the following products:

- MainView AutoOPERATOR
- MainView for CICS
- MainView for DB2
- MainView for IMS
You must specify the target type keyword to denote that the entry is for a specific type when generic targets are specified. Target types must match the values specified in the BBIJNTxx member as follows:

- **MVS**: specifies a MainView AutoOPERATOR target
- **CICS**: specifies a CICS region
- **DB2**: specifies a DB2 region
- **IMS**: specifies an IMS or DBCTL region

**Example**

```
TARGET=*,BLK=BLKCICS,TYPE=CICS
TARGET=PRD*,BLK=PRDCICS,TYPE=CICS
TARGET=P+I*,BLK=PRDIMS,TYPE=IMS
TARGET=DB2+,BLK=PRDDB2,TYPE=DB2
TARGET=AO*,BLK=AOBLK,TYPE=MVS
```

**Enhancement to security options for CICS actions**

MainView *for CICS* checks the authority of a user to perform actions that would modify CICS or CICS resources. To do so, MainView *for CICS* uses standard System Authorization Facility (SAF) calls to external security managers (ESMs) when basic or enhanced security is active on the BBI-SS PAS.

After confirming the user's authority, MainView *for CICS* runs the action on the CICS side under the authority of the user ID that started the long-running task JNL2.

Version 6.6 adds the ability to execute these modification requests under the authority of the *original* requesting user ID (instead of the JNL2 user ID) when you specify ACTNSEC=YES in CICS SIT. To implement this feature, add the following statement to the SIT parameters for the CICS region:

```
INITPARM=(OLTCNTL='ACTNSEC=YES')
```

If your site is already using the INITPARM for OLTCNTL to provide a BBI-SS PAS directive for target management, you can concatenate the new parameter as shown in the following example:

```
INITPARM=(OLTCNTL='SUBSYS=WBPD,ACTNSEC=YES')
```
**DUMP display changes**

The full-screen DUMP display and Storage options panels now run in windows mode in version 6.6.

The new DUMP (or PK) view replaces these full-screen options. The new windows mode view supports hyperlinks from various MainView for CICS views to display storage information in MainView Explorer.

In the DUMP view, you can use the hyperlinks to examine CICS storage, and you can display the unformatted content of many CICS control blocks.

For more information about the DUMP view, see “Security considerations for the DUMP view” on page 26 and the MainView for CICS Online Services Reference Manual.

**Security considerations for the DUMP view**

The DUMP view includes support for the Z (ZAP) line command that allows you to modify storage in the BBI-SS PAS or a target address space.

Modifying storage can have unintended results, including abends in the BBI-SS PAS or target address space. BMC strongly recommends that you implement security to control who has access to use this command.

To enable the ZAP command, you must specify `PKZAP=YES` in BBPARM member BBIISP00 and restart the PAS.

The default, `PKZAP=NO`, prevents the ZAP line command from taking any action before you have customized security for accessing the ZAP command.

**Dynamically enabling the ZAP line command**

To dynamically change the setting of the PKZAP parameter without having to restart the PAS, use the PKZAP primary command on the DUMP view. You should also enable security for the PKZAP command. For more information, see the MainView Security Reference Manual.

Issue the following primary command:
PKZAP [parm]

parm can be YES, NO, ON, or OFF.

The following confirmation dialog is displayed:

**Figure 1: Confirm PKZAP option dialog**

```
COMMAND ===> SCROLL ===> CSR

User BOLUSR1 is modifying the PKZAP runtime option to YES (Yes/No/On/Off)

END to exit saving changes
CANCEL to exit without saving changes
HELP to view related help
RESET to reset to initial data
```

The following message is displayed:

BBQPE019W User BOLUSR1 modified the PKZAP option to YES

**Changes to the task kill exit**

The task kill exit now supports the following three new thresholds:

- Temporary Storage access
- Transient storage access
- IBM WebSphere for MQ (MQS) access

Earlier releases included MQS access in database thresholds. In this version, MQS access has its own threshold, and the database threshold values no longer include MQS access values.

**To ensure MQS threshold monitoring is performed for a task in the enhanced task kill exit**

1. Review the CMRTTH.xx member to determine the target groups or transaction groups that you need to modify to support the new MQS Call threshold and updated Database Call threshold.

2. Make the required modifications.
Note
If you do not use the Database Call threshold to monitor MQS calls, no changes are required.
If a transaction group or target group does not have a Database Call threshold for monitoring MQS calls, you do not have to modify that group.

Changes to the CMRDATA data segment

The CMRDATA data segment contains miscellaneous database data that MainView for CICS collects. You can append CMRDATA to the CICS SMF 110 Task Performance record by adding a CMRDATA definition to the CICS Monitoring Control Table (MCT).

The appended CMRDATA segment now includes data from the T6EUDAT2 user data field in the history record. The T6EUDAT2 field contains 64 bytes of user data from the CMRCUEX user exit utility. This feature is not available for users running CICS TS Version 3.1.

This change increases the size of the CMRDATA segment. If you have the CMRDATA definition in your MCT, you must reassemble a CICS MCT and include the new BBSAMP member CMRNDMCT that defines the CMRDATA segment. If you do not reassemble the MCT, the following message is displayed when the first transaction data record is processed, and each time the MainView for CICS extractor restarts:

FT006W - CMRDATA area is too small for additional data

Obsolete views

The following views are no longer accessible:

- CWOVERD
- CWOVERDS

Changes to views

This version includes the following changes and enhancements to views:

- The CREGDSAS view provides additional data that you can use to set alarms for storage utilization.
The CREGION view provides statistics about region Max Tasks for use in setting alarms.

The CCONN* view provides additional information and 14 new SET actions.

The CTASKD and CHISTD views display the name of the last abended program. The CMRDETL record also includes this new field.

The following table lists all of the views that have changed or were modified in this release.

These changes include adding new data to support the new release of CICS, enhancements to the data collection, removing obsolete data, adding or changing hyperlinks to other views, and reformatting to make the views more useful.

You should review and determine if you have customized any of these views in an earlier release and if so, re-customize the views in the new release. You should also determine if you have defined alarms for any of these views and assess if you need to modify the alarms for the new release.

### Changed views in version 6.6 of MainView for CICS

<table>
<thead>
<tr>
<th>Changed Views</th>
<th>CREGION</th>
<th>CCONN*</th>
<th>CTASKD</th>
<th>CHISTD</th>
</tr>
</thead>
<tbody>
<tr>
<td>C#DSAV</td>
<td>C@ELAP</td>
<td>C@ELTM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C@INPQ</td>
<td>C@RESP</td>
<td>C@RSTM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAIDSD</td>
<td>CAIDSQ</td>
<td></td>
<td>CCONND</td>
<td></td>
</tr>
<tr>
<td>CCONND1</td>
<td>CCONNQ</td>
<td></td>
<td></td>
<td>CCONNR</td>
</tr>
<tr>
<td>CDB2CON2</td>
<td>CDB2CONR</td>
<td></td>
<td></td>
<td>CDB2ENTQ</td>
</tr>
<tr>
<td>CDB2ENTR</td>
<td>CDB2TRNQ</td>
<td></td>
<td></td>
<td>CDB2TRNR</td>
</tr>
<tr>
<td>CDELAYAQ</td>
<td>CDELAYGQ</td>
<td></td>
<td></td>
<td>CDELAYQ</td>
</tr>
<tr>
<td>CDELAYTQ</td>
<td>CDLYACTQ</td>
<td></td>
<td></td>
<td>CDLYDEFQ</td>
</tr>
<tr>
<td>CDOCTMPQ</td>
<td>CDOCTMPR</td>
<td></td>
<td></td>
<td>CDUMPCDQ</td>
</tr>
<tr>
<td>CENQMDLQ</td>
<td>CENQMDLR</td>
<td></td>
<td></td>
<td>CFILED</td>
</tr>
<tr>
<td>CFILED1</td>
<td>CFILED2</td>
<td></td>
<td>CFILRES</td>
<td></td>
</tr>
<tr>
<td>CFILER</td>
<td>CFILRESQ</td>
<td></td>
<td>CFILRESQ</td>
<td></td>
</tr>
<tr>
<td>CFLOW</td>
<td>CFLOWS</td>
<td></td>
<td></td>
<td>CHCONNQ</td>
</tr>
<tr>
<td>CHFILEQ</td>
<td>CHINTQ</td>
<td></td>
<td></td>
<td>CHIST</td>
</tr>
<tr>
<td>CHISTD</td>
<td>CHISTDAP</td>
<td></td>
<td></td>
<td>CHISTDCD</td>
</tr>
<tr>
<td>CHISTDCO</td>
<td>CHISTDIO</td>
<td></td>
<td></td>
<td>CHISTDS2</td>
</tr>
<tr>
<td>CHISTQ</td>
<td>CHPROGQ</td>
<td></td>
<td></td>
<td>CHSESSQ</td>
</tr>
<tr>
<td>CHTDQC</td>
<td>CHTDQD</td>
<td></td>
<td></td>
<td>CHTDQQ</td>
</tr>
<tr>
<td>CHTSQC</td>
<td>CHTSQD</td>
<td></td>
<td></td>
<td>CHTSQQ</td>
</tr>
</tbody>
</table>
### Changed views in version 6.6 of MainView for CICS

<table>
<thead>
<tr>
<th>Current View</th>
<th>New View 1</th>
<th>New View 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICESD</td>
<td>CICESQ</td>
<td>CIPCONN2</td>
</tr>
<tr>
<td>CIPCONND</td>
<td>CIPCONNQ</td>
<td>CIPCONNR</td>
</tr>
<tr>
<td>CJVMPLQ</td>
<td>CJVMSRV</td>
<td>CJVMSRVD</td>
</tr>
<tr>
<td>CJVMSRVQ</td>
<td>CJVMSRVR</td>
<td>CJVMSRVRZ</td>
</tr>
<tr>
<td>CLIBR</td>
<td>CLIBRD</td>
<td>CLIBRQ</td>
</tr>
<tr>
<td>CLIBRR</td>
<td>CLIBRZ</td>
<td>CLSRPOL</td>
</tr>
<tr>
<td>CLSRPOLB</td>
<td>CLSRPOLQ</td>
<td>CMQCONND</td>
</tr>
<tr>
<td>CMQCONNQ</td>
<td>CMQCONNR</td>
<td>CMRDETAQ</td>
</tr>
<tr>
<td>CMRDETAZ</td>
<td>CMRPRBTQ</td>
<td>CMRTTTHR</td>
</tr>
<tr>
<td>CMRTTHRE</td>
<td>CMRTTHRT</td>
<td>COBJ</td>
</tr>
<tr>
<td>COBJR</td>
<td>COBJRZ</td>
<td>COBJS</td>
</tr>
<tr>
<td>COBJSZ</td>
<td>COBJZ</td>
<td>CPIPLNED</td>
</tr>
<tr>
<td>CPIPLNEQ</td>
<td>CPIPLNER</td>
<td>CPROG</td>
</tr>
<tr>
<td>CPROGD</td>
<td>CPROGQ</td>
<td>CPROGR</td>
</tr>
<tr>
<td>CPROGRM</td>
<td>CPROGRMZ</td>
<td>CPROGZ</td>
</tr>
<tr>
<td>CRDELAYQ</td>
<td>CRDLAYGQ</td>
<td>CRDLAYTQ</td>
</tr>
<tr>
<td>CREGAGT</td>
<td>CREGDB2</td>
<td>CREGDB2D</td>
</tr>
<tr>
<td>CREGDB2Q</td>
<td>CREGDBC</td>
<td>CREGDBC2D</td>
</tr>
<tr>
<td>CREGDBCQ</td>
<td>CREGDETL</td>
<td>CREGDSA</td>
</tr>
<tr>
<td>CREGDSAS</td>
<td>CREGHR</td>
<td>CREGHRDZ</td>
</tr>
<tr>
<td>CREGHRHZ</td>
<td>CREGHRZ</td>
<td>CREGION</td>
</tr>
<tr>
<td>CREGIONR</td>
<td>CREGIONZ</td>
<td>CREGMQSQ</td>
</tr>
<tr>
<td>CREGMSGR</td>
<td>CREGOVER</td>
<td>CREGOVR</td>
</tr>
<tr>
<td>CREGSYA</td>
<td>CREGSYAD</td>
<td>CREGSYAZ</td>
</tr>
<tr>
<td>CREGSYS</td>
<td>CREGSYSD</td>
<td>CREGSYSQ</td>
</tr>
<tr>
<td>CREGSYSZ</td>
<td>CREGTDQQ</td>
<td>CREVD1</td>
</tr>
<tr>
<td>CREVD2</td>
<td>CREVD7</td>
<td>CREVDA</td>
</tr>
<tr>
<td>CREVDG</td>
<td>CREVDY</td>
<td>CREVDZ</td>
</tr>
<tr>
<td>CREVIEW</td>
<td>CRGNDASH</td>
<td>CSERVX</td>
</tr>
<tr>
<td>CSESSQ</td>
<td>CTASKDA</td>
<td>CTASKDAP</td>
</tr>
<tr>
<td>CTASKD52</td>
<td>CTCPSRV</td>
<td>CTCPSRVD</td>
</tr>
<tr>
<td>CTCPSRVQ</td>
<td>CTCPSRVR</td>
<td>CTDQUEQ</td>
</tr>
</tbody>
</table>
### Changed views in version 6.6 of MainView for CICS

<table>
<thead>
<tr>
<th>CTDQUER</th>
<th>CTERM</th>
<th>CTERMD</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTERMD1</td>
<td>CTERMD2</td>
<td>CTERMQ</td>
</tr>
<tr>
<td>CTGHICIC</td>
<td>CTGHICICD</td>
<td>CTGHICICQ</td>
</tr>
<tr>
<td>CTGHICIZ</td>
<td>CTGHIST</td>
<td>CTGHIST1</td>
</tr>
<tr>
<td>CTGHIST2</td>
<td>CTGHIST3</td>
<td>CTGHIST4</td>
</tr>
<tr>
<td>CTGHIST5</td>
<td>CTGHISTA</td>
<td>CTGHISTD</td>
</tr>
<tr>
<td>CTGHISTM</td>
<td>CTGHISTO</td>
<td>CTGHISTQ</td>
</tr>
<tr>
<td>CTGHISTZ</td>
<td>CTGSERV</td>
<td>CTGSERVVL</td>
</tr>
<tr>
<td>CTGSERVQ</td>
<td>CTGSRVXT</td>
<td>CTIOT</td>
</tr>
<tr>
<td>CTRAN</td>
<td>CTRAND</td>
<td>CTRAND1</td>
</tr>
<tr>
<td>CTRANQ</td>
<td>CTRANR</td>
<td>CTRANSV</td>
</tr>
<tr>
<td>CTRANZ</td>
<td>CTRNCLAQ</td>
<td>CTRNCLAR</td>
</tr>
<tr>
<td>CTSMODLD</td>
<td>CTSMODLQ</td>
<td>CTSMODLR</td>
</tr>
<tr>
<td>CUOW</td>
<td>CUOWDSN</td>
<td>CUOWENQ</td>
</tr>
<tr>
<td>CUOWLNK</td>
<td>CUOWLNKD</td>
<td>CUOWQ</td>
</tr>
<tr>
<td>CUOWZ</td>
<td>CURIMAP</td>
<td>CURIMAPA</td>
</tr>
<tr>
<td>CURIMAPC</td>
<td>CURIMAPD</td>
<td>CURIMAPJ</td>
</tr>
<tr>
<td>CURIMAPP</td>
<td>CURIMAPQ</td>
<td>CURIMAPR</td>
</tr>
<tr>
<td>CURIMAPU</td>
<td>CURIMAPZ</td>
<td>CWEBSRVQ</td>
</tr>
<tr>
<td>CWEBSRVR</td>
<td>CWOVER</td>
<td>CWOVERS</td>
</tr>
<tr>
<td>EZCCTG</td>
<td>EZCFSSI</td>
<td>EZCSADMIN</td>
</tr>
<tr>
<td>EZCSHIST</td>
<td>EZCSOPS</td>
<td>EZCSREGN</td>
</tr>
<tr>
<td>EZCSRES</td>
<td>EZCSTSK</td>
<td>EZCSWKLD</td>
</tr>
<tr>
<td>SYSOVER</td>
<td>TASK</td>
<td>TASKCLAA</td>
</tr>
<tr>
<td>TASKCLAQ</td>
<td>TASKCONQ</td>
<td>TASKD</td>
</tr>
<tr>
<td>TASKLCK</td>
<td>TASKLCKD</td>
<td>TASKLCKG</td>
</tr>
<tr>
<td>TASKLCKQ</td>
<td>TASKQ</td>
<td>TASKSESQ</td>
</tr>
<tr>
<td>TASKTDQQ</td>
<td>TASKTSQQ</td>
<td>TASKZ</td>
</tr>
<tr>
<td>THINTQ</td>
<td>TSQUECFD</td>
<td>TSQUECFQ</td>
</tr>
<tr>
<td>TSQUEUE</td>
<td>TSQUEUEUD</td>
<td>TSQUEUEEQ</td>
</tr>
</tbody>
</table>
Release-specific considerations for version 6.5

This section describes migration considerations that are specific to version 6.5 of the MainView for CICS product.

Changes in the CREVIEW family of views

The CREVIEW family of views provides information about the CICS System Initialization Table (SIT).

This family of views has been substantially updated and includes all SIT parameters for CICS TS versions through 5.1.

In addition, all unsupported SIT parameters from versions earlier than CICS TS version 3.1 have been eliminated. If your site has customized CREVIEW views, you should carefully review these customizations.

Changed default fields for summary views

For the following summary views, this release changed the default field that the view uses as the basis for the summary:

<table>
<thead>
<tr>
<th>View</th>
<th>Data is summarized by</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDOCTMPZ</td>
<td>Document Template name</td>
</tr>
<tr>
<td>CIPCONNZ</td>
<td>IP Connection name</td>
</tr>
<tr>
<td>CPIPLNEZ</td>
<td>Pipeline name</td>
</tr>
<tr>
<td>CSESSZ</td>
<td>Session ID</td>
</tr>
<tr>
<td>CTCPSRVZ</td>
<td>TCPIP Service ID</td>
</tr>
<tr>
<td>CTDQUEZ</td>
<td>TD Queue name</td>
</tr>
<tr>
<td>CURIMAPZ</td>
<td>URI Map name</td>
</tr>
<tr>
<td>CWEBSRVZ</td>
<td>Web Service name</td>
</tr>
</tbody>
</table>

Consequently, the views’ returned data might look different from data returned in the previous release. You should review any alarm definitions that you might have created for these views and modify them as needed.
New views to replace the CREGDB2*, CREGMQS* and CREGSYS* views

Version 6.5.00 of the MainView for CICS product introduces the following new views to replace earlier views:

- The CDB2CON* views replace the CREGDB2* view.
- The CMQCONN* views replace the CREGMQS* views.
- The CREGSYT*, CREGSYA*, and CREGSYM* views replace the CREGSYS* views.

The new views contain more information than the replaced views. For compatibility, the earlier views are still available if you enter their names on the COMMAND line. However, MainView for CICS will remove support for these views in a future release. If you have alarm definitions based on these views, you should modify them to use the new views.

Changes to the task kill exit

The task kill exit includes the following changes in version 6.5.00 of the MainView for CICS product.

Reworded messages

The task kill exit now issues "threshold exceeded" messages to the CICS job log, the BBI-SS PAS journal, or both when you specify WTO=CON/LOG/BOTH in the task kill exit threshold table (CMRTTHnn).

The following messages now include the name of the program (pgm) that was in use when the task exceeded the threshold:

FT700 TASK name #nnnnn userid A/T=code pgm CPUTIME value > threshold
FT710 TASK name #nnnnn userid A/T=code pgm FILECALL value > threshold
FT720 TASK name #nnnnn userid A/T=code pgm DBCALL value > threshold
FT730 TASK name #nnnnn userid A/T=code pgm STGABOVE value > threshold
FT740 TASK name #nnnnn userid A/T=code pgm STGBELOW value > threshold
FT750 TASK name #nnnnn userid A/T=code pgm STGTOTAL value > threshold
FT760 TASK name #nnnnn userid A/T=code pgm ELAPSED value > threshold
FT770 TASK name #nnnnn userid A/T=code pgm DB2 value > threshold
FT780 TASK name #nnnnn userid A/T=code pgm DBCTL value > threshold

To ensure that automation continues to work as expected, review and modify any automation processes that rely on the content and format of these messages.

**Added exclusions from the task kill exit**

The task kill exit now excludes transactions running in programs that begin with EYU (CPSM modules), EQA (IBM Debug Tool), and EZACIC (CICS Sockets Exits) from resource limitation.

Lastly, the exit will not start in a CPSM CMAS or WUI CICS region.

**Changes to command messages**

The following command result messages have changed message numbers, message format, or both.

**Earlier messages**

**FT4801** SETDMPDS COMPLETE xxxxxxxx VALUE IS yyyyyyyy
**FT480E** SETDMPDS FAILED xxxxxxxx VALUE IS yyyyyyyy zzzzzzzzzzzzzz
**FT4801** SETSYS COMPLETE xxxxxxxx VALUE IS yyyyyyyy
**FT480E** SETSYS FAILED xxxxxxxx VALUE IS yyyyyyyy zzzzzzzzzzzzzz
**FT4741** SETWEB COMPLETE xxxxxxxx VALUE IS yyyyyyyy
**FT474E** SETWEB FAILED xxxxxxxx VALUE IS yyyyyyyy zzzzzzzzzzzzzz
**FT4751** SETTCP COMPLETE xxxxxxxx VALUE IS yyyyyyyy
**FT475E** SETTCP FAILED xxxxxxxx VALUE IS yyyyyyyy zzzzzzzzzzzz

**New messages**

**FT5711** SETDMPDS COMPLETE condition keyword(value) reason
FT571E SETDMPDS FAILED condition keyword(value) reason
FT572I SETSYS COMPLETE condition keyword(value) reason
FT572E SETSYS FAILED condition keyword(value) reason

FT474I SETWEB COMPLETE condition keyword(value) reason
FT474E SETWEB FAILED condition keyword(value) reason
FT475I SETTCP COMPLETE condition keyword(value) reason
FT475E SETTCP FAILED condition keyword(value) reason

For more information about the condition, keyword, and reason values in these messages, see the online MSG facility.

To ensure that automation (such as Rules processing) continues to work as expected, review and modify any automation processes that rely on the content and format of these messages.

Changes to the CMRSTATS program

The CMRSTATS program uses the same TYPE number for its records as its equivalent CICS statistic records.

The following changes are made to the CMRSTATS program in version 6.5.00 of the MainView for CICS product to support modifications in CICS TS version 5.1:

- CICS Statistics type 3C (dispatcher) now uses statistics type 3F.
  MainView for CICS still uses the type 3C records for compatibility with earlier releases.

- The following CICS Statistics types are dropped in this release:
  - 6F (Request Model)
  - x'72' (CORBA Server)
  - x'73' (EJB)
  - x'75' (JVMPOOL)
  - x'76' (JVMPROFILE)
Changes to the CMRCMPWN program

In version 6.5.00 of the MainView for CICS product, the CMRCMPWN batch decompression program no longer supports the following specification:

- FORMAT=2.2 or 2.2Y
- FORMAT=2.3 or 2.3Y

The CMRCMPWN program has been enhanced to accept the following specification:

- FORMAT=5.1 or 5.1Y

Changes to the T6E fields

The following table lists the changes made to the exception wait field values in version 6.5.00 of the MainView for CICS product:

Table 1: Changes for the T6E fields in version 6.5.00 of the MainView for CICS product

<table>
<thead>
<tr>
<th>Field name</th>
<th>Value in earlier versions</th>
<th>New value in version 6.5.00</th>
<th>Comments</th>
</tr>
</thead>
</table>
| T6EVSTWT   | File string exception wait time + file buffer exception wait time | File string exception wait time | Two new fields replaced the buffer exception wait time:
  - LSR buffer exception wait time (T6EXLBT)
  - LSR string exception wait time (T6EXLST) |
| T6EVSTWC   | File string exception wait count + file buffer exception wait count | File string wait exception count | Two new fields replaced the buffer exception wait count:
  - LSR buffer exception wait count (T6EXLBC)
  - LSR string exception wait count (T6EXLSC) |
| T6ETSWK    | TS (temporary storage) string exception wait time + TS buffer exception wait time | TS string exception wait time | New field added for TS buffer exception wait time: T6EXTBT |
| T6ETSWC    | TS string wait count + TS buffer wait count | TS string wait count | New field added for TS buffer exception wait count: T6EXTBC |
Release-specific considerations for version 6.4

This section describes migration considerations that are specific to version 6.4 of the MainView for CICS product.

Audit data for parameter updates

In MainView for CICS 6.4.00, the administrative views and their corresponding tables have been updated to record consistent audit information when the definitions are modified.

The following views and tables are affected:

- CMRDETL/CMRDLTLnn for task history file definitions
- CMRPRBT/CMRPRBnn for problem threshold table definitions
- CMRSOPT/CMRSOPnn for startup options table definitions
- CMRTTTHR/CMRTTHnn for task kill exit threshold table definitions

The following audit data is available in the tabular views:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update User</td>
<td>User ID that last modified the definition</td>
</tr>
<tr>
<td>Update Date</td>
<td>Date of the last modification</td>
</tr>
<tr>
<td>Upd Time</td>
<td>Time of the last modification</td>
</tr>
<tr>
<td>Upd Sys</td>
<td>System on which the modification was made</td>
</tr>
</tbody>
</table>
Note

- Audit data is not available for changes made prior to version 6.5.00.
  If an administrative table is modified under version 6.4.00, that table is no longer compatible with earlier releases of MainView for CICS unless the appropriate toleration PTF is applied:
  - BPC5499 for version 6.3.xx
  - BPC5500 for version 6.2.00
  - BPC5501 for version 6.1.00
- If an administrative table is modified under version 6.4.00, that table should not be subsequently modified by any earlier version of the product; otherwise, all audit data will be lost, even if the compatibility PTF has been applied.

New record types in CMRSTATS

The MainView for CICS CMRSTATS program uses the same type number for its records as the equivalent CICS statistic records.

Version 6.4.00 includes the following new record types that were introduced in CICS TS 4.2:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Migration notes</th>
</tr>
</thead>
</table>
| 13   | Storage Manager Domain Subpool Field prefix: DS7 | - This record is equivalent to a Type 05 record for CICS TS 4.1 and earlier. If you have a CMRPRL program that extracts Type 05 records, you should change the program to extract Type 13 records for CICS TS 4.2 data.
- The prefix of the fields in this record has changed to DS7.
- The Type 13 record contains one new field that was not in the Type 05 record. |
### Type 14

**Storage Manager Task Subpool**

**Field prefix:** DT7

- This record is equivalent to a Type 06 record for CICS TS 4.1 and earlier. If you have a CMRPRL program that extracts Type 06 records, you should change the program to extract Type 14 records for CICS TS 4.2 data.
- The prefix of the fields in this record has changed to DT7.

### Type 1D

**Storage Manager DSA**

**Field prefix:** D42

- This record is equivalent to a Type 0E record for CICS TS 3.2 through 4.1. If you have a CMRPRL program that extracts Type 0E records, you should change the program to extract Type 1D records for CICS TS 4.2 data.
- The prefix of the fields in this record has changed to D42.
- The Type 1D record contains several new fields that were not in the Type 0E record.

### Type 90

**Event Processor Adapter**

**Field prefix:** EP

This new record is available for CICS TS 4.1 and later.

Version 6.4.00 also includes new fields that were added to the following CICS statistic records: Type 30, 66, 67, 6A, 6C, 6E, 71, 74, 78, 8C, 8D, and 8E.

---

**Monitor message changes**

The resolution messages issued by the @ACBN, @DB2N, and @IMSN monitors were modified in version 6.4.00.

The suffix for these messages was changed from I to Z so the messages do not appear in the current problem view (CREGPRB) when there is no problem.

The updated messages are as follows:

- `FT517Z DB2 INTERFACE AVAILABLE FOR DB2 ID (nnnn)`
- `FT518Z IMS INTERFACE AVAILABLE FOR IMS ID (nnnn)`
- `FT519Z VTAM ACB IS OPEN`
If you have defined automation based on these message IDs, you should modify the automation accordingly.

### Obsolete views

The TASKX, TASKZX, and TASKD1-7 views have been obsolete since the introduction of CTASK family of views in version 5.7 of MainView for CICS. These views are no longer accessible.

### DUMP display changes

The full-screen DUMP display was enhanced in version 6.4.00 to:

- Display 64-bit addresses for CICS_TS 4.2 regions
- Display data in mixed case

#### Displaying 64-bit addresses

To distinguish between a 31-bit address and a 64-bit address in the body of the dump, the PF11 key is assigned to JUMP64 in MainView for CICS full-screen displays. To display a 64-bit address in the body of the dump, press PF11 (instead of Enter) to display the address represented by the next eight bytes from the cursor position.

**Note**

- **PF11** is not normally used in full-screen displays other than the DUMP display. However, if you have assigned **PF11** to another function, you might have to change that assignment to another PF key.
- The assignment of **PF11** was not changed in the full-screen LOG display.

#### Displaying and modifying data in mixed case

Because data in the DUMP display now appears in mixed case, modifications to virtual storage in the dump output are case-sensitive.
Considerations for any new release

This section describes issues that you should consider when migrating to any new release of MainView for CICS.

Report program considerations

If you use any of the sample batch report programs provided by MainView for CICS, or have written your own report programs using SAS, PRL, or COBOL, the following rules must be followed:

- All COBOL report programs must be re-compiled and relink-edited by using the new version of MainView for CICS libraries. This step ensures that new MainView for CICS programs and copy books are included in the report programs.

- All COBOL, SAS, and PRL report programs must use the new MainView for CICS libraries.

CMRCUEX considerations

User programs that include user exit interface CMRCUEX, CMRCCUEX or CMRUFOX must be relinked with the new version of the modules.

For more information about these modules, see “User exit interface (CMRCUEX)” on page 255.

CICS Transaction Gateway support

MainView for CICS provides support for the IBM CICS Transaction Gateway (CTG) product.

To enable CTG monitoring in your applications, you must modify both the MainView for CICS and CTG environments as described in “Monitoring the CICS Transaction Gateway” on page 171.

During the installation of MainView for CICS, a UNIX file called BDRHFS is created to hold the executable code that supports CTG monitoring. This file, which is used in the CTG address space, should not be combined with any other UNIX file.
Updated views might issue messages

A new version of MainView for CICS might add elements (fields) to an existing view.

If you use such a view with a context of multiple PASs running different MainView for CICS versions, you might see messages similar to the following:

BBMXV334I Target $SSI on system xxx contains 1 un-resolved elements
BBMXV335I Message(s) received from targets on xxx
BBMXBD19I There were invalid elements found in form FORM00

Related:BBMXB422W The xxxxxx element does not exist in any of the source maps
BBMXV336I --- End of message(s) from targets on xxx

After the messages are displayed, all available data might or might not be displayed in the view and you might get some additional error messages. The result depends on the release level of the BBI-SS PAS that owns the first target which MainView Infrastructure selects from the SSI context. However, if the returned view is an updated view, the new element does not display data for PASs that are running earlier versions of the product.

Note
To avoid these messages, use a context that includes only targets from BBI-SS PASs running the same release of MainView for CICS.

New views not downward-compatible

New views that are introduced in a new version of MainView for CICS are not downward-compatible.
They should not be used with a context of multiple PASs running different MainView for CICS versions; the results could be unpredictable. You might see messages similar to the following:

```
BBMXBG16E Query select failed
-Related:BBMXBE06E Failed to allocate a Join Object for Query.
---Related:BBMXC016E The Allocate Function for Join returned - RC: 8
----Related:BBMXB606E Failed to allocate DataRecord Object
------Related:BBMXC016E The Allocate Function for DataRecord returned - RC: 8
-------Related:BBMXB505E Failed to get DataManagerStep data
--------Related:BBMXC016E The Allocate Function for Selector returned - RC: 8
---------Related:BBMXCP04E Table xxxxxxxx Not Found in Selector Dictionary.
----------Related:BBMXCLO5E Selector xxxxxxxx Not Found
```

These or similar messages will be displayed for every target in the context that is not running with the new release of MainView for CICS. When you press Enter after all the messages, the view might or might not display any available data. The result depends on the release level of the BBI-SS PAS that owns the first target which MainView Infrastructure selects from the SSI context.

**Note**

To avoid these messages, use a context that includes only targets from BBI-SS PASs running the new release of MainView for CICS.

### Updating or switching the BBI-SS PAS to different releases

If a BBI-SS PAS with the same SSID on the same IBM z/OS image is upgraded to a different MainView for CICS release, or if you need to revert to an older release, you must either re-IPL the IBM z/OS image with the desired libraries or use the following procedure:

1. Bring down all the MainView for CICS and MainView AutoOPERATOR for CICS BBI-SS PASs that are running in the same LPAR at the same time.

2. Restart the affected BBI-SS PAS by using the same libraries with the following parameter:

   **START=FREE**

   START=FREE releases any control blocks held by the existing BBI-SS PAS. The BBI-SS PAS starts and then terminates immediately after the storage is released.

3. Change the BBI-SS PAS JCL to use the new libraries.
4 If the following conditions are true, you do not have to recycle your CICS regions:

- The latest Release Notes say you do not have to recycle CICS.
- You can replace the contents of BBLINK in DFHRPL without having to recycle CICS.

Otherwise, perform the following actions:

- Bring down all the CICS regions that are connected to the affected BBI-SS PAS.
- Change the BBLINK specification in DFHRPL to point to the correct BBLINK.

5 Start the new BBI-SS PAS with the following parameter:

```
START=COLD
```

START=COLD obtains new control blocks for the new BBI-SS PAS.

6 Restart the CICS regions that are connected to the affected BBI-SS PAS (if you brought them down in Step 4 on page 44).
How product libraries should be used

Several distributed libraries are included with MainView for CICS, including a parameter library (BBPARM), a samples library (BBSAMP), and a profile library (BBPROF).

Use the contents of these distributed libraries as models to create site-customized product libraries, either manually or with MainView Customization.

**WARNING**
The distributed libraries themselves should not be modified. If you change the distributed libraries, subsequent SMP maintenance will overwrite your changes.

Throughout the MainView documentation set, references to these libraries use the distributed name. However, when you need to make changes, be sure to use the corresponding library that has been customized for your site. Table 2 on page 45 lists the distributed name and the corresponding customized library created by MainView Customization, and leaves space for you to note any library you have created.

**Table 2: Product libraries**

<table>
<thead>
<tr>
<th>Distributed library name</th>
<th>Library created by MainView Customization</th>
<th>Other site-customized copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBPARM</td>
<td>UBBPARM</td>
<td></td>
</tr>
<tr>
<td>BBSAMP</td>
<td>UBBSAMP</td>
<td></td>
</tr>
<tr>
<td>BBPROF</td>
<td>SBBPROF</td>
<td></td>
</tr>
</tbody>
</table>

For more information about all the product libraries, see the discussion of MainView product libraries in the *MainView Customization Reference* or the *MainView Administration Guide*.
Auto maintenance facility

The auto maintenance facility enables most software upgrades to be applied to MainView for CICS without restarting target CICS systems. Implementing auto maintenance requires one-time modifications to the CICS CSD, PLT and SIT, and to the BBI-SS JCL, as described in “Completing implementation” on page 110. After these modifications are complete, you can use the auto maintenance facility to apply updates and patches to MainView for CICS; modifications to the CSD, PLT, and SIT will no longer be required when you upgrade to a new release of CICS or MainView for CICS.

Security considerations related to auto maintenance are discussed in “Site-specific security” on page 107.

For a list of the MainView AutoOPERATOR for CICS functions that are available when communications between a BBI-SS PAS and CICS is established, see “Additional MainView AutoOPERATOR for CICS functions” on page 219.

Using the auto maintenance facility

The auto maintenance facility provides INIT and TERM functions that refresh MainView for CICS modules without affecting CICS itself.

The INIT and TERM functions can be initiated through commands or at BBI-SS PAS initialization and termination.
Note

If either of the following modules is modified by maintenance or a new release of MainView for CICS, target CICS systems will need to be restarted:

- **CMRTRUEX**, which runs as a task-related user exit (TRUE) in CICS. After TERM processing ends, CMRTRUEX must remain running to free storage allocated for any task that started when the extractor was on, but ended after TERM processing completed.

- **OLTFSET**, which processes the INIT and TERM functions. This module must remain in the system since it cannot refresh itself.

If either module is changed by a PTF, the PTF will contain an ACTION ++HOLD statement indicating that the CICS system must be restarted to implement the maintenance.

If either module is changed in a new release, the Release Notes will indicate that CICS systems must be restarted to implement the new release.

The TERM function terminates all active MainView for CICS and MainView AutoOPERATOR for CICS functions and removes those MainView for CICS resources in CICS that were created by the INIT function. The TERM function stores the MainView processing environment in a status record and writes it to a CICS temporary storage queue (as described in “CICS temporary storage and enqueue name conflicts” on page 225).

The INIT function re-creates all resources and restarts all functions according to the status record. The only exception is if the TERM function is issued when the CICS target is not connected to a BBI-SS PAS. When no PAS connection is in effect, no agents are active. Consequently, the subsequent INIT function initiates agents based on the settings in CMRSOPT and not based on the TSQ status record indicators.

For additional information, see the following sections:

- Implementing the auto maintenance facility is described in “Initiating auto maintenance” on page 49.

- CSD definitions that are required by the auto maintenance facility are described in “Auto maintenance resources and messages” on page 49.

- The resources that are managed by the auto maintenance facility are described in “Managed resources” on page 50.

- The JCL members that are used to set up the auto maintenance facility are listed in Table 12 on page 57.
Initiating auto maintenance

Auto maintenance can be started in several ways:

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

- at BBI-SS PAS restart
  When the BBI-SS PAS terminates, it is detected by MainView for CICS in the CICS system and a TERM function is initiated. When the BBI-SS PAS restarts, a request to INIT MainView for CICS is sent to the target CICS systems through the External CICS Interface facility (EXCI). The EXCI facility is also used to send the action commands from the CREGAGT view to the CICS system.

- by issuing the FST2 TERM/INIT sequence from the console, a CICS terminal, or MainView AutoOPERATOR EXEC
  For more information, see “Controlling functions manually” on page 229.

- by issuing the FST2 RES sequence from the console, a CICS terminal, or MainView AutoOPERATOR EXEC

- when MainView for CICS is started in the program list table (PLT)

Auto maintenance resources and messages

The following sections describe the applicable resources and messages.

CSD definitions

The following CSD definitions are required by MainView for CICS.

Programs

- OLTFSSET definition for PLT and FST2.

- CMRVCMR verifies MainView for CICS presence.
Transactions

- FST2 (MainView for CICS initialization and control)
- BMCE (transaction started by EXCI through link request from the BBI-SS PAS)

Connection and session definitions

User-defined EXCI connection and session definitions are required. See “Step 3: Create EXCI connection and session definitions” on page 114 for information.

Note

The EXCI facility enables a BBI-SS PAS to initiate the link to a CICS system that was connected to the PAS before the PAS was restarted.

The EXCI facility is also required for most of the line commands in the CREGAGT view (see “CREGAGT view commands” on page 267) and the task kill exit (see “Implementing the task kill function” on page 155).

Managed resources

The following tables describe the transactions and programs that are created and discarded by MainView for CICS. These tables list the definition parameters for transaction and program resources along with the values defined for them.

Note

For a list of transaction and program descriptions, see “Resource descriptions and conflicts” on page 243. You should verify that the transaction IDs do not conflict with existing IDs.

Table 3: MainView for CICS transactions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>BCRT</th>
<th>FCD2</th>
<th>JNL2</th>
<th>SMN2</th>
<th>FIC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTION</td>
<td>backout</td>
<td>backout</td>
<td>backout</td>
<td>backout</td>
<td>backout</td>
</tr>
<tr>
<td>CMDSEC</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>CONFDATA</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>DTIMOUT</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>DUMP</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>DYNAMIC</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>ISOLATE</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>PRIORITY</td>
<td>240</td>
<td>255</td>
<td>255</td>
<td>255</td>
<td>1</td>
</tr>
</tbody>
</table>

Auto maintenance resources and messages
<table>
<thead>
<tr>
<th>Parameter</th>
<th>BCRT</th>
<th>FCD2</th>
<th>JNL2</th>
<th>SMN2</th>
<th>FIC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROFILE</td>
<td>DFHCICST</td>
<td>DFHCICST</td>
<td>DFHCICST</td>
<td>DFHCICST</td>
<td>DFHCICST</td>
</tr>
<tr>
<td>Program</td>
<td>BBK1CRTR</td>
<td>CMRFCMDx</td>
<td>OLTJRNLx</td>
<td>CMRTMON</td>
<td>CMRROLLx</td>
</tr>
<tr>
<td>RESSEC</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>RESTART</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>ROUTABLE</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>RUNAWAY</td>
<td>system</td>
<td>system</td>
<td>system</td>
<td>system</td>
<td>system</td>
</tr>
<tr>
<td>SHUTDOWN</td>
<td>disabled</td>
<td>disabled</td>
<td>disabled</td>
<td>disabled</td>
<td>disabled</td>
</tr>
<tr>
<td>SPURGE</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>STATUS</td>
<td>enabled</td>
<td>enabled</td>
<td>enabled</td>
<td>enabled</td>
<td>enabled</td>
</tr>
<tr>
<td>STORAGECLEAR</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>TASKDATAKEY</td>
<td>CICS</td>
<td>CICS</td>
<td>CICS</td>
<td>CICS</td>
<td>CICS</td>
</tr>
<tr>
<td>TASKDATALOC</td>
<td>any</td>
<td>any</td>
<td>any</td>
<td>any</td>
<td>any</td>
</tr>
<tr>
<td>TPURGE</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>TRACE</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>TRANCLASS</td>
<td>DFHTCL00</td>
<td>DFHTCL00</td>
<td>DFHTCL00</td>
<td>DFHTCL00</td>
<td>DFHTCL00</td>
</tr>
<tr>
<td>TWASIZE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>WAIT</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>WAITTIME</td>
<td>0,0,0</td>
<td>0,0,0</td>
<td>0,0,0</td>
<td>0,0,0</td>
<td>0,0,0</td>
</tr>
</tbody>
</table>

Table 4: MainView for CICS programs BALPUTLV—BBK1rrrx

<table>
<thead>
<tr>
<th>Parameter</th>
<th>BALPUTLV</th>
<th>BBK1CCE</th>
<th>BBK1CRTR</th>
<th>BBK1rrrx</th>
</tr>
</thead>
<tbody>
<tr>
<td>LANGUAGE</td>
<td>Assembler</td>
<td>Assembler</td>
<td>Assembler</td>
<td>Assembler</td>
</tr>
<tr>
<td>RELOAD</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>RESIDENT</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>USAGE</td>
<td>normal</td>
<td>normal</td>
<td>normal</td>
<td>normal</td>
</tr>
<tr>
<td>USELPACOPY</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>STATUS</td>
<td>enabled</td>
<td>enabled</td>
<td>enabled</td>
<td>enabled</td>
</tr>
<tr>
<td>CEDF</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>DATALOCATION</td>
<td>below</td>
<td>any</td>
<td>any</td>
<td>any</td>
</tr>
<tr>
<td>EXECKEY</td>
<td>CICS</td>
<td>CICS</td>
<td>CICS</td>
<td>CICS</td>
</tr>
<tr>
<td>CONCURRENCY</td>
<td>QUASIRENT</td>
<td>QUASIRENT</td>
<td>QUASIRENT</td>
<td>QUASIRENT</td>
</tr>
</tbody>
</table>
BBK1rrrx represents programs that collect data for views in windows mode. The variable rrr represents the record ID, and x represents a suffix based on the CICS release. For more information, see “Resource descriptions and conflicts” on page 243.

### Table 5: MainView for CICS programs CMR9RLSE—CMRKCPXx

<table>
<thead>
<tr>
<th>Parameter</th>
<th>CMR9RLSE</th>
<th>CMRCMPXx</th>
<th>CMRFCMDx</th>
<th>CMRINT2</th>
<th>CMRKCPXx</th>
</tr>
</thead>
<tbody>
<tr>
<td>LANGUAGE</td>
<td>Assembler</td>
<td>Assembler</td>
<td>Assembler</td>
<td>Assembler</td>
<td>Assembler</td>
</tr>
<tr>
<td>RELOAD</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>RESIDENT</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>USAGE</td>
<td>normal</td>
<td>normal</td>
<td>normal</td>
<td>normal</td>
<td>normal</td>
</tr>
<tr>
<td>USELPACOPY</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>STATUS</td>
<td>enabled</td>
<td>enabled</td>
<td>enabled</td>
<td>enabled</td>
<td>enabled</td>
</tr>
<tr>
<td>CEDF</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>DATALOCATION</td>
<td>any</td>
<td>any</td>
<td>any</td>
<td>any</td>
<td>any</td>
</tr>
<tr>
<td>EXECKEY</td>
<td>CICS</td>
<td>CICS</td>
<td>CICS</td>
<td>CICS</td>
<td>CICS</td>
</tr>
<tr>
<td>CONCURRENCY</td>
<td>QUASIRENT</td>
<td>QUASIRENT</td>
<td>QUASIRENT</td>
<td>QUASIRENT</td>
<td>QUASIRENT</td>
</tr>
<tr>
<td>DYNAMIC</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>EXECUTIONSET</td>
<td>FULLAPI</td>
<td>FULLAPI</td>
<td>FULLAPI</td>
<td>FULLAPI</td>
<td>FULLAPI</td>
</tr>
<tr>
<td>JVM</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

x represents a suffix based on the CICS release.

### Table 6: MainView for CICS programs CMRROLLx—CMRTRUEX

<table>
<thead>
<tr>
<th>Parameter</th>
<th>CMRROLLx</th>
<th>CMRSLOGx</th>
<th>CMRTMAP</th>
<th>CMRMON</th>
<th>CMRTRUEx</th>
<th>CMRTRUEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>LANGUAGE</td>
<td>Assembler</td>
<td>Assembler</td>
<td>Assembler</td>
<td>Assembler</td>
<td>Assembler</td>
<td>Assembler</td>
</tr>
<tr>
<td>RELOAD</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

x represents a suffix based on the CICS release.
### Table 7: MainView for CICS programs CMRXEIO x—a—OLTCCRE

<table>
<thead>
<tr>
<th>Parameter</th>
<th>CMRXEIO x</th>
<th>CMRXEIST</th>
<th>CMRXTFG</th>
<th>CSLOAD</th>
<th>CSLOADC</th>
<th>OLTCCRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>LANGUAGE</td>
<td>Assembler</td>
<td>Assembler</td>
<td>Assembler</td>
<td>Assembler</td>
<td>Assembler</td>
<td>Assembler</td>
</tr>
<tr>
<td>RELOAD</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>RESIDENT</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>USAGE</td>
<td>normal</td>
<td>normal</td>
<td>normal</td>
<td>normal</td>
<td>normal</td>
<td>normal</td>
</tr>
<tr>
<td>USELPACOPY</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>STATUS</td>
<td>enabled</td>
<td>enabled</td>
<td>enabled</td>
<td>enabled</td>
<td>enabled</td>
<td>enabled</td>
</tr>
<tr>
<td>CEDF</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>DATALOCATION</td>
<td>any</td>
<td>any</td>
<td>any</td>
<td>any</td>
<td>any</td>
<td>any</td>
</tr>
<tr>
<td>EXECKEY</td>
<td>CICS</td>
<td>CICS</td>
<td>CICS</td>
<td>CICS</td>
<td>CICS</td>
<td>CICS</td>
</tr>
<tr>
<td>CONCURRENCY</td>
<td>QUASIRENT</td>
<td>QUASIRENT</td>
<td>QUASIRENT</td>
<td>QUASIRENT</td>
<td>QUASIRENT</td>
<td>QUASIRENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(THREADSAFE for TS 2.2 and later)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Note: not discarded</td>
</tr>
<tr>
<td>DYNAMIC</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>EXECUTIONSET</td>
<td>FULLAPI</td>
<td>FULLAPI</td>
<td>FULLAPI</td>
<td>FULLAPI</td>
<td>FULLAPI</td>
<td>FULLAPI</td>
</tr>
<tr>
<td>JVM</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

\(x\) represents a suffix based on the CICS release.

---

Table 7: MainView for CICS programs CMRXEIO x—a—OLTCCRE

<table>
<thead>
<tr>
<th>Parameter</th>
<th>CMRXEIO x</th>
<th>CMRXEIST</th>
<th>CMRXTFG</th>
<th>CSLOAD</th>
<th>CSLOADC</th>
<th>OLTCCRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>LANGUAGE</td>
<td>Assembler</td>
<td>Assembler</td>
<td>Assembler</td>
<td>Assembler</td>
<td>Assembler</td>
<td>Assembler</td>
</tr>
<tr>
<td>RELOAD</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>RESIDENT</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>USAGE</td>
<td>normal</td>
<td>normal</td>
<td>normal</td>
<td>normal</td>
<td>normal</td>
<td>normal</td>
</tr>
<tr>
<td>USELPACOPY</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>STATUS</td>
<td>enabled</td>
<td>enabled</td>
<td>enabled</td>
<td>enabled</td>
<td>enabled</td>
<td>enabled</td>
</tr>
<tr>
<td>CEDF</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>DATALOCATION</td>
<td>any</td>
<td>any</td>
<td>below</td>
<td>any</td>
<td>below</td>
<td>any</td>
</tr>
<tr>
<td>EXECKEY</td>
<td>CICS</td>
<td>CICS</td>
<td>CICS</td>
<td>CICS</td>
<td>CICS</td>
<td>CICS</td>
</tr>
</tbody>
</table>
### Table 8: MainView for CICS programs OLTCTL—OLTVER

<table>
<thead>
<tr>
<th>Parameter</th>
<th>OLTCTL</th>
<th>OLTSET&lt;sup&gt;a&lt;/sup&gt;</th>
<th>OLTJRNL&lt;sup&gt;a&lt;/sup&gt;</th>
<th>OLTSCAO</th>
<th>OLTSMRX&lt;sup&gt;a&lt;/sup&gt;</th>
<th>OLTVER</th>
</tr>
</thead>
<tbody>
<tr>
<td>LANGUAGE</td>
<td>Assembler</td>
<td>Assembler</td>
<td>Assembler</td>
<td>Assembler</td>
<td>Assembler</td>
<td>Assembler</td>
</tr>
<tr>
<td>RELOAD</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>RESIDENT</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>USAGE</td>
<td>normal</td>
<td>normal</td>
<td>normal</td>
<td>normal</td>
<td>normal</td>
<td>normal</td>
</tr>
<tr>
<td>USELPACOPY</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>STATUS</td>
<td>enabled</td>
<td>enabled</td>
<td>enabled</td>
<td>enabled</td>
<td>enabled</td>
<td>enabled</td>
</tr>
<tr>
<td>CEDF</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>DATALOCATION</td>
<td>any</td>
<td>any</td>
<td>any</td>
<td>any</td>
<td>any</td>
<td>below</td>
</tr>
<tr>
<td>EXECKEY</td>
<td>CICS</td>
<td>CICS</td>
<td>CICS</td>
<td>CICS</td>
<td>CICS</td>
<td>CICS</td>
</tr>
<tr>
<td>CONCURRENCY</td>
<td>QUASIRENT</td>
<td>QUASIRENT</td>
<td>QUASIRENT</td>
<td>QUASIRENT</td>
<td>QUASIRENT</td>
<td>QUASIRENT</td>
</tr>
<tr>
<td>DYNAMIC</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>EXECUTIONSET</td>
<td>FULLAPI</td>
<td>FULLAPI</td>
<td>FULLAPI</td>
<td>FULLAPI</td>
<td>FULLAPI</td>
<td>FULLAPI</td>
</tr>
<tr>
<td>JVM</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

<sup>a</sup> x represents a suffix based on the CICS release.

### Table 9: MainView for CICS programs OLTACON, OLTACICS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>OLTACON</th>
<th>OLTACICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>LANGUAGE</td>
<td>Assembler</td>
<td>Assembler</td>
</tr>
<tr>
<td>RELOAD</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Parameter</td>
<td>OLTACON</td>
<td>OLTACICS</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>RESIDENT</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>USAGE</td>
<td>normal</td>
<td>normal</td>
</tr>
<tr>
<td>USELPACOPY</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>STATUS</td>
<td>enabled</td>
<td>enabled</td>
</tr>
<tr>
<td>CEDF</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>DATALLOCATION</td>
<td>any</td>
<td>any</td>
</tr>
<tr>
<td>EXECKEY</td>
<td>CICS</td>
<td>CICS</td>
</tr>
<tr>
<td>CONCURRENCY</td>
<td>QUASIRENT</td>
<td>QUASIRENT</td>
</tr>
<tr>
<td>API</td>
<td>CICSAPI</td>
<td>CICSAPI</td>
</tr>
<tr>
<td>DYNAMIC</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>EXECUTIONSET</td>
<td>FULLAPI</td>
<td>FULLAPI</td>
</tr>
<tr>
<td>JVM</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>JVMPROFILE</td>
<td>DFHJVMPR</td>
<td>DFHJVMPR</td>
</tr>
</tbody>
</table>

Table 10: MainView AutoOPERATOR for CICS transactions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>FCM1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTION</td>
<td>backout</td>
</tr>
<tr>
<td>CMDSEC</td>
<td>no</td>
</tr>
<tr>
<td>CONFDATA</td>
<td>no</td>
</tr>
<tr>
<td>DTIMOUT</td>
<td>no</td>
</tr>
<tr>
<td>DUMP</td>
<td>yes</td>
</tr>
<tr>
<td>DYNAMIC</td>
<td>no</td>
</tr>
<tr>
<td>ISOLATE</td>
<td>no</td>
</tr>
<tr>
<td>PRIORITY</td>
<td>1</td>
</tr>
<tr>
<td>PROFILE</td>
<td>DFHCICST</td>
</tr>
<tr>
<td>Program</td>
<td>CBQCMD</td>
</tr>
<tr>
<td>RESSEC</td>
<td>no</td>
</tr>
<tr>
<td>RESTART</td>
<td>no</td>
</tr>
<tr>
<td>ROUTABLE</td>
<td>no</td>
</tr>
<tr>
<td>RUNAWAY</td>
<td>system</td>
</tr>
<tr>
<td>SHUTDOWN</td>
<td>disabled</td>
</tr>
<tr>
<td>Parameter</td>
<td>FCM1</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>SPURGE</td>
<td>no</td>
</tr>
<tr>
<td>STATUS</td>
<td>enabled</td>
</tr>
<tr>
<td>STORAGECLEAR</td>
<td>no</td>
</tr>
<tr>
<td>TASKDATAKEY</td>
<td>CICS</td>
</tr>
<tr>
<td>TASKDATALOC</td>
<td>any</td>
</tr>
<tr>
<td>TPURGE</td>
<td>no</td>
</tr>
<tr>
<td>TRACE</td>
<td>yes</td>
</tr>
<tr>
<td>TRANCLASS</td>
<td>DFHTCL00</td>
</tr>
<tr>
<td>TWASIZE</td>
<td>0</td>
</tr>
<tr>
<td>WAIT</td>
<td>yes</td>
</tr>
<tr>
<td>WAITTIME</td>
<td>0,0,0</td>
</tr>
</tbody>
</table>

**Table 11: MainView AutoOPERATOR for CICS programs**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>CBQCMD</th>
<th>CMRTDPXx</th>
<th>OLTCAORL</th>
</tr>
</thead>
<tbody>
<tr>
<td>LANGUAGE</td>
<td>Assembler</td>
<td>Assembler</td>
<td>Assembler</td>
</tr>
<tr>
<td>RELOAD</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>RESIDENT</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>USAGE</td>
<td>normal</td>
<td>normal</td>
<td>normal</td>
</tr>
<tr>
<td>USELPACOPY</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>STATUS</td>
<td>enabled</td>
<td>enabled</td>
<td>enabled</td>
</tr>
<tr>
<td>CEDF</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>DATALLOCATION</td>
<td>any</td>
<td>any</td>
<td>any</td>
</tr>
<tr>
<td>EXECKEY</td>
<td>CICS</td>
<td>CICS</td>
<td>CICS</td>
</tr>
<tr>
<td>CONCURRENCY</td>
<td>QUASIRENT</td>
<td>QUASIRENT</td>
<td>QUASIRENT</td>
</tr>
<tr>
<td>DYNAMIC</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>EXECUTIONSET</td>
<td>FULLAPI</td>
<td>FULLAPI</td>
<td>FULLAPI</td>
</tr>
<tr>
<td>JVM</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>
Messages

MainView for CICS messages are documented in the BBMLIB data set and can be viewed by using:

- MESSAGES General Services option on the MainView for CICS Primary Option menu
- MSG command from the COMMAND line in windows mode

Note

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

Auto maintenance BBSAMP members

The following BBSAMP members are used to modify the CICS CSD resource groups and lists used by MainView for CICS.

The first five members contain information about how to use the members.

Table 12: BBSAMP members

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMRCSDAL</td>
<td>JCL used to create the CSD resource groups used by MainView for CICS or MainView AutoOPERATOR</td>
</tr>
<tr>
<td>CMRCSDEL</td>
<td>JCL used to delete resources from the BOOLEPPT and BOOLEPCT resource groups (these are old resource groups that might still exist in your CSD)</td>
</tr>
<tr>
<td>CMRCSDLA</td>
<td>JCL used to modify the group list that contains the resource groups installed at CICS initialization (MainView AutoOPERATOR only)</td>
</tr>
<tr>
<td>CMRCSDLB</td>
<td>JCL used to modify the group list that contains the resource groups installed at CICS initialization (MainView AutoOPERATOR and MainView for CICS)</td>
</tr>
<tr>
<td>CMRCSDLM</td>
<td>JCL used to modify the group list that contains the resource groups installed at CICS initialization (MainView for CICS only)</td>
</tr>
<tr>
<td>CSDCMRDF</td>
<td>contains the DEFINE statements for the required program and transaction resources for the MainView for CICS CICS component These definitions are for all releases of CICS.</td>
</tr>
<tr>
<td>CSDOLTDF</td>
<td>contains the DEFINE statements required for the program and transaction resources used by the MainView for CICS common component (for all releases of CICS)</td>
</tr>
</tbody>
</table>
BBI-SS PAS and CICS target connection

This chapter provides topics about the BBI-SS PAS and CICS target connection.

Connection facility

MainView for CICS establishes communications with target CICS regions using the connection facility. It is based on entries in the Jobname Node Table (JNT), which is allocated in private storage of the BBI-SS PAS.

This table is also used by various other BMC products. Entries are added to the JNT in the following ways:

- from the static elements contained in the UBBPARM member BBIJNT00
- from Dynamic Target Definition (TGTDEF) elements
- built at the request of the AutoConnect facility (recommended)

These sources, used separately or together, provide a wide range of options and methods to manage your CICS regions.

Note

You must specify that each CICS target connected to the PASplex with a unique jobname (or STC name) and alias. The BBI-SS PAS does not support duplicate target names. If you do not specify one in BBPARM member BBIJNT00 or in a dynamic target definition, the PASplex uses the CICS applid as an alias name for the CICS target. To workaround the duplicate target names restriction, see knowledge article KA428535.

The connection facility has two parts:

- Code that executes during MainView for CICS agent initialization on the CICS region requests the assignment of a BBI-SS PAS, or requests a specific BBI-SS PAS
by using a BBI-SS PAS directive, as described in “Providing a BBI-SS PAS
directive to a target CICS region” on page 64

■ Code that executes on the BBI-SS PAS

— manages the common request queue (this queue is shared by each BBI-SS PAS
running on an LPAR)

— processes the requests from the common request queue and assigns the
appropriate BBI-SS PAS based on a user requested control.

— manages the creation and deletion of the resources (control blocks, tasks, and
so on) that are needed to manage an assigned CICS region

The connection facility enables you to have as much or as little control over which
CICS connects to which BBI-SS PAS running in the same LPAR as you want. Control
is specified with parameter ACONCICS= in member BBISSP nn in UBBPARM.

**ACONCICS=NO (default)**

You have full control of PAS-CICS connections with manually created static or
dynamic target definitions and BBI-SS PAS directive.

The connection facility uses the target definitions to determine which CICS can
connect to this BBI-SS PAS, and the connection facility on the CICS side uses the
directive to know which BBI-SS PAS to connect. Static or dynamic target definitions
and a BBI-SS PAS directive are required. The directive and the target definitions
must have the same SSID.

**ACONCICS=YES**

The connection facility uses the AutoConnect facility to control connection between
the PAS and CICS.

Manually created static or dynamic target definitions and BBI-SS PAS directive plays
a major role in connection decision (see “AutoConnect affinity factors” on page 70).
A CICS that is not defined to the BBI-SS PAS by a static or dynamic target definition
can also connect to an available BBI-SS PAS running in the same LPAR. A BBI-SS
PAS directive is optional.
**Note**

A CONCICS=YES is highly recommended. It honors the static or dynamic target definition and the PAS directive, if provided. If no target definition or PAS directive is provided, A CONCICS=YES determines which BBI-SS PAS the CICS will connect to, based on the affinity factors described on “AutoConnect affinity factors” on page 70.

---

**A CONCICS=ONLY**

The connection facility gives the AutoConnect facility full control of connection decision.

Manually created static and dynamic target definitions are ignored and a BBI-SS directive is optional.

A CONCICS=ONLY requires the least manual setup and is recommended for the following situations:

- there is only one MainView *for CICS* BBI-PAS in an LPAR
- there is multiple MainView *for CICS* BBI-PASs in an LPAR but it does not matter to which PAS the CICS is connected
- there is a requirement to move CICS from one LPAR to another

If you specify A CONCICS=ONLY or A CONCICS=YES with no static or dynamic target definitions, a CASID=xxxx value must be specified in BBISSPxx, and a PAS connection to that CAS is required.

If you are running MainView AutoOPERATOR *for CICS* and using A CONCICS=ONLY or A CONCICS=YES with no static or dynamic target definition, all CICS commands that can be used in MainView AutoOPERATOR rules and execs are dependent, which means they will not be available until the CICS region target has established communications with the BBI-SS PAS.

---

**Using static BBIJNT00 entries**

JNT entries that are created from elements defined in the UBBPARM member BBIJNT00 are considered static definitions.

They are created when the BBI-SS PAS initializes and cannot be removed or modified for the life of the BBI-SS PAS. These entries override any of the dynamic types of JNT entries.
These elements should only be used to define CICS targets when the relationship between the CICS and the BBI-SS PAS being used to manage it does not change. This is because changing this relationship requires modification of the BBIJNT00 member on each BBI-SS PAS in the PAS-PLEX and restarting them.

Information on creating BBIJNT00 elements can be found in "Define BBI-SS PAS Suffixes and Target System Parameters" in the MainView Customization Reference.

Implementing dynamic target definitions

Dynamic target definitions enable you to move CICS region monitoring from one BBI-SS PAS to another without restarting the PAS.

To implement dynamic target support, you must do the following actions for each PAS that has access to the dynamic targets:

- remove static target definitions for targets that are going to be dynamic
- enable dynamic target support for each PAS
- associate each CICS region with a PAS
- create target definitions
- ensure each PAS has a CAS connection

For a discussion of managing targets and target contexts, see the MainView Administration Guide.

Removing static target definitions

Targets defined in the UBBPARM member BBIJNT00 are static definitions and override dynamic targets for CICS regions with the same name.

To prevent this override from happening, you must remove any TARGET statements that contain the TYPE=CICS parameter from member BBIJNT00.
Enabling dynamic target support

For each PAS that supports or has access to dynamic targets, add the following parameter to the UBBPARM member BBISSP00:

\[ \text{DYNTGT=\text{YES}} \]

This parameter enables dynamic targets. The default is DYNTGT=NO. If you do not specify DYNTGT=YES, dynamic targets are ignored.

Creating target definitions

To create target definitions, use the Plex Manager TGTDEF view and one of the following commands:

- **ADD** displays the ADD TARGET DEFINITION panel, which enables you to create or update a single target definition at a time
- **GENERATE** automatically creates a target definition for each target that appears on the TGTACT (PLEX) view with a status of Active

Dynamic target definitions are saved in BBPARM member BBMTXP00. They are appended to the end of an in-core list of JNT entries from BBIJNT00 when the dynamic target definition is installed (either when the CAS starts or when the TGTDEF INSTALL command is issued).

See the *MainView Administration Guide* for a description of the TGTDEF view and information about saving and installing dynamic target definitions. Ensure that each PAS has a CAS connection. Each PAS with access to dynamic targets must have a connection to a CAS. The connection is defined by parameter CASID=xxxx in UBBPARM member BBISSP00.

Special considerations

Note the following considerations when using dynamically defined targets:

- If multiple target definitions are defined for the same CICS region, the first one found in the combined BBIJNT00/BBMTXP00 in-memory list is used.
- Target definitions specified in BBIJNT00 override dynamic target definitions in BBMTXP00.
- Existing targets defined in BBIJNT00 cannot be modified from the TGTDEF view. If they are modified, the changes are ignored.
Historical (that is, CMRDETL) data is not available online for inactive CICS regions that are defined dynamically.

Providing a BBI-SS PAS directive to a target CICS region

When using static elements from the BBIJNT00 member or dynamic target definitions, you must identify the BBI-SS PAS for the target CICS region.

When using the AutoConnect facility, you can also identify the preferred BBI-SS PAS to the target CICS region. This process is referred to as providing a BBI-SS PAS directive to the CICS region. You can provide the BBI-SS PAS directive in one of two ways:

- Specify the system initialization (SIT) parameter INITPARM either in the SIT itself or in the CICS region's startup JCL. For example, you could specify the following value as a SIT override:

  \[ \text{INITPARM} = \text{OLTCNTL='SUBSYS='subsys'} \]

  and define \textit{subs}sys as the subsystem identifier for the PAS.

  \textit{Note}

  The \texttt{SUBSYS} parameter must be enclosed in double-single quotation marks (two single quotation marks at the beginning and end of the \texttt{SUBSYS} parameter) if specified in the \texttt{PARM} statement in the CICS JCL.

- Add BBIPARM DD in CICS JCL to point to the UBBPARM data set, and add a \texttt{TARGET} statement to one of these UBBPARM members:

<table>
<thead>
<tr>
<th>UBBPARM member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCICS00</td>
<td>This member can contain \texttt{TARGET} statements for CICS regions. The format of \texttt{TARGET} statements is the same as the \texttt{TARGET} statements in BBIJNT00. The connection facility on the CICS region scans the member during initialization. The BBI-SS PAS does not process BMCICS00. This member should be used for dynamic targets, mixed static and dynamic targets, and AutoConnect environments.</td>
</tr>
</tbody>
</table>
UBBPARM member | Description
--- | ---
BBIJNT00 | This member can contain TARGET statements for both CICS regions and the BBI-SS PAS. Targets defined in BBIJNT00 are considered static definitions; they override dynamic target definitions for CICS regions with the same name. This member should be used for static targets.

The search order that is used by the CICS region in determining its proper BBI-SS PAS directive is:

1. INITPARM value in CICS SIT overrides
2. Entry in BMCICS00 in UBBPARM
3. Entry in BBIJNT00 in UBBPARM

After a BMCICS00 member is found in the BBIPARM concatenation of the CICS region, the MainView for CICS agent will only look within the BMCICS00 member for target statements. It will not look at BBIJNT00 even if a target is not found in the BMCICS00 member. If you have mixed static and dynamic targets, define the static targets in this member also.

To change the BBI-SS PAS that monitors the CICS region, change the subsystem specified on the INITPARM or TARGET statement and restart the CICS region. Also restart the BBI-SS PAS if a static target is used.

The following example shows BBIPARM DD in CICS JCL:

//BBIPARM DD DISP=SHR, DSN=prefix.UBBPARM

**AutoConnect**

The Automatic Connection facility—AutoConnect—enables you to manage the connection between a CICS region and its managing BBI-SS PAS without manually creating and maintaining the definitions that relate the region to the BBI-SS PAS.

With AutoConnect a CICS region does not need prior knowledge of any BBI-SS PAS, nor does a BBI-SS PAS need prior knowledge of any CICS region that it is to manage.

AutoConnect is most effective for sites with a large number of CICS targets that are managed by any one of a number of BBI-SS PASs, such as sites with cloned AORs that are moved from LPAR to LPAR. For these sites, the Dynamic Targets facility requirement of creating and changing target definitions can prove unwieldy.
Using AutoConnect, BBI-SS PAS selection is initiated by a CICS region posting a request to a common queue that is serviced by the BBI-SS PASs running on an LPAR. A BBI-SS PAS determines the relative strength of each candidate BBI-SS PAS by applying a defined set of weights to affinity factors, such as the presence of a CMRDETL file, or a previous communications connection between the CICS region and the BBI-SS PAS.

When a candidate is selected, the CICS region waits while the selected BBI-SS PAS builds the required structures and establishes communications with the target CICS region.

**Note**

Due to the nature of the criteria used to select the appropriate BBI-SS PAS, some sites may find it beneficial to use a combination of AutoConnect and the less flexible dynamic or static JNT.

AutoConnect introduces the program resource OLTACICS. See “Managed resources” on page 50 for more information.

**AutoConnect BBI-SS PAS definitions**

The manual definitions that are used to connect a CICS region to a BBI-SS PAS have traditionally been maintained on the BBI-SS PAS in either the static BBIJNT00 parameter member or the more dynamic TGTDEFs.

When a static or dynamic JNT is used, a CICS region must be started with the managing BBI-SS PAS defined before it can initiate communications. This is accomplished either through an INITPARM parameter in the CICS SIT or a BBIJNT00/BMCICS00 parameter member in a BBIPARM DD statement in the CICS startup JCL (see “Step 1: Modify CICS startup JCL” on page 112).

- With the static JNT, the BBI-SS PAS must have a target entry for the CICS region in its BBIJNT00 parameter member and the in-core block generated by this entry is unchanged during the life of the BBI-SS PAS.

- With the dynamic JNT, the BBI-SS PAS must have a TGTDEF for the CICS region. See “Implementing dynamic target definitions” on page 62, the *MainView Administration Guide*, and the *MainView Customization Reference*.

In either case, you must indicate the BBI-SS PAS name to its target CICS regions in order to establish the communications path between the MainView for CICS agent running on the CICS region and the managing BBI-SS PAS.

AutoConnect provides compatibility with older versions of MainView for CICS as well as providing simple implementation rules for new users. To provide backward
compatibility, AutoConnect operates under two sets of conditions from within a CICS region:

■ The name of the managing BBI-SS PAS is manually defined. This implementation is standard under previous releases. Information on providing the name of the BBI-SS PAS to the CICS region can be found in “Step 1: Modify CICS startup JCL” on page 112.

■ No information on the name of the BBI-SS PAS is available to the CICS region.

When the MainView for CICS agent is started on the CICS region (through PLTPI, or the FST2 INIT or the FST2 QON transaction), it identifies the BBI-SS PAS by searching for an INITPARM value or a target entry in either the BMCICS00 or BBIJNT00 parameter member in an allocated BBIPARM library. The presence or absence of the BBI-SS PAS name drives agent processing down one of two paths. These two paths can be simplified as follows:

■ If a directive to use a specific BBI-SS PAS is found
  — The agent code searches for the subsystem entry for the target BBI-SS PAS. If those blocks are unavailable, the agent will wait until the target BBI-SS PAS becomes available.
  — Once the subsystem entry for the target BBI-SS PAS becomes available and active, the agent code running on the CICS region determines if the target BBI-SS PAS is participating in AutoConnect. If the target BBI-SS PAS is not participating in AutoConnect, the communications process continues in the static fashion. If the target BBI-SS PAS is participating in AutoConnect, the communications process falls through to the second path.

■ If a directive to use a specific BBI-SS PAS is not found
  — The agent code locates the common communications request queue and places a request that contains information about the CICS region that is used to determine the most eligible BBI-SS PAS.
  — The agent then waits for the reply from the request queue and if successful, the task ends and the agent goes dormant and waits for the designated BBI-SS PAS to finalize communications.

The communications requests are handled in a first-in, first-out basis by one of the BBI-SS PASs that are participating in AutoConnect. The BBI-SS PAS makes a determination based upon various affinity factors regarding which BBI-SS PAS is best suited to manage the CICS region. The affinity factors are described in “AutoConnect affinity factors” on page 70.

Regardless of the number of CICS regions or BBI-SS PASs residing on an LPAR, a single common connection request queue that is used by all of them. Figure 2 on
page 68 illustrates the relationship between multiple BBI-SS PASs, multiple CICS regions and the common connection request queue.

**Implementing AutoConnect**

This section describes how AutoConnect is implemented and the various factors used to determine the affinity between the CICS region and the available BBI-SS PASs.

**Initializing AutoConnect on the CICS region**

The AutoConnect facility is initiated automatically on the CICS region under the following conditions:

- The MainView for CICS agent code is unable to locate a directive to use a particular BBI-SS PAS from either an INITPARM (from the CICS SIT) or a BBIJNT00/BMCICS00 entry (from the BBIPARM DD statement in the CICS startup JCL).
The MainView *for CICS* agent code locates a directive to use a particular BBI-SS PAS and determines that the targeted BBI-SS PAS is participating in AutoConnect.

In both cases, the MainView *for CICS* agent code initiates an AutoConnect request through the common communication request queue.

### Initializing AutoConnect on the BBI–SS PAS

AutoConnect is initiated on the BBI-SS PAS by parameter ACONCICS, which is contained in member BBISSP00 in UBBPARM.

**ACONCICS = {YES | ONLY}**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>The PAS will participate in AutoConnect. JNT elements will be built for target entries in the BBINJT00 parameter member, target entries defined with TGTDEFS, or any target CICS region that it has been assigned to manage by AutoConnect. In this mode, the standard static (BBINJT00) or dynamic (DYNTGT) JNT entries take precedence.</td>
</tr>
<tr>
<td>ONLY</td>
<td>The PAS will participate in AutoConnect. It will ignore any target entries from the BBINJT00 parameter member and any TGTDEFS and only create JNT elements for any target CICS region that it has been assigned to manage by AutoConnect.</td>
</tr>
</tbody>
</table>

If you specify ACONCICS=ONLY or ACONCICS=YES, a CASID=xxxx value must be specified in BBISSPxx, and a PAS connection to that CAS is required. Also any other PAS in the PAS-PLEX must have DYNTGT=YES or the same ACONCICS= value specified and a CASID=xxx specified in BBISSPxx and a CAS connection to have access to the auto connected targets that are connected to the AutoConnect PAS.

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

### Terminated AutoConnect targets

When CICS targets that were connected by ACONCICS=YES are terminated, the service point is detached unless there is a static target definition for the CICS.

When CICS targets connected by ACONCICS=ONLY are terminated, the service point is detached.
As a result of a detached service point, all active timers, including monitors, traces, and so on, are deleted.

**AutoConnect affinity factors**

When AutoConnect receives a request from a CICS region to establish communications with a BBI-SS PAS, it uses various factors to determine the affinity between the CICS region and the available BBI-SS PASs.

The factors include:

- presence of a static or dynamic JNT element on the BBI-SS PAS for the CICS region
- presence of a CMRDTL00 definition on the BBI-SS PAS for the CICS region
- a previous connection between the CICS region and the BBI-SS PAS
- presence of the name of a prospective BBI-SS PAS in the communication request
- the current count of active CICS regions being managed by a prospective BBI-SS PAS

Each of these factors is given a weight and the weights are accumulated for each prospective BBI-SS PAS. The BBI-SS PAS with the greatest weight is assigned to manage the CICS region. You can use some of these factors to ensure that CICS regions connect with the appropriate BBI-SS PAS.

Table 13 on page 70 describes the affinity factors and their associated weights.

**Table 13: Affinity factors**

<table>
<thead>
<tr>
<th>Factor</th>
<th>Weight (decimal)</th>
<th>Weight (hexadecimal)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>The initial weight factor.</td>
<td>1,024</td>
<td>400</td>
<td>This weight is assigned to every request as a starting point.</td>
</tr>
<tr>
<td>The BBI-SS PAS has a communications link to a CICS region.</td>
<td>-10</td>
<td>FFFFFFF6</td>
<td>Negative weight is used to balance the load when more than one BBI-SS PAS is participating in AutoConnect. This factor is added for each CICS region already connected to the BBI-SS PAS.</td>
</tr>
<tr>
<td>The target CICS region was previously attached to the BBI-SS PAS during the current execution of the BBI-SS PAS.</td>
<td>8,192</td>
<td>2000</td>
<td>This factor may be the result of recycling the CICS region, or terminating and restarting the MainView for CICS agent on the CICS region.</td>
</tr>
<tr>
<td>Factor</td>
<td>Weight (decimal)</td>
<td>Weight (hexadecimal)</td>
<td>Comment</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------</td>
<td>------------------</td>
<td>----------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>The BBI-SS PAS has a static JNT element for the target CICS region</td>
<td>16,384</td>
<td>4000</td>
<td>This factor can be provided by using a BBIJNT00 element for the target CICS region on the BBI-SS PAS. Not used if the site utilizes the ACONCICS=ONLY parameter.</td>
</tr>
<tr>
<td>(from a target entry in the BBIJNT00 parameter member)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The BBI-SS PAS has a dynamic JNT element for the target CICS region</td>
<td>8,191</td>
<td>1FF</td>
<td>This factor can be provided by using a TGTDEF for the target CICS region. Not used if the site utilizes the ACONCICS=ONLY parameter.</td>
</tr>
<tr>
<td>(from a TGTDEF)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The target CICS region was provided the name of a BBI-SS PAS from an</td>
<td>10,240</td>
<td>2800</td>
<td>This factor can be provided by using an INITPARM parameter in the CICS SIT.</td>
</tr>
<tr>
<td>INITPARM parameter.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The target CICS region was provided the name of a BBI-SS PAS internally.</td>
<td>15,360</td>
<td>3C00</td>
<td></td>
</tr>
<tr>
<td>The target CICS region was provided the name of a candidate BBI-SS</td>
<td>1,073,741,823</td>
<td>3FFFFFFFF</td>
<td>The FST2 transaction can be used to force the connection of a CICS region to a specified BBI-SS PAS by issuing FST2 INIT xxxx where xxxx is the name of the candidate BBI-SS PAS.</td>
</tr>
<tr>
<td>PAS through the FST2 transaction.</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The BBI-SS PAS already has an active service point for the target</td>
<td>805,306,367</td>
<td>2FFFFFFFF</td>
<td>This normally means the CICS region has previously connected to the BBI-SS PAS.</td>
</tr>
<tr>
<td>CICS region.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The BBI-SS PAS has a static or dynamic JNT element for the target</td>
<td>536,870,911</td>
<td>1FFFFFFFF</td>
<td>This large weight results from the combination of both the BBI-SS PAS and the CICS region basically agreeing that they should establish communications.</td>
</tr>
<tr>
<td>CICS region and the CICS region has provided the name of the BBI-SS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAS as its preferred candidate.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The BBI-SS PAS has a CMRDETL definition for the target CICS region in</td>
<td>3,072</td>
<td>C00</td>
<td></td>
</tr>
<tr>
<td>it CMRDTL00 parameter member and the file can be allocated to the</td>
<td></td>
<td></td>
<td>This weight is an indication of the ability of the BBI-SS PAS to record MainView for CICS transaction performance data as history records.</td>
</tr>
<tr>
<td>BBI-SS PAS job/stated task.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The target CICS provides a directive in the member BMCICSn or BBIJNTnn.</td>
<td>14,336</td>
<td>3800</td>
<td>This factor can be provided in the CICS JCL by specifying a BBIPARM DD statement that refers to the data set that contains the member BMCICSn or BBIJNTnn.</td>
</tr>
</tbody>
</table>
Affinity factor examples

The following examples demonstrate the affect that the affinity factors have on deciding which BBI-SS PAS should manage a particular CICS region.

Additionally, they illustrate how to use the parameters to guide AutoConnect in determining what CICS regions should be managed by which BBI-SS PAS. In all of these examples, the following configuration is used:

- Single LPAR running a single MainView CAS (CASA)
- Three BBI-SS PASs (PASA, PASB, PASC)
  - PASA and PASB are considered production BBI-SS PASs and are used to monitor the production CICS regions. They share a common UBBPARM data set.
  - PASC is considered a test BBI-SS PAS and is used to monitor the test CICS regions. Its UBBPARM library is separate from that used by the other two BBI-SS PASs.
- 75 target CICS regions (CICS01 - CICS75)
  - CICS01-CICS60 are production CICS regions
  - CICS61-CICS75 are test regions
- PASA currently monitoring 27 CICS regions
- PASB currently monitoring 24 CICS regions
- PASC currently monitoring 5 CICS regions

Affinity factor example one

This topic presents an affinity factor example.

Assume that these parameter settings are in effect:
All BBI-SS PASs are running with ACONCICS=YES set in their respective BBISSP00 parameter members.

All BBI-SS PASs are running with DYNTGT=NO set in their respective BBISSP00 parameter members (indicating that TGTDEFs will have no affect on the systems).

The UBBPARM data set for PASA and PASB contains:

— a BBIJNT00 member with no CICS entries
— a CMRDTL00 member with an entry for each of the production CICS regions (CICS01-CICS60) only

The UBBPARM data set for PASC contains:

— a BBIJNT00 member with no CICS entries
— a CMRDTL00 member with an entry for each of the test CICS regions (CICS61-CICS72) only

None of the CICS regions use either the INITPARM SIT parameter or a BBIJNT00/BMCIC00 to indicate to the region which PAS it is to use.

### Case 1.A—a new production CICS region (CICS52) becomes active

Because the CICS region does not have a directive to use a specific BBI-SS PAS, it places a request into the common communications request queue. This element is picked up by one of the three BBI-SS PASs, which determines that it should connect to PASB based upon the following accumulated affinity weights:

<table>
<thead>
<tr>
<th>PAS</th>
<th>Weights</th>
<th>Description</th>
</tr>
</thead>
</table>
| PASA | 1024 -270 3072 3826 | Initial weight  
Weight for each connected CICS region (negative)  
Weight for BBI-SS PAS having CMRDTL00 entry for the region  
Total weight for BBI-SS PAS |
| PASB | 1024 -240 3072 3856 | Initial weight  
Weight for each connected CICS region (negative)  
Weight for BBI-SS PAS having CMRDTL00 entry for the region  
Total weight for BBI-SS PAS |
| PASC | 1024 -50 974 | Initial weight  
Weight for each connected CICS region (negative)  
Total weight for BBI-SS PAS |
**Case 1.B—a new test CICS region (CICS66) becomes active**

Because the CICS region does not have a directive to use a specific BBI-SS PAS, it places a request into the common communications request queue. This element is picked up by one of the three BBI-SS PASs, which determines that it should connect to PASC based upon the following accumulated affinity weights:

<table>
<thead>
<tr>
<th>PAS</th>
<th>Weights</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PASA</td>
<td>1024</td>
<td>Initial weight</td>
</tr>
<tr>
<td></td>
<td>-270</td>
<td>Weight for each connected CICS region (negative)</td>
</tr>
<tr>
<td></td>
<td>754</td>
<td>Total weight for BBI-SS PAS</td>
</tr>
<tr>
<td>PASB</td>
<td>1024</td>
<td>Initial weight</td>
</tr>
<tr>
<td></td>
<td>-240</td>
<td>Weight for each connected CICS region (negative)</td>
</tr>
<tr>
<td></td>
<td>784</td>
<td>Total weight for PAS</td>
</tr>
<tr>
<td>PASC</td>
<td>1024</td>
<td>Initial weight</td>
</tr>
<tr>
<td></td>
<td>-50</td>
<td>Weight for each connected CICS region (negative)</td>
</tr>
<tr>
<td></td>
<td>3072</td>
<td>Weight for BBI-SS PAS having CMRDTL00 entry for the region</td>
</tr>
<tr>
<td></td>
<td>4046</td>
<td>Total weight for BBI-SS PAS</td>
</tr>
</tbody>
</table>

**Affinity factor example two**

This topic presents an affinity factor example.

Assume that these parameter settings are in effect:

- PASA and PASB are running with ACONCICS=YES set in their respective BBISSP00 parameter members.
- PASC is running without an ACONCICS= parameter (the same as ACONCICS=NO).
- All BBI-SS PASs are running with DYNTGT=YES set in their respective BBISSP00 parameter members (indicating that TGTDIFs will be honored on the systems)
- CASA contains TGTDIFs relating PASA to CICS01–CICS30, PASB to CICS31–CICS60 and PASC to CICS61–CICS75
- The UBBPARM data set for PASA and PASB contains:
  - A BBIJNT00 member with no CICS entries
  - A CMRDTL00 member with an entry for each of the production CICS regions (CICS01–CICS60) only
The UBBPARM data set for PASC contains:

— a BBIJNT00 member with no CICS entries

— a CMRDTL00 member with an entry for each of the test CICS regions (CICS61–CICS72) only

CICS01–CICS60 do not use either the INITPARM SIT parameter or a BBIJNT00/ BMCICS00 to indicate to the region which PAS it is to use.

CICS61–CICS75 use an INITPARM SIT parameter to indicate to the regions that they are to connect with PASC.

Case 2.A—a new production CICS region (CICS52) becomes active

Because the CICS region does not have a directive to use a specific BBI-SS PAS, it places a request into the common communications request queue. This element is picked up by either PASA or PASB (PASC is not participating in AutoConnect), which determines that it should connect to PASB based upon the following accumulated affinity weights:

<table>
<thead>
<tr>
<th>PAS</th>
<th>Weights</th>
<th>Description</th>
</tr>
</thead>
</table>
| PASA | 1024 -240 3072 3826 | Initial weight
Weight for each connected CICS region (negative)
Weight for BBI-SS PAS having CMRDTL00 entry for the region
Total weight for BBI-SS PAS |
| PASB | 1024 -240 3072 8191 12047 | Initial weight
Weight for each connected CICS region (negative)
Weight for BBI-SS PAS having CMRDTL00 entry for the region
Weight for a TGTDEF relating CICS52 to PASB
Total weight for BBI-SS PAS |
| PASC | NA | Not participating in AutoConnect |

Case 2.B—a new production CICS region (CICS53) becomes active

The user wants the new CICS to be connected to PASA, so a directive of PASA is specified by using the INITPARM in CICS. CICS53 will connect to PASA based upon the following accumulated affinity weights:
### PASA

<table>
<thead>
<tr>
<th>Weights</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1024</td>
<td>Initial weight</td>
</tr>
<tr>
<td>-240</td>
<td>Weight for each connected CICS region (negative)</td>
</tr>
<tr>
<td>10240</td>
<td>Weight for specifying PASA in INITPARM</td>
</tr>
<tr>
<td>3072</td>
<td>Weight for BBI-SS PAS having CMRDTL00 entry for the region</td>
</tr>
<tr>
<td>14066</td>
<td>Total weight for BBI-SS PAS</td>
</tr>
</tbody>
</table>

### PASB

<table>
<thead>
<tr>
<th>Weights</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1024</td>
<td>Initial weight</td>
</tr>
<tr>
<td>-240</td>
<td>Weight for each connected CICS region (negative)</td>
</tr>
<tr>
<td>3072</td>
<td>Weight for BBI-SS PAS having CMRDTL00 entry for the region</td>
</tr>
<tr>
<td>8191</td>
<td>Weight for a TGTDEF relating CICS52 to PASB</td>
</tr>
<tr>
<td>12047</td>
<td>Total weight for BBI-SS PAS</td>
</tr>
</tbody>
</table>

### PASC

<table>
<thead>
<tr>
<th>Weights</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NA</td>
<td>Not participating in AutoConnect</td>
</tr>
</tbody>
</table>

---

**Case 2.C—a new test CICS region (CICS66) becomes active**

Because the CICS region has a directive to use a specific BBI-SS PAS, it examines the subsystem blocks for PASC and determines that it is not participating in AutoConnect. It then makes a direct connection to PASC.

<table>
<thead>
<tr>
<th>PAS</th>
<th>Weights</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PASA</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>PASB</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>PASC</td>
<td>NA</td>
<td>Not participating in AutoConnect</td>
</tr>
</tbody>
</table>
Using CMRDETL data sets

MainView for CICS enables you to collect task performance data for online viewing and batch reporting.

For additional information, see the MainView for CICS PERFORMANCE REPORTER Data Reference and MainView for CICS PERFORMANCE REPORTER User Guide.

If you do not want to collect transaction performance history data, specify CMRDETL=NO in the CMRSOPT; see “Setting monitoring conditions (CMRSOPT)” on page 125. In addition to providing historical records of task performance, this data is used to determine overall transaction rates and response times for individual CICS regions.

CICS task performance data (referred to as T6E records) are collected on the CICS region (by what is referred to as the extractor) and are shipped to the BBI-SS PAS for preservation.

The extractor is made up of a Task Related User Exit (TRUE) and several CICS Global User Exits (GLUE). The extractor gathers performance and resource utilization information for each task that executes on the CICS region. The extractor delivers the data to the BBI-SS PAS at end of task (and other points) where it is compressed and written to the CMRDETL data sets as transaction history records (T6E). T6E records are delivered to the BBI-SS PAS at the following points:

- end of task
- syncpoint (if MNSYNC=YES)
- end of monitor interval (if MNFREQ=mmss)
- every time a DBCTL call is made (if PERFORM=DELIVER is specified for DBCTL in the MCT)
- end of conversation (if MNCONV=YES)

In order to obtain CICS-provided monitoring, performance, and exception data, MainView for CICS ensures that the MN=, MNPER=, and MNEXC= parameters are turned on while it is active. When the extractor is terminated these parameters are
reset to their original values. Additionally, if these parameters were not on when the extractor was activated, MainView for CICS will ensure that CICS does not write the associated CICS SMF records that it normally would.

Individual transaction IDs can be excluded from data collection by the extractor by use of the CMRSOPT (see “Setting monitoring conditions (CMRSOPT)” on page 125). In addition, the extractor does not collect performance data for the following tasks:

- system tasks with task number of 99xxx
- task running transactions CSNC, CSNE, and CSSY (unless you specify CMRSOPT Include System Task option to collect them from the CMRSOPT views)

**Note**

Resource utilization and performance data for tasks that began prior to the extractor is limited.

Data collection by the extractor can be controlled manually by the QON and QOF commands of the FST2 transaction and the CREGAGT view (see “FST2 transaction” on page 263, and “CREGAGT view commands” on page 267.)

The T6E records are preserved in two ways:

- for long-term use in VSAM files

  This method makes the data available for both the online views and batch reports. There can be either single- or dual-detail (CMRDETL) data sets. In general, BMC recommends that you use dual-detail data sets.

**Note**

The VSAM data set must be a regular single volume VSAM KSDS with no secondary allocations defined and it can be an extended format VSAM data set (with the extended addressability attribute). The data set cannot have VSAM attributes or be under any DASD management that involves:

- data striping
- compression of any kind
- buffer optimization of any kind
- partial release of free space
- retention restriction
- any other management attribute that will manipulate the data or affect the availability of the data set

You can monitor CMRDETL data set status and usage with the CMRDETA family of views. For more information, see Chapter 3, “Administration Views” in the MainView for CICS Online Services Reference Manual.
for short-term use in a data space

Data preserved in a data space is only available for the life of the CICS region (or BBI-SS PAS), and is only available to the online views.

The method of preserving the transaction performance history data is controlled by Task Performance Detail History File Definitions (CMRDETL definitions stored in member CMRDTL00 in UBBPARM). For information about creating these definitions, see “Creating task performance detail history file definitions” on page 80.

Single-detail data sets can be specified in the BBI-SS JCL with the CICS name as the DD name, or specified with CMRDETL definitions. Dual and data space CMRDETL data sets must be specified with CMRDETL definitions.

**Note**

BMC does not recommend specifying detail data sets in the BBI-SS JCL.

---

**Using a history data space**

MainView *for CICS* enables you to place MainView *for CICS* task performance detail history data (T6E records) in a private data space rather than in a data set.

This feature enables you to have temporary access to data from the CHIST views without having to allocate disk storage and this is intended for test CICS regions or other configurations where the data need not be permanently saved.

The data space is defined in accordance with a maximum size parameter and a time limit parameter. The maximum size parameter determines how much space will be allocated to contain the data. The time limit parameter determines how long the data will be maintained in the data space.

When data arrives in the data space and the data space is full or the time limit has expired, sufficient amounts of the expired data are rolled out of the data space to permit the addition of the new data.

Generally, the T6E records for a specific task are kept in the data space until one of the following conditions is met:

- Time limit for retaining data has expired.
- Data space is full.
- Target CICS region terminates.
Restrictions

If the data space becomes full before the time limit expires, the data space will not be expanded.

Old data will be rolled out of the data space in order to make room for the new.

If T6E records are being placed into a data space, they are not recorded to a data set for future use and are not accessible for any batch reporting facility including the MainView for CICS PERFORMANCE REPORTER.

Creating task performance detail history file definitions

Task performance detail history file definitions are created and updated with the CMRDETL view, which saves the definitions in CMRDTL00 in UBBPARM.

This tabular view runs with the context set to the subsystem name of the BBI-SS PAS where the definitions are stored. It displays the definitions that are available on a BBI-SS PAS when a target CICS region becomes active.

Figure 3: CMRDETL view

Using the CMRDETL view

Use the following procedure to use the CMRDETL view.

1. Use the EDIT command (type EDIT on the command line) to enter Edit mode.

   In Edit mode you can:

   ■ create an entirely new definition, or use an existing definition as the template for a new one
delete a definition
To delete a definition, type **DEL** in the **CMD** field next to the definition you want to delete; then press **Enter**.

modify a definition
To modify a definition, type **CHA** (for CHAnge) in the **CMD** field next to the definition you want to use; then press **Enter**.

verify how the definition will work for a given target
To verify a definition, type **VER** in the **CMD** field next to the definition you want to verify; then press **Enter**. The MainView for CICS History File Definition Verification dialog is displayed. See “Verifying definitions” on page 83.

To create a definition, perform one of the following actions:

- To create an entirely new definition, type **ADD** on the command line; then press **Enter**.

- To build a definition from an existing one, type **ADD** in the **CMD** field next to the definition you want to use; then press **Enter**.

The History File Specification dialog is displayed. If you use an existing definition, some of the fields will contain data; otherwise, the fields are blank.

**Figure 4: History File Specification dialog**

```
COMMAND ===> SCROLL ===> PAGE

Targets
..(1) BCVD640
..(2) 
..(3) 
..(4) 

Target Type

Dataset I
DSName BCVM.BCVCD640.CMRDETL1
DDName 

Dataset II
DSName BCVM.BCVCD640.CMRDETL2
DDName 

Dataset Options
Continue YES
Stop NO
Archive JCL Create

Dataspace Options
DataSpace DSP Size DSP Minutes
```

For information about the fields on this dialog, see “History file specification dialog fields” on page 84. After working with this dialog, you can:
- save your work and return to the CMRDETL view by using the END command

- if you are working with an existing definition, you can revert to its initial values by using the RESet command

- return to the CMRDETL view without saving your changes by using the CANcel command

- verify that the definition will work as you intend by using the VERify command; see “Verifying definitions” on page 83.

4 If you want to automatically create history data sets as they are needed, type NEXT on the command line.

The History File Allocation dialog is displayed.

**Figure 5: History File Allocation dialog**

<table>
<thead>
<tr>
<th>Dataset I Allocation Data</th>
<th>Dataset II Allocation Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSName: BCVM.BCVCD410.CMRDETL1</td>
<td>DSName: BCVM.BCVCD410.CMRDETL2</td>
</tr>
<tr>
<td><strong>Primary Records</strong></td>
<td><strong>Primary Records</strong></td>
</tr>
<tr>
<td>VSAM Volume</td>
<td>VSAM Volume</td>
</tr>
<tr>
<td><strong>SMS Stge Class</strong></td>
<td><strong>SMS Stge Class</strong></td>
</tr>
<tr>
<td><strong>SMS Data Class</strong></td>
<td><strong>SMS Data Class</strong></td>
</tr>
<tr>
<td><strong>SMS Mgmt Class</strong></td>
<td><strong>SMS Mgmt Class</strong></td>
</tr>
</tbody>
</table>

5 For information about the fields on this dialog, see “History file allocation dialog fields” on page 87. After working with this dialog, you can:

- save your work and return to the CMRDETL view by using the END command

- if you are working with an existing definition, you can revert to its initial values by using the RESet command

- return to the CMRDETL view without saving your changes by using the CANcel command

- verify that the definition will work as you intend by using the VERify command; see “Verifying definitions” on page 83.

- return to the History File Specification dialog without saving your changes by using the PREvious command
Verifying definitions

The History File Definition Verification dialog enables you to determine if your transaction history detail file definitions work as you intend. It is displayed when you use the VERify command on the CMRDETL view, or on the History File Specification History File Allocations dialogs.

Figure 6: History File Definition Verification dialog

To verify a definition

1. Provide verification information:
   - In the Target Name field, type the name of the CICS target that you want to verify.
     The eight-character name determines which definition to use, or if a selected definition is valid. During verification processing, it is used in place of &CMRTARG variables. See “System variables” on page 88.
   - In the Job Name field, type the name of the job name.
     The eight-character name is used during verification in place of any &CMRJOBN variables. See .
   - In the APPLID field, type the application ID.
     The eight-character name is used during verification in place of any &CMRAPPL variables. See “System variables” on page 88.

MainView for CICS provides several variables that can be used in data set names that represent things such as the release of MainView for CICS, the SSID of the BBI-SS PAS, the MainView for CICS target name (the target CICS job name), and the target CICS APPLID. Additionally, system symbolics that are provided by z/OS and the local system can also be utilized. See “System variables” on page 88.

2. Use the End command to verify the definition by using the supplied information.
If the definition can be verified, the Detail File Definition dialog is displayed. Otherwise, an error message is displayed.

**Figure 7: Detail File Definition dialog**

```plaintext
*---- This target will use VSAM datasets ----*
Target Name: BCVCD410

*---- Options ----*
Continue: YES
Stop: NO
Archive JCL: CMRDJCL

*---- Dataset 1 ----*
DD Name: 
DS Name: BCVM.BCVCD410.CMRDETL1

*---- Dataset 2 ----*
DD Name: 
DS Name: BCVM.BCVCD410.CMRDETL2
```

3. When you are finished, use the **End** command to return to the view where you used the **Verify** command.

**History file specification dialog fields**

This topic provides a table that describes the history file specification dialog fields shown in the following figure:

**Figure 8: History File Specification dialog**

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>SCROLL</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Targets</td>
<td></td>
<td></td>
</tr>
<tr>
<td>..(1)..</td>
<td>BCVCD640</td>
<td></td>
</tr>
<tr>
<td>..(2)..</td>
<td></td>
<td></td>
</tr>
<tr>
<td>..(3)..</td>
<td></td>
<td></td>
</tr>
<tr>
<td>..(4)..</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target Type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dataset I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSName</td>
<td>BCVM.BCVCD640.CMRDETL1</td>
<td></td>
</tr>
<tr>
<td>DDName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dataset II</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSName</td>
<td>BCVM.BCVCD640.CMRDETL2</td>
<td></td>
</tr>
<tr>
<td>DDName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dataset Options</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continue</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Stop</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>Archive JCL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dataspace Options</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dataspace</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSP Size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSP Minutes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 14: History file specification dialog fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Targets</strong></td>
<td>The four 62-byte areas contain target specifications for the definition. The four areas are considered to be contiguous and all spaces are collapsed. Target names are separated by commas, and can contain wildcards. Valid wildcard characters are as follows:&lt;br&gt;- + represents a single character.&lt;br&gt;- * represents any number of characters (including none).&lt;br&gt;When using single-character replacement, the target name may contain any character at the position of the wildcard. Non-wildcard characters in positions prior to or after the + wildcard must match exactly. The target must contain a character at the wildcard location. Multiple + wildcards can be used, and be contiguously specified.&lt;br&gt;<strong>Example:</strong> Examples are AB+CD, AB++C+D, and ABCD+. Multiple character replacement, designated by the asterisk, <em>, means that any number of characters (or none) can exist starting at the position of the wildcard. Characters preceding the * wildcard must match exactly. All characters are accepted until the next non-wildcard character is matched or the end of the field is reached. Multiple * wildcards can be used in the target name.&lt;br&gt;<strong>Example:</strong> Examples are AB</em>, AB<em>CD and AB</em>CD*. The two wildcards can be combined in a single template, such as AB++D*, AB<em>D+E and A</em>B+C*D.</td>
</tr>
<tr>
<td><strong>Target Type</strong></td>
<td>This field indicates what type of system will use this definition. Valid values are:&lt;br&gt;- CICS indicates that this definition will be used for CICS Transaction Server tasks.&lt;br&gt;- CTG indicates that this definition will be used for CICS Transaction Gateway tasks.&lt;br&gt;For more information, see the DSPSIZE parameter description in the “Set the CTG monitoring parameters in member CMRBEXmn” on page 173 topic.</td>
</tr>
<tr>
<td>Dataset I DSName field</td>
<td>This field identifies the data set name to be used when allocating the primary history file. It is important to ensure that the data set names are unique for each target CICS region. The names can be completely qualified or contain system variables. See “System variables” on page 88. The data set is allocated with a disposition of SHR. The data set must not be protected by VSAM password protection. IBM RACF or equivalent security can be used as long as the BBI-SS PAS has write access to the data set.</td>
</tr>
<tr>
<td>Dataset I DDNAME field</td>
<td>This field identifies the DDNAME of a DD statement specified in the BBI-SS PAS startup JCL to be used for the primary history file. If both DSName and DDNAME are used, this name is used for dynamic allocation.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Dataset II DSName</td>
<td>This field identifies the data set name to be used when allocating the secondary history file. It is important to ensure that the data set names are unique for each target CICS region. The names can be completely qualified or contain system variables. See “System variables” on page 88. The data set is allocated with a disposition of SHR. The data set must not be protected by VSAM password protection. RACF or equivalent security can be used as long as the BBI-SS PAS has write access to the data set.</td>
</tr>
<tr>
<td>Dataset II DDNAME</td>
<td>This field identifies the DDNAME of a DD statement specified in the BBI-SS PAS startup JCL to be used for the secondary history file. If both DSName and DDNAME are used, this name is used for dynamic allocation.</td>
</tr>
<tr>
<td>Dataset Options:</td>
<td></td>
</tr>
<tr>
<td>Continue</td>
<td>This field specifies whether data collection should resume on the history file that was active when recording stopped. Valid values are as follows:</td>
</tr>
<tr>
<td></td>
<td>■ YES specifies that this value specifies that recording begins in the history file that was active when data collection stopped. An archive request is submitted when the data set becomes full. This option is the default.</td>
</tr>
<tr>
<td></td>
<td>■ NO specifies that this value specifies that recording begins in the history file that was inactive when data collection stopped. An archive request is submitted when the data set becomes full, when data collection stops, or when the BBI_SS PAS terminates normally.</td>
</tr>
<tr>
<td></td>
<td>“Controlling data set switching with CONTINUE=YES” on page 91 gives an example of how the Continue option controls history file recording.</td>
</tr>
<tr>
<td>Dataset Options:</td>
<td></td>
</tr>
<tr>
<td>Stop</td>
<td>This field specifies whether data collection should stop when a data set switch occurs and the target data set contains data that has not been archived nor is being archived (the archive in progress flag is not on). Valid values are as follows:</td>
</tr>
<tr>
<td></td>
<td>■ YES – This value terminates recording to the history file. Data collection must be stopped and restarted to reactivate data recording. This option is the default.</td>
</tr>
<tr>
<td></td>
<td>■ NO – This value specifies that recording is to continue. If unarchived records are detected on the switched data set, an FT224 message is issued to warn that existing records are being overwritten.</td>
</tr>
<tr>
<td>Dataset Options:</td>
<td></td>
</tr>
<tr>
<td>Archive JCL</td>
<td>This field specifies the name of the started task that requests an archive of the history file. See “CMRDETL data set switch exit” on page 94 for the format of the request. Valid values are as follows:</td>
</tr>
<tr>
<td></td>
<td>■ CMRDJCL – This value is the default name of the archive PROC. If no value is specified, when a switch occurs, this name will be used to request archiving.</td>
</tr>
<tr>
<td></td>
<td>■ archjcl – This value can be any valid started task name.</td>
</tr>
<tr>
<td></td>
<td>■ NONE – This value specifies that an archive request should not be submitted when a data set becomes full. NONE can be used when an automation product submits an archive request in response to messages, or when data archiving is not required.</td>
</tr>
</tbody>
</table>

86  MainView for CICS Customization Guide
### History file allocation dialog fields

The following table describes the history file allocation dialog fields displayed in the following figure:

**Figure 9: History File Allocation dialog**

```
----- Dataset I Allocation Data -----  
DSName: BCVM.BCVCD410.CMRDETL1

Primary Records
VSAM Volume
SMS Stge Class
SMS Data Class
SMS Mgmt Class

----- Dataset II Allocation Data -----  
DSName: BCVM.BCVCD410.CMRDETL2

Primary Records
VSAM Volume
SMS Stge Class
SMS Data Class
SMS Mgmt Class
```
Table 15: History file allocation dialog fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Dataset I / Dataset II: Primary Records | Contains the count of records used to create the history file data set  
If this value is not specified, a default value of 15,000 will be used. See “Calculating primary record specification” on page 90. |
| Dataset I / Dataset II: VSAM Volume | Contains the names of the VSAM volume that is used to create the history file data set |
| Dataset I / Dataset II: SMS Stge Class | Contains the SMS storage class used in the history file data set |
| Dataset I / Dataset II: SMS Data Class | Contains the SMS data class used in the history file data set |
| Dataset I / Dataset II: SMS Mgmt Class | Contains the SMS management class used in the history file data set |

Note

The SMS storage class must not allow multiple volumes. The SMS data class can the extended-format and have extended addressability attributes. The data class cannot have attributes that include:
- specifications to compress
- buffer optimization of any kind
- striping
- system-managed buffering
- any other attribute that will manipulate the data

The SMS management class must not allow partial release, retention restriction, or other management attributes that will affect the availability of the data set.

System variables

System variables can be used in the data set names of the history files.

In addition to the system variables implemented by z/OS and those you have implemented, several variables are provided to assist in building data set names:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
</table>
| &CMRREL  | four-character release ID of MainView for CICS in the form vrmm  
MainView for CICS 6.7.00 would be returned as 6700, for example. |
| &CMRPAS  | four-character subsystem identifier of the MainView for CICS BBI-SS PAS |
Variable | Description
--- | ---
&CMRSMF | four-character SMF ID for the z/OS LPAR
&CMRTARG | eight-character MainView for CICS target name for the CICS region (normally the jobname)
&CMRJOBN | eight-character JOBNAME of the target CICS region
&CMRAPPL | eight-character APPLID of the target CICS region

These variables are resolved at runtime.

**Automatic data set creation**

MainView for CICS enables you to have task performance detail history datasets created automatically.

The feature requires that the Data Set Create Option is set to YES.

This feature performs a VSAM create action for the data sets based on the settings for data set name, record count, and the volume or SMS options specified in the History File Specification and History File Allocation dialogs. See “Creating task performance detail history file definitions” on page 80.

**Manual data set creation**

To manually create history data sets, perform the following steps. These steps must be repeated for each managed region.

1. Copy member CMRDDTL from your BBSAMP to UBBSAMP data sets.
   
   Ensure the new member name conforms to your site's naming conventions.

2. Edit the member created in the previous step as instructed in the comments of the member and in the following steps.

3. Replace the line that begins with the following text with a valid job statement for your site:
   
   ```
   //CMRDDTL JOB
   ```

4. Calculate the primary record extent that is required for the CMRDDET data sets by estimating the number of CICS transactions to be stored on a single data set.

   See “Calculating primary record specification” on page 90.
5 Replace all occurrences of CMR.CICS1 with

\[ prefix.cicsName \]

Define \textit{cicsName} as the name of the CICS region. This step creates unique data sets for each CICS region.

6 Replace all occurrences of CICSMGR with the prefix of the BBLINK data set name.

\[ prefix \]

7 Replace all occurrences of ???????? with the ID of the volume where the CMRDETL data sets are located.

8 Replace \texttt{RECORDS(15000 0)} with \texttt{RECORDS(\textit{primary} 0)}, where \textit{primary} is the primary record allocations that were calculated in \textit{Step 4 on page 89}.

The secondary allocation should always be 0.

9 Submit the job.

\textit{Note}

Do not change any other VSAM attributes in the JCL. The VSAM data set must be a regular single volume VSAM KSDS with no secondary allocations defined. The data set can be in extended format with extended addressability but it cannot have VSAM attributes or be under any DASD management that involves:

- data striping
- compression of any kind
- buffer optimization of any kind
- partial release of free space
- retention restriction
- any other management attribute that will manipulate the data or affect the availability of the data set

\section*{Calculating primary record specification}

The size of the CMRDETL data sets is a factor of the length of time that you want to preserve T6E records for online history views.

A simple calculation of the estimated number of transactions per day multiplied by the maximum number of days provides the general size. This calculation corresponds to the formula:

\[ \text{CMRDETL Size} = (\text{transactions per day}) \times (\text{number of days}) \]
If you are using a single CMRDETL data set, it must be large enough to contain data for all the days that you want to preserve, which means that the primary record specification should be equal to the CMRDETL size by using the above formula.

If you are using dual CMRDETL data sets, the CMRDETL size can be distributed between the individual data sets.

Although VSAM allocates space according to the specified number of records, other factors such as the method of loading records and differences in the compressed size of the T6E records will affect the ability of VSAM to completely use all of a given control area. *It is a good idea to increase the primary record specification by 25%.*

For optimum VSAM performance, the primary allocation for CMRDETL should never be less than one cylinder.

The secondary extent should always be zero.

**Detail data-collection scenarios**

This section provides examples of how to use the CONTINUE and STOP parameters to control switching between dual CMRDETL data sets.

**Controlling data set switching with CONTINUE=YES**

This topic provides an example of controlling data set switching with CONTINUE=YES.
Figure 10 on page 92 shows that DSN1 and DSN2 contain data collected over a seven-day period.

When data collection starts for a CICS region, MainView for CICS determines which data set was used last. DSN2 was the last data set used; thus, when CONTINUE=YES is specified in BBPARM member CMRDTL00, data collected on the eighth day is stored on DSN2. Recording continues on DSN2 until it becomes full. The CMRUSR01 user exit is invoked and a start command to archive DSN2 is issued. See “CMRDETL data set switch exit” on page 94, for the format of the archive request. An attempt is made to switch data collection to DSN1.

When DSN1 contains unarchived data, the value of the STOP parameter determines the next data collection step:

- If STOP=YES, data collection stops if unarchived data is present. It can be restarted after the data set is archived by stopping and restarting data collection.

- If STOP=NO, data is overwritten if the data set is not being archived (the archive in progress flag is not on).

If data has been archived, all existing DSN1 data (days 1 through 4) is deleted and detail collection starts from the beginning of the data set.

When the CICS region terminates or data collection is deactivated, the data set is closed and an archive request is not issued.
Controlling data set switching with \texttt{CONTINUE=NO}

This topic provides an example of controlling data set switching with \texttt{CONTINUE=NO}.

\textbf{Figure 11 on page 93} shows that DSN1 and DSN2 store data collected over a seven-hour period.

DSN2 was the last data set used. Specifying \texttt{CONTINUE=NO} requests that data collection begin on DSN1; it was not being used when data collection stopped.

\textbf{Figure 11: Example of dual CMRDETL data sets with hourly data}

![Diagram showing two data sets](image)

When data collection for the CICS region starts, MainView for CICS determines which of the data sets was used last and starts recording on the other data set (because it had more time to be archived). If DSN1 contains data that has not been archived, the \texttt{STOP} parameter determines the next data collection step:

- If \texttt{STOP=YES}, data collection stops. It can be restarted after the data set is archived by stopping and restarting data collection.
- If \texttt{STOP=NO}, data is overwritten if the data set is not being archived (the archive in progress flag is not on).

If data has been archived, all existing DSN1 data (hours 1 through 4) is deleted.

Collection continues on DSN1 until it becomes full. The CMRUSR01 user exit is invoked and a start command to archive DSN1 is issued. See “CMRDETL data set switch exit” on page 94, for the format of the archive request.

An attempt is made to switch data collection to DSN2. If it has any unarchived data, the \texttt{STOP} options determine the action to be taken:
- If STOP=YES, data collection stops. It can be restarted after the data set is archived by stopping and restarting data collection.

- If STOP=NO, data is overwritten if the data set is not being archived (the archive in progress flag is not on).

If data has been archived, all existing DSN2 data (hours 5 to 7) is deleted and detail data set collection starts at the beginning of the data set.

The data set is closed and an archive request is issued under the following conditions:

- CICS region becomes inactive.

- Data collection stops.

- BBI-SS PAS becomes inactive.

# CMRDETL data set switch exit

The CMRUSR01 user exit is invoked when a CMRDETL data set becomes full.

CMRUSR01 starts a task to archive CMRDETL data unless ARCHJCL=NONE is specified in CMRDTL00. The started task name is specified like this example:

```sql
S CMRDJCL,CICS=cicsName,ARCHDSN='sample.data'
```

The exit is given the following parameters:

- BBI-SS PAS ID

- archive JCL name (ARCHJCL)

- CICS region name

- associated data set name

A sample of CMRUSR01 is provided in the BBSAMP data set. This sample can be edited and used if needed, but it is not recommended. BMC Software does not support any user changes made to CMRUSR01.
Dual data set considerations

Keep the following special considerations in mind when using dual CMRDETL data sets:

- **Data set switching messages**
  Data set switching messages can be seen in the BBI-SS joblog and the BBI-SS journal. The messages are documented under the MESSAGES service (MSG msg#).

- **Online access of CMRDETL data**
  The selection criteria specified in the online History service determines which data set to read.

- **Batch processing of CMRDETL data sets**
  Batch jobs should never be run against the inactive CMRDETL data set. For example, if a Performance Reporting Language (PRL) program uses CMRDETL data, the job must run against either the active data set or an archived copy of the inactive data set. Your batch job can fail if it is actively selecting records from the inactive CMRDETL data set when a switch occurs.

  If necessary, the primary and archived data can be REPROed with IDCAMS into a single file for batch processing.

Single data set considerations

When working with single data sets, consider the following guidelines:

- You must manually start the archive job when the data set is allocated in the BBI-SS PAS proc.

- Specify the CMRDETL data set in CMRDTL00 in UBBPARM will avoid recycling the BBI-SS PAS when the data set name is changed.

- If CMRDETL data archives are not needed, define the data set in CMRDTL00 and specify CONTINUE= YES, STOP= NO, ARCHJCL= NONE, and specify only DSN1. When the data set is full, it will overwrite the data set from the beginning and recording will continue.

- If CMRDETL data archives are needed, define the data set in CMRDTL00 and specify CONTINUE= YES, STOP= YES, specify DSN1 only, and set ARCHJCL to a proc that only runs CMRPURG (a sample can be found in member CMRSJCL in
BBSAMP). When the data set is full, the archive proc will be submitted and recording will stop.
Enter **FST2 QOFF** and then **FST2 QON** to restart recording. When archiving is desired, dual CMRDETL data sets is recommended.

- Specify ARCHJCL= **NONE** if STOP= **NO** in CMRDTL00 for a single CMRDETL data set.

### Archiving a CMRDETL data set

CMRDETL data sets can be archived by using the CMRPURG program or other standard z/OS archive utilities.

The CMRPURG program archives and purges data stored in the MainView for CICS CMRDETL data sets. CMRPURG can also merge CMRDETL records with other records archived to tape.

---

**Note**

CMRPURG cannot request an archive of both CMRDETL data sets at the same time. An archive job must be run against one CMRDETL data set at a time.

---

### Archiving a single CMRDETL data set

This section describes the procedure to archive and purge records from single CMRDETL data set environments.

Refer to “Archiving dual CMRDETL data sets” on page 99 for a description of the procedure to process dual CMRDETL data set environments with CMRPURG.

Before submitting the CMRPURG job, data recording must be stopped automatically when a data set is full, or manually by using the QOF command of the FST2 transaction or the CREGAGT view (see “FST2 transaction” on page 263, and “CREGAGT view commands” on page 267.)

A sample procedure to archive and purge all data by using the CMRPURG program can be found in member CMRSJCL in BBSAMP. The procedure can be used in parameter ARCHJCL in CMRDTL00 for single CMRDETL data sets only. See “Single data set considerations” on page 95 for more information.

Sample JCL to archive, merge and purge detail records with the CMRPURG program can be found in the CMRPURGE member of your BBSAMP data set.
12 on page 97 shows an example of the JCL to run the CMRPURG batch program. Descriptions of the control statements follow the example.

Figure 12: Sample JCL to run CMRPURG for a single CMRDETL data set

```plaintext
//jobname  JOB  user parameters
/*JOBPARM  user parameters
//JOBLIB   DD   DISP=SHR,DSN=CMR.CMRV3.BBLINK
//ARCHIVE  EXEC PGM=CMRPURG
//TAPEIN   DD   DISP=OLD,DSN=PREV.HISTORY.FILE
//TAPEOUT  DD   DISP=(NEW,CATLG),DSN=CMR.MERGED.ARCHIVE,UNIT=TAPE
//CMRWRK1  DD   DSN=CMR.CMRWRK1,SPACE=(CYL,(10,10)),UNIT=SYSDA
// CMRDETL  DD   DSN=CMR.CMRDETL,SPACE=(CYL,(10,10)),UNIT=SYSDA
// REPORT   DD  SYSOUT=*  
//SYSPRINT DD   SYSOUT=* 
//SYSIN    DD   *
DATE=mmddyyyy
HISTORY=YES
```

// TAPEIN identifies the data set to be merged and archived with the purged CMRDETL records.

// TAPEOUT identifies the output tape data set for the purged CMRDETL records.

//CMRWRK1 specifies a data set, CMRWRK1, to hold all records that are not archived during the reorganization phase. The recorder data sets also must be defined with REUSE so that reorganization can occur. Records are written to tape or disk.

//CMRDETL identifies the MainView for CICS CMRDETL data sets to be processed.

DATE specifies the purge date. This statement must start in column 1. All data stamped with this date or a prior date is purged and, if requested, archived to the output tape data set.

DATE=12319999 purges all the data from the recorder data set.

---

**Note**

If 12319999 is specified, the CMRWRK1 data set is not required.

The purge date can also be specified as the number of days preceding the current date. The format is

```
DATE=*-nn
```

Define \( nn \) as a two-digit number of days prior to the current date. Note that the date calculated can represent a day in the previous year. The actual date calculated by the program is displayed by an informational message on the report output.
HISTORY specifies the type of history tape manipulation to be performed. This statement must start in column 1. Valid values are as follows:

- **NO** specifies that no input or output history tapes are to be used and purges all data that has a date earlier than the date specified.

- **OUT** specifies that no input tape is to be used. Only an output tape is to be created containing the data to be purged.

- **YES** specifies that both input and output tapes are to be used. The data on the input tape is to be merged with the data to be purged from the MainView for CICS recorder data sets and written to the output tape.

### Archive and purge

The process is as follows:

1. Read data from the recorder data sets; merge it with another archive data set (HISTORY=YES); write it to the history output tape (HISTORY=YES or HISTORY=OUT).

   If HISTORY=NO is specified, the data is deleted and not archived.

2. Purge data from the recorder data sets.

   **Note**

   Purging is performed here so that abends (such as tape errors) do not leave the recorder data sets in a state of reduced integrity.

The data that is purged is determined by the DATE control statement shown in Figure 12 on page 97. The entire data set can be purged by specifying DATE=12319999, or a portion of the recorder data set can be purged as follows:

- If only part of a data set is to be purged, define the VSAM REUSE parameter when the CMRDETL recorder data set is defined.

- Allocate a temporary data set, as shown by CMRWRK1 in Figure 12 on page 97. CMRPURG uses CMRWRK1 to store records temporarily that will remain on the recorder data set.

- REUSE reorganizes and resets the CMRDETL recorder data sets so that records written to CMRWRK1 from these data sets can be reloaded.

- If NOREUSE is specified, the selected records are deleted from CMRDETL; this process can be a time-consuming update to the CMRDETL data set.
Each phase of the archive-purge process is recorded in the Recorder File Purge Report. The following example shows a typical listing for the CMRDETL recorder data set.

**Figure 13: Sample file purge report**

<table>
<thead>
<tr>
<th>CMRDETL File Purge Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS MANAGER RECORDER FILE PURGE</td>
</tr>
<tr>
<td>FT2901-DETAIL TRANSACTION RECORDER FILE BEING PROCESSED</td>
</tr>
<tr>
<td>FT2931-BEGINNING ARCHIVE OF REQUESTED RECORDS</td>
</tr>
<tr>
<td>FT2941-ARCHIVE OF REQUESTED RECORDS COMPLETED</td>
</tr>
<tr>
<td>FT2951-BEGINNING PURGE OF REQUESTED RECORDS</td>
</tr>
<tr>
<td>FT2961-PURGE OF REQUESTED RECORDS COMPLETED</td>
</tr>
<tr>
<td>FT2971-CMRWRK1 DATASET FOUND - BEGINNING REORGANIZATION</td>
</tr>
<tr>
<td>FT2981-ENTERING SECOND PHASE OF REORGANIZATION</td>
</tr>
<tr>
<td>FT3011-REORGANIZATION OF RECORDER FILE COMPLETED</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>----- RECORDER FILE -----</th>
<th>----- HISTORY FILE ACTIVITY -----</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>PURGED</td>
</tr>
<tr>
<td>000001667</td>
<td>000001666</td>
</tr>
</tbody>
</table>

If an abend occurs while archiving data, check the messages in the report to determine when it occurred. CMRPURG does not write the same record twice if it exists on both the history and the recorder data sets.

**Archive–Purge completion**

This topic provides information about archive–purge completion.

When the archive-purge process is complete, restart data collection with the QOF command of the FST2 transaction, or the CREGAGT view (see “FST2 transaction” on page 263, and “CREGAGT view commands” on page 267.) This action starts recording to the CMRDETL data set again.

**Archiving dual CMRDETL data sets**

The use of CMRPURG to process dual CMRDETL data sets is restricted to archival only.

When ARCHJCL=CMRDJCL (default) or ARCHJCL=proc is specified in CMRDTL00, the archive process is initiated automatically when CMRDETL becomes full, an I/O error occurs, or a switch is manually requested with the FST2 SWITCH command.

BBSAMP member CMRDJCL contains sample JCL to execute CMRPURG when a detail file switch occurs. The steps in CMRDJCL must run when archiving a CMRDETL data set in a dual data set environment.
BBSAMP member CMRDPARM contains the necessary CMRPURG control statements to archive a CMRDETL data set in a dual CMRDETL data set environment.

The following control statements must start in column 1 and be included when you use CMRPURG to archive dual CMRDETL data sets:

- HISTORY=OUT
- DATE=12319999
- LEAVE (specifies whether to leave data in CMRDETL or purge it after archival)
  Valid values are as follows:
  — NO specifies that data is removed from CMRDETL. This value is not recommended. Data will not be available for online access after the switch.
  — YES specifies that data remains in CMRDETL after archival. This value is recommended. It provides HISTORY access to the data after a data set switch occurs.

**Using CMRDJCL**

CMRDJCL contains three steps: ARCHPRE, ARCHIVE and ARCHPOST.

- ARCHPRE runs program CMRARCH1 to turn on the archive-in-progress flag in the CMRDETL seed record. This flag prevents the data set from being used for recording while it is being archived.
- ARCHIVE runs program CMRPURG to archive the data.
- ARCHPOST runs program CMRARCH2 to turn off the archive-in-progress flag in the CMRDETL seed record, which indicates that the data set can be used for recording.

In the event that ARCHIVE fails (a tape-drive error, for example), ARCHPOST will not run and the data set will not be eligible for recording. Even with STOP=NO, the archive-in-progress flag will cause recording to stop if there is no eligible data set.

In a dual data set environment when ARCHJCL is not set to NONE, MainView for CICS will close a data set when it becomes full, submit the archive proc to off-load the data set, and then try to switch to the other data set. If the other data set has the archive-in-progress flag on, the FT226W ARCHIVE IN PROGRESS message will be issued every fifteen seconds for two minutes, after which processing will stop for this data set and the FT224W NOT ARCHIVED message will be issued. At this point, MainView for CICS will attempt to write to the data set that was just archived.
If the archive-in-progress flag is still on for this data set, the application issues the FT226W message until the data set is ready to be used or when the two minutes is reached. After two minutes pass, the application issues the FT224W message for this data set, stops recording, and issues the FT225E DETAIL FILE COLLECTION STOPPED and FT255I DETAIL TRANSACTION RECORDER FILE IS CLOSED messages. For information about resolving this issue, see the online Help for the FT226W message.

FST2 QOFF and FST2 QON are required in order to restart recording after resolving the FT226W related problem.

**Using another archive utility**

You can use any z/OS archival utility, such as IDCAMS REPRO, to archive CMRDETL data sets.

The output data set should have the following DCB characteristics:

DSORG=PS,RECFM=VB,LRECL=32736,BKSIZE=32740

To indicate the start and end of archiving, step ARCHPRE must precede the archive step and ARCHPOST must follow the archive step, as shown in BBSAMP member CMRDJCL.

See “Using CMRDJCL” on page 100 for more information.

**CMRLDCTL reload utility**

You can use the CMRLDCTL batch utility to load a VSAM data set with T6E records from one or more flat files.

The VSAM records can then be viewed with the MainView for CICS online services.

The CMRLDCTL batch utility reads the input records, builds a master record, and writes the records to the output VSAM data set. The master record contains the date and time of the first and last record in the VSAM data set.

The CMRLDCTL utility supports T6E record sizes up to 32K.

The CMRLDCTL utility requirements are:

- The input DD name must be **FLATIN**.

**FLATIN** can refer to a single data set or concatenated data sets.
The input data set must be DSORG=PS.

The input data set can be output from CMRPURGE or IDCAMS REPRO.

The output VSAM data set must be empty and defined with the same attributes as the VSAM CMRDETL data set; do not seed the data set.

The output DD name must be **VSAMOUT**.

**SYSIN statements**

The SYSIN statements for the CMRLDTL utility are shown below.

**IN-TARGET** = *target*

(Required.) Filters input records by target name. Only input records matching the CICS target name will be copied. In this example only input records from target CICSTST1 will be processed:

```
IN-TARGET=CICSTST1
```

**Out Target** = *target*

(Optional.) Specifies an output target name that is different from the original name. For example:

```
OUT-TARGET=CICSTST2
```

OUT-TARGET is useful if you already have a history data set connected to the BBI-SS PAS and you want to view the archived records. For example, if CICS target CICSTST1 is connected to the BBI-SS PAS and is collecting history records, you can load CICSTST1 records into CICSTST2 by specifying OUT-TARGET=CICSTST2. When the dataset is connected to the BBI-SS PAS the target name will be CICSTST2 and it will not conflict with CICSTST1.

**FROM-DATE** = *date*

(Optional.) Loads only the input records that were created on or after the specified date. For example:

```
FROM-DATE=01/01/2009
```

The date can be any of the following formats:

```
MM/DD/YYYY
```
If two digits are specified for the year (YY), 1900 is the assumed century.

You can specify both, either, or neither of the FROM-DATE and TO-DATE statements. Use both statements to load a specific range of records. If FROM-DATE is omitted, the oldest record in FLATIN begins the range. If TO-DATE is omitted, the newest record in FLATIN ends the range.

**TO-DATE=** *date*

(Optional.) Loads only the input records that were created on or before the specified date. For example:

**TO-DATE=01/01/2009**

The date formats are the same as the FROM-DATE date formats.

If you specify TO-DATE without FROM-DATE, all of the input records created on or before TO-DATE are copied.

Figure 14 on page 103 provides an example of an output report produced by CMRLDTL.

**Figure 14: Sample CMRLDTL output**

```
MAINVIEW FOR CICS - CMRDETL RELOAD UTILITY - PROCESSING SUMMARY
RUN DATE: 03/24/2009
RUN TIME: 13:11:21
SYSIN STATEMENTS:
   IN-TARGET=CICSTST1
   OUT-TARGET=CICSTST2
   TO-DATE=01/01/2010
   FROM-DATE=01/01/2001
FILE TYPE PROCESSING: FLATIN AND VSAMOUT
DATE OF FIRST RECORD: 01/01/2008
DATE OF LAST RECORD: 03/17/2009
TOTAL RECORDS READ            : 200,909
TOTAL MATCHING IN-TARGET       : 200,908
MASTER RECORDS SKIPPED         : 1
RECORDS SKIPPED BY DATE        : 0
TOTAL RECORDS SKIPPED          : 1
MASTER RECORDS WRITTEN         : 1
TOTAL RECORDS WRITTEN          : 200,909
```
Records are skipped if they do not meet the SYSIN requirements or if they are master records.

**Loading archived history records example**

The following illustrates the steps you would take to load and view archived history records.

1. Create a data set named SYS4.CICSTST1.ARCHIVE from a CMRDETL data set by using IDCAMS REPRO.

2. Allocate a VSAM data set called SYS4.CICSTST2.CMRDETL1 to hold the loaded records.

3. Run CMRLDTL with the following data sets as input and output:

   ```
   //JOBNAME JOB ()
   //*
   //CMRLDTL EXEC PGM=CMRLDTL
   //STEPLIB DD DISP=SHR,DSN=SYS2.BBLINK
   //SYSPRINT DD SYSOUT=*  
   //FLATIN DD DISP=SHR,DSN=SYS4.CICSTST1.ARCHIVE
   //VSAMOUT DD DISP=SHR,DSN=SYS4.CICSTST2.CMRDETL1
   //REPORT DD SYSOUT=* 
   //SYSIN DD *
   IN-TARGET=CICSTST1
   OUT-TARGET=CICSTST2
   ```

4. Add CICSTST2 to UBBPARM(BBIJNT00) and recycle the PAS, and be sure to use parameter `ATTACH=IMMED`:

   ```
   TARGET=CICSTST2,TYPE=CICS,SUBSYS=XXXX,ATTACH=IMMED
   ```

5. Add CICSTST2 and SYS4.CICSTST2.CMRDETL1 to the PAS using the CMRDETL view.

   See “Using the CMRDETL view” on page 80.

6. Use the CHISTORY view or full screen History to browse the CICSTST2 history records.

   See BBSAMP member CMRLDTL for sample usage.
This chapter describes the procedures that are required to implement functions between a BBI-SS PAS and a CICS region for the first time, or to implement auto maintenance.

If you are upgrading MainView for CICS or MainView AutoOPERATOR for CICS and have already implemented auto maintenance, these procedures do not have to be repeated.

The MainView for CICS auto maintenance facility enables most software upgrades to be applied to the product without restarting target CICS systems. Implementing auto maintenance requires modifications to the CICS CSD, CICS PLT and SIT, and to the BBI-SS JCL. These procedures are described in “Completing implementation” on page 110.

When the modifications are completed, you can use the facility to apply updates and patches to MainView for CICS. See “Using the auto maintenance facility” on page 47 for information about using the facility.

Before applying a product update or patch, familiarize yourself with the topics discussed in “General considerations” on page 106.

Before implementing or using the auto maintenance facility, familiarize yourself with the topics discussed in “Completing implementation” on page 110.

Security considerations are discussed in “Site-specific security” on page 107.

---

**Note**

For a list of the MainView AutoOPERATOR for CICS functions that are available when communications between a BBI-SS PAS and CICS is established, see “Additional MainView AutoOPERATOR for CICS functions” on page 219.
General considerations

This section discusses several issues you should consider before implementing MainView for CICS, MainView AutoOPERATOR for CICS, or both products together.

Temporary storage queue conflicts

MainView for CICS creates a temporary storage queue named BMCMVCTQ, which it uses to save status information across terminations and initialization of MainView for CICS in an auto maintenance environment.

The queue is created when FST2 TERM is entered, and it remains in the system until CICS is recycled. It is deleted and re-created for subsequent TERM/INIT processes.

MainView for CICS issues EXEC CICS STARTs of its own transactions to process action commands that are issued by users. The EXEC CICS START is issued with the REQID parameter requesting CICS to create a temporary storage queue with a prefix of CMRI.

Neither type can use a temporary storage model that activates the RECOVERY or REMOTE attributes. If you issue the CEMT INQUIRE TSMODEL(*) command and see a model definition with the RECOVERY or REMOTE attributes activated and a PREFIX defined that will include the MainView for CICS temporary storage queue names, refer to “CICS temporary storage and enqueue name conflicts” on page 225. This appendix explains how to apply a ZAP that changes the prefix of the MainView for CICS temporary storage queue names.

Enqueue name conflicts

The FST2 transaction issues an enqueue on BMCFST2Q when an FST2 transaction is initiated.

Subsequent FST2 transactions attempt to enqueue on this name before running, and terminate if the enqueue is busy.

The BCRT transaction issues an enqueue on BBK1CRTR when the transaction is initiated. Subsequent BCRT transactions attempt to enqueue on this name before running, and terminate if the enqueue is busy.

The JNL2 transaction issues an enqueue on CMRJRNL when the transaction starts. This enqueue exists for the life of the JNL2 task and is used to prevent additional JNL2 tasks from initializing.
Issue the CEMT INQUIRE ENQMODEL(*) command to see if there is an existing ENQ model that will include the MainView for CICS enqueue names. If a conflict exists, see “CICS temporary storage and enqueue name conflicts” on page 225 for information about applying a zap that changes the enqueue name.

**Processing CICS statistical records**

MainView for CICS writes additional SMF 110 subtype x’0002’ and subtype x’0B02’ statistical records to SMF data sets when it monitors a CICS region. You can turn off recording these records by editing the CMRSOPT.

See “SMF data recording” on page 140 for information about the parameter that is used to control statistical recording.

Some MainView for CICS records are written with a subtype of X’0B02’ compared to the X’0002’ subtype of a CICS SMF 110 record. These records are not compatible with the CICS statistics program and some vendors’ products that use SMF 110 records. In this case, MainView for CICS records must be excluded from the SMF data set before CICS statistics are processed.

An IBM utility program named DFHSTUP can exclude non-SMF records from the data set. DFHSTUP is described in the IBM publication CICS Installation and Operations Guide. BBSAMP member CMRSTATX has JCL to use this utility to remove MainView for CICS records from the SMF data set.

**Site-specific security**

The BBI-SS PAS uses the EXCI function to reconnect to CICS after the BBI-SS PAS is recycled.

The CREGAGT windows-mode views provide commands to control MainView for CICS initialization and termination, and its agent functions—extractor, task kill, SMF recording of the CICS CMF 110 records, and MainView AutoOPERATOR for CICS. These commands also use the CICS External Interface (EXCI) facility to communicate to the target CICS systems. When the PAS is recycled, it uses the EXCI facility to reconnect to CICS regions. Therefore all BBI-SS systems must have the proper security authorization to issue commands.

When CICS security is active with certain SIT parameters, the BBI-SS PAS user ID must have proper security authorization in order to start MainView for CICS transactions, as well as to create and discard transactions and program definitions.
When XCMD=YES and CMDSEC=ALWAYS in the SIT, the BBI-SS PAS address
space user ID must have access defined for the following resources in resource class
CCICSCMD (or site-specified resource class for XCMD):

```
secprfx.EXITPROGRAM  ACCESS(UPDATE)
.PROGRAM      ACCESS(ALTER)
.SYSTEM       ACCESS(READ)
.TRANSACTION  ACCESS(ALTER)
.MONITOR      ACCESS(UPDATE)
.TASK         ACCESS(READ)
.IRC          ACCESS(UPDATE)
.TSQUEUE      ACCESS(READ)
.CONNECT      ACCESS(READ)
.STATISTIC    ACCESS(READ)
.FILE         ACCESS(UPDATE)
```

`secprfx` is the security prefix that is specified by the SECPRFX parameter in the SIT, if
any.

When XPPT=YES or XPCT=YES, and RESSEC=ALWAYS, the BBI-SS PAS user ID
must have ACCESS(ALTER) privileges for resources in the resource classes
MCICSPPT (XPPT) and ACICSPCT (XPCT or site-specified resource class for XPPT
and XPCT) in order to create program and transaction definitions, respectively. See
“Managed resources” on page 50 for the list of programs and transactions.

When XTRAN=YES (regardless of CMDSEC and RESSEC settings), the BBI-SS PAS
user ID must have ACCESS(READ) privileges for transactions (resource) BMCE,
FST2, BCRT, FCD2, JNL2 and FIC2 in resource class TCICSTRN (or site-specified
resource class for XTRAN).

In the target CICS if CICS surrogate user checking is turned on through the SIT
parameter XUSER=YES (regardless of CMDSEC and RESSEC settings), the BBI-SS
PAS user ID must be authorized as a surrogate user in the CICS region. This setting
can be accomplished by using the following RACF command:

```
PERMIT userid1.DFHSTART CLASS(SURROGAT) ID(userid2) ACCESS(READ)
```

`userid1` is the ID of the BBI-SS PAS and `userid2` is the user ID of the CICS region.

If a security manager other than RACF is being used, refer to appropriate security
guide for more information.

Refer to the *CICS RACF Security Guide* for details about CICS security checking.
Refer to the *CICS System Definition Guide* for details about CICS security system
initialization parameters (CMDSEC, RESSEC, XTRAN, XUSER, and so on).

---

**Note**

If CMDSEC and RESSEC are set to ASIS, the only SIT security parameters that affect
the BBI-SS PAS are XTRAN and XUSER.
OLTCNTL parameter ACTNSEC

The MainView OLT component defines a set of options to the CICS/TS INITPARM values.

These options control the initialization processing within the individual CICS/TS region.

When MainView for CICS Action Commands are executed by the CICS/TS regions to alter a managed resource, these commands invoke a CICS/TS SPI command in the CICS/TS region. You can use the OLTCNTL parameter ACTNSEC to control the user ID referenced by the Security Authorization Facility (SAF) RACROUTE authorization checks for specific commands before SPI command executes the Action Command request.

Valid values for the ACTNSEC parameter are NO and YES (ACTNSEC=_NO | YES). These options operate in conjunction with the System Initialization Table parameters of the CICS/TS region and the user ID associated with the Action Command at the time it was invoked.

The special case Action Commands for which MainView for CICS invokes the SAF RACROUTE authorization checks before the CICS SPI command is executed are called Restricted Action Commands.

**ACTNSEC option Restricted Action Commands**

The Restricted Action Commands must be executed within the JNL2 transaction without starting a new task. These commands are listed in the following table. For all other commands, the processing attaches the FCD2 transaction executing for the user ID associated with the Action Command at the time the command was invoked.

When no user ID is associated with the action request, the FCD2 task executes for the user ID associated with the JNL2 transaction.

The Restricted Action Commands do not attach the FCD2 task to process to avoid any possible delay that might occur while attaching a new task. A slight delay could occur if the CICS/TS resources required to attach a new task are not available, and the task might not be dispatched immediately. Therefore, the FCD2 task executes for the user ID associated with the JNL2 transaction.

**Table 16: Restricted Action Commands**

<table>
<thead>
<tr>
<th>Cmd function</th>
<th>Cmd token-1</th>
<th>Cmd token-2</th>
<th>EXEC CICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET</td>
<td>ALTMAXT</td>
<td>null</td>
<td>SET SYSTEM MAXTASKS(nn)</td>
</tr>
<tr>
<td>SET</td>
<td>SYSTEM</td>
<td>MAXTASK</td>
<td>SET SYSTEM MAXTASKS(nn)</td>
</tr>
<tr>
<td>SET</td>
<td>SYSTEM</td>
<td>DSALIM</td>
<td>SET SYSTEM DSALIM(nn)</td>
</tr>
</tbody>
</table>
The OLTCNTL option ACTNSEC enables one of the following approaches to check the user ID authorization for each CICS/TS SPI command executed by MainView for CICS in the CICS/TS region. The CICS/TS SIT options determine which security authorization checks are invoked at the time the CICS/TS SPI command is executed. The ACTNSEC options are:

<table>
<thead>
<tr>
<th>ACTNSEC value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO (default)</td>
<td>Security authorization processing continues to operate as it has in the past. For restricted commands, any security authorization checks at the time the CICS/TS SPI command is executed are against the user ID associated with the JNL2 task. The FCD2 task is never attached for these commands. For unrestricted commands, starts the FCD2 task to process the command. At the time the task is started, continues using the existing logic that tests the command request for a user ID. When the user ID parameter is defined to the MainView AutoOPERATOR command CICSTRAN, starts the FCD2 task for that user ID. Otherwise, the FCD2 tasks run with the same user ID as the JNL2 task.</td>
</tr>
<tr>
<td>YES</td>
<td>The FCD2 task will be attached to process all commands with the exception of restricted commands. For restricted commands, any security authorization checks are processed by the OLT Function Package, before the command is processed by JNL2. The authorization checks are made against the user ID that invoked the command. The FCD2 task is never attached for these commands. For unrestricted commands, when the user ID parameter is defined to the MainView AutoOPERATOR command CICSTRAN, starts the FCD2 task for that user ID. Otherwise, starts the FCD2 task to process the command for the user ID that invoked the command.</td>
</tr>
</tbody>
</table>

Completing implementation

This section gives an overview of the major steps to implement MainView for CICS and MainView AutoOPERATOR for CICS in CICS systems.

Implementation is accomplished by completing a series of procedures. Each step of the overall implementation procedure consists of one or more smaller procedures that must be completed before proceeding to the next step.
Figure 15 on page 111 illustrates the major steps that are used to implement MainView for CICS and MainView AutoOPERATOR for CICS.

Figure 15: Major implementation steps

The following implementation steps are for MainView for CICS only.

1. Modify CICS Startup JCL
2. Update the CSD
3. Define EXCI connection
4. Create PLT
5. Modify PLT
6. Existing PLT?
   Yes
   Implement Security
   Step 7
   No
   Create MCT
   Modify MCT
7. Use IBM-Supplied MCT
8. Specify CMRDETL Data Set
9. Include CICS SDFHEXCI in STEPLIB concatenation
10. AO for CICS only
    Yes
    Modify BBI-SS PAS JCL
    Step 9
    No
    COLD or INITIAL Start CICS
    Step 10
    END
Steps 4 and 6 provide a choice of several procedures to implement the Program List Table (PLT) and Monitoring Control Table (MCT), respectively.

Only the first four steps must be completed to implement MainView AutoOPERATOR for CICS. All steps are required for MainView for CICS. The rest of this chapter shows how to complete the implementation steps shown in Figure 15 on page 111.

**Step 1: Modify CICS startup JCL**

Complete the following procedure for each CICS region that communicates with a BBI-SS PAS. This procedure applies to both MainView for CICS and MainView AutoOPERATOR for CICS.

1. Add the following DD statement to the DFHRPL library concatenation in the CICS execution step:
   
   ```
   //   DD DSN=prefix.BBLINK,DISP=SHR
   ```

2. Provide a BBI-SS PAS directive to the CICS region; follow the procedures described in “Providing a BBI-SS PAS directive to a target CICS region” on page 64.

   If you are using AutoConnect facility to connect a CICS to the BBI-SS PAS, this directive is optional. For more information about AutoConnect, see “BBI-SS PAS and CICS target connection” on page 59.

**Step 2: Update the CSD**

The CICS CSD data set must be updated with the transaction definitions and program resources that are used by MainView for CICS, and MainView AutoOPERATOR for CICS.

The transaction definitions become part of the internal CICS Program Control Table (PCT); the program definitions become part of the internal CICS Processing Program Table (PPT). This procedure modifies or adds resource groups and group lists to the CICS CSD. The resource groups contain the resource definitions for the PCT and PPT that identify MainView for CICS and MainView AutoOPERATOR for CICS transactions and programs to a CICS system. The group lists identify the resource groups that are installed at CICS initialization.

The CSDs for each CICS region that communicate with a BBI-SS PAS must be modified.
The BBSAMP members CMRCSDAL, CMRCSDLA, CMRCSDLB, CMRCSDLM, and CMDCSDEL contain sample JCL with inline program and transaction resource definitions. Use the JCL to define the resource groups that are required for MainView for CICS and MainView AutoOPERATOR for CICS. Each member contains documentation for customizing the JCL.

**To perform the task**

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

CMRCSDAL creates the BMCMVOLT, BMCMVCMR and/or BMCMVAAO groups. These groups contain the following resources:

- **BMCMVOLT** contains common resources (OLTFSET program, FST2 transaction, BMCE transaction).
- **BMCMVCMR** contains MainView for CICS resources (CMRVCMR program, CMRXEIST program).
- **BMCMVAAO** contains MainView AutoOPERATOR for CICS resources (OLTVCAO program).

CMRCSDAL inputs the CSDcccDF members based on your modifications, where ccc is OLT, or CMR, or AAO. For information about the BBSAMP data set, see “BBSAMP data set members” on page 249.

2. Submit the JCL.

3. Check the output and ensure the return code is no higher than 4.

4. Verify that the transaction IDs do not conflict with any existing transaction IDs.

   (If a conflict exists and the existing transaction cannot be changed, see “Resource descriptions and conflicts” on page 243).

5. Modify and run the BBSAMP member CMRCSDLA, CMRCSDLB, or CMRCSDLM.

The CMRCSDAL member explains which member to use. These members modify the group list used at CICS initialization that contains the BMC Software resource groups. For information about the BBSAMP data set, see “BBSAMP data set members” on page 249.
Step 3: Create EXCI connection and session definitions

The BBI-SS PAS uses the EXCI function to reconnect to CICS (after the BBI-SS PAS is recycled) to handle line commands that are issued in the CREGAGT view and to pass the task kill exit table to the task kill exit running in CICS.

For more information, see “CREGAGT view commands” on page 267 and “Implementing the task kill function” on page 155.

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

You must create the connection and session definitions yourself. The connection definitions can be anything you want as long as they specify the EXCI protocol, IRC access method, and CONNTYPE(GENERIC). The CICS supplied group DFH$EXCI can be used. DFH$EXCI has ATTACHCSEC(IDENTIFY) specified; the BBI-SS PAS address space must have a valid user ID associated with it or transaction BMCE will abend with ATCY.

EXCI requires that ISC=YES and IRCSTRT=YES are specified in the SIT. Refer to the CICS External Interfaces Guide for more information about the EXCI interface.

The CREGAGT view provides commands that control the MainView agents (extractor, task kill, MainView AutoOPERATOR for CICS, SMF recording of the CICS CMF 110 records, and auto maintenance). These commands are directed to target CICS systems through EXCI. If you are using DFHXCURM, the EXCI user-replaceable module, to redirect EXCI requests, you must ensure that MainView for CICS requests are not redirected. A MainView for CICS request can be detected by checking the URMPROG parameter in DFHXCURM for the program name "OLTFSET".

Refer to “Site-specific security” on page 107 for important security considerations when you use the EXCI facility.

Next, modify the PLT table to automate initiation of BBI-SS PAS to CICS communication at CICS startup by following the instructions in “Step 4: Modify the PLT” on page 114.

Step 4: Modify the PLT

The Program List Table (PLT) specifies which programs are executed in the post-initialization phase of CICS startup.
This procedure modifies the PLT to initiate BBI-SS PAS to CICS communication at CICS startup. In addition, MainView for CICS data collection begins at CICS startup according to the parameters that are specified with the START statement of the CMRSOPT macro.

Use of the PLT for MainView for CICS and MainView AutoOPERATOR for CICS is optional. However, if you do not use the PLT, you must manually start BBI-SS PAS to CICS communication using either the FST2 transaction from the IBM MVS console, or the INI line command from the CREGAGT view. You can also use these transactions to manually start data collection. For more information about using these transactions, see “Controlling functions manually” on page 229.

Two different procedures are in this section:

- “4A (optional) Modify an existing PLT” on page 115
- “4B (optional) create a PLT” on page 116

Complete one procedure based on whether you have an existing PLT. Either procedure automates the initiation of BBI-SS PAS to CICS communication and MainView for CICS data collection at CICS startup.

4A (optional) Modify an existing PLT

Complete this procedure if you have an existing PLT that you want to modify.

1. Modify the PLT by inserting the appropriate program or by using the CMRPLT macro.

   **Note**
   BMC recommends that you insert the program name manually.

   - To insert the program name manually, find the following statement:
     
     ```
     TYPE=ENTRY,PROGRAM=DFHDELIM
     ```
     Insert the program name immediately after the `PROGRAM=DFHDELIM` statement:
     
     ```
     DFHPLT TYPE=ENTRY,PROGRAM=OLTFSET
     ```
     The task kill function can be started during the CICS PLT processing by specifying TASKKILL=YES in the CMRSOPT options table.

   - To use the CMRPLT macro, insert the CMRPLT macro invocation immediately after the `PROGRAM=DFHDELIM` statement.

2. Assemble the newly revised PLT.
You must include the BBSAMP data set in the assembler SYSLIB concatenation if you use the CMRPLT macro.

3 Add the PLTPI parameter to the SIT to identify this updated PLT to CICS:

```
PLTPI=xx
```

Define `xx` as the suffix of this PLT.

When you have successfully completed the preceding steps, you have modified the PLT for automatic start up of BBI-SS PAS to CICS communication and MainView for CICS data collection.

For MainView AutoOPERATOR for CICS users, implementation is complete. Optional customization, as described in “Customizing MainView AutoOPERATOR for CICS” on page 185, can be performed to adjust MainView AutoOPERATOR for CICS to your site’s requirements.

For MainView for CICS users, next, modify the SIT to define the CICS environment during startup by following the instructions in “Step 5: Modify the SIT” on page 117.

**4B (optional) create a PLT**

Complete this procedure if you create a PLT.

1 Create a new PLT in your CICS table library by using a member name that adheres to the naming conventions at your site.

The new member should contain either a CMRPLT macro invocation or the appropriate program name:

- To use the program name, use these statements:

  ```
  DFHPLT TYPE=INITIAL,SUFFIX=xx
  DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
  DFHPLT TYPE=ENTRY,PROGRAM=OLTFSET
  DFHPLT TYPE=FINAL
  END
  ```

- To use the CMRPLT macro, use these statements:

  ```
  DFHPLT TYPE=INITIAL,SUFFIX=xx
  DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
  CMRPLT MODE=AUTO
  DFHPLT TYPE=FINAL
  END
  ```

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

3 Add the PLTPI parameter to the SIT to identify this updated PLT to CICS:

\[
\text{PLTPI=xx}
\]

where \( xx \) is the suffix of this PLT defined in Step 1 on page 116.

When you have successfully completed the preceding steps, you have modified the PLT for automatic startup of BBI-SS PAS to CICS communication and MainView for CICS data collection.

For MainView AutoOPERATOR for CICS users, implementation for MainView AutoOPERATOR for CICS is complete. Optional customization, as described in “Customizing MainView AutoOPERATOR for CICS” on page 185, can be performed to adjust MainView AutoOPERATOR for CICS to your site’s requirements.

For MainView for CICS users, next, modify the SIT to define the CICS environment during startup by following the instructions in “Step 6: Modify the MCT” on page 118.

**Step 5: Modify the SIT**

Entries in the CICS System Initialization Table (SIT) define the CICS environment during startup. Edit the SIT for each CICS region that communicates with a BBI-SS PAS:

1 Verify the following parameter is specified:

\[
\text{PLTPI=xx}
\]

Define \( xx \) as the suffix of the PLT you updated in “Step 4: Modify the PLT” on page 114.

2 Specify the following CICS monitoring options:

\[
\text{MN=ON} \\
\text{MNPER=ON} \\
\text{MNEXC=ON}
\]

**Note**

If the SIT is not modified as shown here, options are dynamically altered at data collection. If you do not want CICS performance or exception records written to SMF, specify OFF for these values.

3 (optional) Add MCT=2S to use the IBM-supplied Monitoring Control Table (MCT) if you do not have an existing MCT and do not have DBCTL active.
Otherwise, complete the instructions in “Step 6: Modify the MCT” on page 118, and add the suffix for that modified MCT.

4 *(optional)* Specify the following parameter:

\[ \text{MNCONV=YES} \]

If MNCONV=YES, MainView for CICS creates a history record for conversational tasks when a RECEIVE or CONVERSE is made.

5 Specify the following parameters:

\[ \text{IRCSTRT=YES and ISC=YES in the CICS SIT} \]

*Note*

This step is required. The inter-region communications link must be available and open for the MainView for CICS auto maintenance facility to operate.

Ensure that all CICS systems have the CICS SYSIN IRCSTRT parameter set to YES.

In order for the EXCI interface to be used, the IRC function must be available within the target CICS region. The inter-region communications link must be available and OPEN in order for the auto maintenance facility to operate.

6 If you made any changes to the SIT, assemble the newly revised SIT.

*Note*

As an alternative to modifying the SIT, the parameters required by MainView for CICS can be specified as override parameters during CICS initialization, or through CICS SYSIN. Be sure to check these overrides to ensure the table change is effective.

If you change the SIT options described above, CICS recognizes some of these options only after cold starting the system. On a warm or emergency restart, NEWSIT=YES must be specified to enable CICS to recognize the changes.

When you have successfully completed the preceding steps, you have modified the SIT to define the CICS environment during startup.

Next, modify the MCT.

**Step 6: Modify the MCT**

The Monitoring Control Table (MCT) defines which resources are monitored by MainView for CICS for each CICS region.

Modify the MCT if you want MainView for CICS to:
- collect additional performance data for DBCTL resources
- store additional MainView for CICS data in CICS SMF 110 records for use by tools (such as MICS) that process those records
- store additional MainView for CICS collected data specific to DB2 in the CICS SMF 110 performance records

MainView for CICS provides two MCT definitions that can be used to collect additional data for the CICS SMF 110 performance record:

- CMRDATA contains miscellaneous information about IBM DB2, SAP (from SAP AG), ADABAS, MQ, and other databases.
- CMRDB2 contains detailed DB2 information about specific SQL requests. It contains count and time totals.

Either one or both can be specified.

**CICS TS 3.2**

MainView for CICS supports 8-byte timing fields in the two SMF appended areas, which are available for CICS TS 3.2 and later.

Because of the increase in size, you must reassemble your MCT if you are requesting this data in the 110 SMF records. If you do not reassemble the MCT, you might see the FT006x message indicating a data area is too small and no data will be recorded.

MainView for CICS provides new MCT definitions to the BBSAMP library to support CICS TS 3.2 and later. The new MCT definitions are as follows:

- CMRNDMCT
  - CMRDATA MCT definition for CICS TS 3.2 and later
  - counterpart to the CMR$DMCT definition for CICS TS 3.1 and earlier
- CMRN2MCT
  - CMRDB2 MCT definition for CICS TS 3.2 and later
  - counterpart to the CMR$2MCT definition for CICS TS 3.1 and earlier

Because the MCT definition only defines the size of the data area, you can use the new MCT definitions for all CICS releases. You should be aware of the format of the data in the SMF appendages, however. Copybooks in the BBSAMP library describe the format of the data. They are as follows:
CMRNDATA
— CMRDATA data area format for CICS TS 3.2 and later
— counterpart to the CMR$DATA copybook for CICS TS 3.1 and earlier

CMRNDSAS
— CMRDATA data area SAS format for CICS TS 3.2 and later
— counterpart to the CMR$DSAS copybook for CICS TS 3.1 and earlier

CMRNDB2
— CMRDB2 data area format for CICS TS 3.2 and later
— counterpart to the CMR$DB2 copybook for CICS TS 3.1 and earlier

CMRN2SAS
— CMRDB2 data area SAS format for CICS TS 3.2 and later
— counterpart to the CMR$2SAS copybook for CICS TS 3.1 and earlier

The following sections describe how to use these MCT definitions. Additional information about MCT usage is found in the CICS Customization Guide, the CICS System Definition Guide, and the CICS Resource Definition Guide.

Monitor DBCTL resources

CICS provides a copybook member in the SDFHSAMP library that is used to provide DBCTL information in the CICS SMF 110 performance record.

Requesting this DBCTL information through the MCT makes it available to the transaction history record that is produced by MainView for CICS.

To provide the DBCTL information, add the following entry to your MCT:

```
COPY DFH$MCTD
```

This copybook member is available for CICS releases 4.10 and later.

Collect general MainView for CICS data on SMF

MainView for CICS provides a copybook member in the BBSAMP library that can be used to add general MainView for CICS data to the CICS SMF 110 performance record.
This general data is identified as CMRDATA.

To append CMRDATA information to the CICS SMF 110 performance record, add the following statement in your MCT:

```
COPY CMR$DMCT
```

The MainView for CICS BBSAMP library contains two members that illustrate the format of the CMRDATA area. CMR$DSAS contains SAS definitions and CMR$DATA contains an Assembler DSECT.

The appended CMRDATA segment includes data from the T6EUDAT2 user data field in the history record. The T6EUDAT2 field contains 64 bytes of user data populated by the CMRCUEX user exit utility.

When the following conditions exist, the CMRDATA segment is appended to every CICS SMF 110 Task Performance record that is written to the SMF data set:

- You are using CICS TS version 3.2 or later.
- CICS was initialized with SIT parameters specifying MN=ON, MNPER=ON, and MCT=xx, where xx specifies the MCT that includes the CMRDATA segment definition.

The CMRCMPWN utility, which can convert MainView for CICS T6E history records to the CICS SMF 110 format, also provides the new data in the CMRDATA segment. The CMRDATA segment is always added to the CICS SMF 110 Task Performance records when this utility converts MainView for CICS history (CMRDETL) records to the SMF 110 format.

**Collect DB2 data that is specific to MainView for CICS on SMF**

MainView for CICS provides a copybook member in the BBSAMP library that can be used to add DB2 data that is specific to MainView for CICS to the CICS SMF 110 performance record.

This data is identified as CMRDB2.

To append CMRDB2 information to the CICS SMF 110 performance record, add the following statement in your MCT:

```
COPY CMR$2MCT
```

The MainView for CICS BBSAMP library contains two members that illustrate the format of the CMRDB2 area. CMR$2SAS contains SAS definitions and CMR$DB2 contains an Assembler DSECT.
The CMRDB2 information is provided for CICS 4.1 and later, and DB2 5.1 and later.

**Step 7: Implement security**

The CREGAGT views provide action commands to control MainView for CICS initialization and termination, and its agent functions-- extractor, task kill, and MainView AutoOPERATOR for CICS.

These commands use the CICS External Interface (EXCI) facility to communicate to the target CICS systems. When the BBI-SS PAS is recycled, it uses the EXCI facility to reconnect to CICS regions. Therefore all BBI-SS systems must have the proper security authorization to issue create and discard commands, and to start the CMR transactions.

See “Site-specific security” on page 107 for more information about implementing security.

**Step 8: Specify CMRDETL data sets**

Specify the CMRDETL data sets that are needed for the target CICS regions.

Information about how to create the CMRDETL definitions and how to define the data sets is available in “Using CMRDETL data sets” on page 77.

**Step 9: Modify the BBI-SS PAS JCL**

The external CICS interface modules (EXCI) are supplied in the CICS data set SDFHEXCI. The SDFHEXCI data set must be available to the MainView for CICS and MainView AutoOPERATOR for CICS BBI-SS PAS.

1. The BBI-SS PAS can access only the modules in SDFHEXCI through either the LINKLIST or the PAS STEPLIB.

   The BBI-SS PAS cannot access the modules from any other libraries, including DFHLPA. 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.

   EXCI is downward compatible and IBM recommends that the highest level available should be used. 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:

//STEPLIB DD DISP=SHR,DSN=hilevel.BBLINK
//          DD DISP=SHR,DSN=CICS510.SDFHEXCI (FOR EXCI)

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

An example follows:

//SYSMDUMP DD SYSOUT=* 

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,VOLSER=.....,UNIT=....

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

Refer to the _CICS External Interfaces Guide_ for more information.
Optional implementation procedures

This chapter describes optional procedures that you can use to customize MainView for CICS.

You can install and run MainView for CICS and MainView AutoOPERATOR for CICS as delivered. However, BMC recommends that you review the default values to make sure they work for your environment.

Setting monitoring conditions (CMRSOPT)

The CMRSOPT options table adapts MainView for CICS operations and monitoring to your specific CICS environment.

This options table is multipurpose and it specifies:

- Transactions excluded from monitoring
- Start up of agent functions (extractor, task kill, program timing, and so on)
- Monitored resources (terminals, transactions, files, and so on)
- Data collection intervals control how often CICS statistics are written to SMF where data can be converted to CMRSTATS records by the CMRSTATS utility
- Thresholds of system-wide response time service levels
- Data collection routines for supported fourth generation languages (4GLs) and database management systems
- Graphing resources (terminals, transactions, files, and so on)
  Note that selective resource level graphing is only available on an assembled/linked CMRSOPT.
Default settings for the CMRSOPT options table are specified in BBSAMP member CMRUSOPT and are assembled and linked into the distributed load module CMRSOPT. No modifications are required, but you should review the settings to make sure they work for your environment.

**Note**

By default, MainView for CICS monitoring begins at PLT or manual startup. If you want to manually start MainView for CICS monitoring, refer to “Controlling functions manually” on page 229.

The CMRSOPT can be maintained with an online administrative function from the CMRSOPT view, or it can be assembled and loaded based on the parameters that are provided by the CMRSOPT macro. For greater ease of use, BMC recommends using the CMRSOPT view instead of using the CMRSOPT macro.

When the CMRSOPT is maintained with the online administrative function, data is stored in the BBIPARM data set with the member name of CMRSOPxx, and you can access the data through the CMRSOPT view. No user modifications are required because the earlier assembled CMRSOPT modules are also supported. To use the administrative function, you must first use the CMRSOPT views to add the default entry in the BBIPARM data set. For more information, see the MainView for CICS Online Services Reference Manual.

When a target CICS connects to the BBI-SS PAS for the first time, the CMRSOPT is loaded into ECSA. MainView for CICS looks for CMRSOPT in the following sequence:

- Attempts to locate a CMRSOPxx member in the BBI-SS PAS BBIPARM concatenation; if one is found, MainView for CICS uses it.
- CMRSOP00 is the default member name. The BBCFG facility can be used to specify a different suffix, as described in the MainView Customization Reference.

If MainView for CICS does not find a definition for the current target in the member, processing continues to the following steps to locate the appropriate load module:

- Attempts to load a module with the target name; if one is found, MainView for CICS uses it
- Attempts to load module CMRSOPT (default in BBLINK)

When a CMRSOPxx is found in BBIPARM, MainView for CICS builds a CMRSOPT load module from the data. MainView for CICS uses the created module to build the internal CMRSOPT data areas.

The CMRSOPT is refreshed when:
- the BBI-SS PAS is recycled
- CICS is recycled
- FST2 INIT is processed after a FST2 TERM
- one of the RESET commands is issued

See “Resetting the CMRSOPT” on page 149 for information about the RESET commands.

All features that are provided by the CMRSOPT macro are supported in the administrative views with the exception of selective resource graphing (TYPE=resource). Although global graphing is supported (system, storage, and so on), you cannot perform selective file, transaction, program, terminal, PSB or BMS resource graphing through the PARMLIB CMRSOPT.

To use the CMRSOPT administrative views, see the MainView for CICS Online Services Reference Manual. For information about the individual parameters, see “CMRSOPT parameters” on page 127. For more information about the CMRSOPT macro, see Modifying the CMRSOPT with the macro on page 145.

**CMRSOPT parameters**

This section describes the parameters that provide initial system settings for MainView for CICS operations and monitoring.

The parameters are broken into the following subsections:

- target selection
- default monitoring conditions
- databases and fourth-generation languages (5GLs)
- additional data types
- SMF data recording

Unless otherwise noted, these parameters are available on the CMRSOPT view or in coding the TYPE=INITIAL statement of the CMRSOPT macro.

**Target selection**

This section describes the parameter that is used to set the CICS targets that will make use of a specific CMRSOPT parameter table from the CMRSOPT view.
**TARGET=xxxxxxx**

On the CMRSOPT view (not available in the CMRSOPT macro) this parameter identifies the target or targets for which the BBIPARM data is used. At least one target name must be supplied. Additional names can be supplied up to the limit of the CMRSOPT view’s input lines.

The generic characters + and * can be used in a target name as follows:

- + designates the existence of any single character (can be used multiple times in a target name).
- * designates the existence of zero or more characters starting from that point (can be used once in a target name).

---

**Note**

You can turn on data collection for program timing, session and connection data collection, temporary storage, and transient data queue data with the CMRSOPT view but there are no parameters to perform equivalent functions in the CMRSOPT macro.

---

**Default monitoring conditions**

This section describes parameters that set the global default monitoring environment for MainView for CICS. The default values for the parameters are underlined.

**START=**YES | NO

This parameter specifies whether data collection begins when a PLT entry activates MainView for CICS at CICS startup (see “Step 4: Modify the PLT” on page 114).

- YES starts data collection and establishes communication with the BBI-SS PAS.
- NO specifies that data collection does not start when communication to the BBI-SS PAS is established. Data collection can be started through the QON line command from the CREGAGT view; the FST2 transaction from a CICS console; or the FST2 transaction from a CICS terminal.

**SHTDLY=(0-10, 0-10)**

On the CMRSOPT view (not available in the CMRSOPT macro), this parameter specifies how many seconds to delay the MainView for CICS agents shutdown at CICS termination and the number of times MainView for CICS agents will wait for this interval before shutting down at CICS termination.
CMRDETL=YES | NO

This parameter specifies whether detail transaction records are stored on the CMRDETL data set or data space.

- YES specifies that detail transaction records are to be recorded.
- NO specifies that detail transaction records are not to be recorded. Data is still collected for online access to data in views and displays.

CRGRESPLOG=YES | NO

This parameter determines whether MainView for CICS generates the FT1010W message when average response time exceeds the value defined for the @RESP CREGION monitor, which is automatically started by MainView for CICS for each connected CICS region.

- YES enables message generation.
- NO specifies that message generation is suppressed.

DLOGPROC=DEFAULT | TRSIZE

This parameter defines the detail trace log block size for application tracing.

- DEFAULT means 48K block size is used.
- TRSIZE specifies that block size is determined by the TRSIZE parameter for the detail trace startup.
**DTRACSIZ=25 | nnn**

This parameter specifies the maximum number of megabytes (MB) in the CICS data space to be used for detail tracing.

Valid values are 0 to 1024. A value of 0 means no detail tracing will be performed.

The sum of the DTRACSIZ and STRACSIZ values cannot exceed 1024 MB.

**DYNAMIC=0 | nnnn**

This parameter specifies the maximum z/OS high private storage (non-DSA) used by MainView for CICS in kilobytes (KB).

The storage holds statistical data for DB2 plan activity. The PLAN display, (replaced with the CDB2ENT and CDB2TRN views) uses the data. When you use the default value of 0, data collection for the PLAN display is suppressed. If you are using the PLAN display, MainView for CICS collects an average of 24 KB of recorded data for every 100 DB2 plans. For better system performance, BMC recommends that you specify at least 1000 KB.

Specifying DYNAMIC=0 does not affect the collection of statistics for DB2 plan access for each transaction.

**Note**

MainView for CICS will round up to the nearest multiple of 8 any value that is greater than 0 and is not a multiple of 8.

BMC plans to withdraw support for the PLAN display in a future release of MainView for CICS.

**EXCLUDE=YES | NO**

This parameter specifies whether a CICS transaction should be excluded from monitoring.

- **YES** specifies that specified CICS transactions in the TYPE=EXCLUDE statement will not be monitored.

- **NO** specifies that all transactions are monitored.

If you specify YES, an entry must be made in the CMRSOPT table that identifies the transactions to be excluded from monitoring:

```
TYPE=EXCLUDE, NAME=(FIC2, CE+F, CSA*)
```
Excluded transactions cannot be monitored, traced, or graphed.

**FIC2=** **YES | NO**

On the CMRSOPT view (not available in the CMRSOPT macro), this parameter specifies whether the MainView for CICS agent starts the FIC2 transaction in CICS to collect CICS and MainView for CICS statistics.

The CICS statistics are the ones you selected under section SMF Data Recording Periods in CMRSOPT. The following list describes where MainView for CICS statistics are used:

- All full-screen graphic displays
- Most fields in the full-screen SUMMARY display
- Message FT046 CICS CPU usage monitoring provided by monitors @GLBT and @PRB4, PROBLEM display and the CREGPRB view

If FIC2 is not started, the data is not collected. If you do not need both SMF statistics and CMR statistics, you can set this value to NO.

**FILEOUT=** **YES | NO**

This parameter specifies whether CICS resource usage are included in a transaction detail record.

- YES specifies that detail records include CICS file and database usage for each transaction.
- NO specifies that detail records do not include a transaction’s CICS file and database usage. NO uses less DASD space, but data is not collected for individual file activity.

**FILEXPND=NO | YES**

This parameter specifies whether expanded resources should be collected and written to the CMRDETL history data set. Expanded information includes individual function counts and timings based on type. It also includes volume serial numbers for files and the complete 48-byte IBM WebSphere MQ queue name if MQFILE=YES.

- NO specifies that expanded file information is not collected.
YES specifies that expanded information is collected and stored in the CMRDETL history data set.

See also information about the following parameters:

- “FILEOUT=YES | NO” on page 131
- “MAXFILE=20 | nnn” on page 132
- “PSB=YES | NO” on page 140
- “DB2=YES | NO” on page 136
- “MQFILE=NO | YES” on page 134
- “MQFOFF=0 | nn” on page 134

**GPREAD**

This parameter defines the cycle interval for graph frames (bars) in minutes.

Valid values are 2 to 30.

**INCLUDE=YES | NO**

This parameter is not supported in the CMRSOPT macro. On the CMRSOPT view, you can use this parameter to specify whether the CICS system transactions CSNC, CSNE, and CSSY should be included for monitoring.

- YES specifies that the system transactions CSNC, CSNE, and CSSY will be monitored.
- NO specifies that they will not be monitored (the default).

**MAXFILE=20 | nnn**

This parameter specifies the maximum number of resource entries that can be included in a single CMRDETL record. Resources include DB2, DBCTL and other databases, file names, MQ queue names, CPU timing program names, and session and connection data.

Valid values are 1 to 255 if FILEXPND=NO (the default) is specified. Valid values are 1 to 75 if FILEXPND=YES is specified.

When you specify the MAXFILE parameter, you must also specify FILEOUT=YES (the default).
Increasing the potential number of resources that can be included in a 6E or 6D transaction detail record might affect the amount of required DASD storage allocated for the CMRDETL data set. Each resource entry increases the size of a detail record by 16-22 bytes (FILEXPND=NO) or 220-298 bytes (FILEXPND=YES). The record size of a CMRDETL data set should be increased to accommodate larger average sized 6D and 6E detail records. Sample member CMRDDTL of your BBSAMP data set contains the following statement as part of an IDCAMS command to define the CMRDETL data set.

**RECORDSIZE(380 32736)**

The first number within the parentheses, 380, is the default average size of a detail record. The second number reflects the maximum size of a CMRDETL data set record.

With experience, you should be able to estimate the average number of files included in a typical transaction detail record. You may want to change the default average record size to more accurately reflect your site’s conditions. You can avoid frequent CMRDETL set switching when the average is too small. You can avoid infrequent switching when the average record size is too large also.

“Step 8: Specify CMRDETL data sets” on page 122 describes the procedure to define CMRDETL data sets.

**MAXLOST=256 | nnnnn**

This parameter specifies the number of detail transaction records that can be lost before MainView for CICS stops collecting data because of insufficient queue space.

Valid values are 1 to 32767.

**MAXLOSTT=256 | nnnnn**

This parameter specifies the number of detail trace records that can be lost before application tracing is quiesced because of insufficient queue space.

Valid values are 1 to 32767.

**MAXRESP=null | 20 | nnn**

This parameter defines the maximum acceptable transaction response time service level in tenths of a second. This value is the default for all transactions.

Valid values are 1 to 999.
**MQFILE=NO | YES**

This parameter determines whether WebSphere MQ detail information is collected.

- **NO** specifies that WebSphere MQ detail information is not collected; information is combined into one resource called MQSERIES.

- **YES** specifies that extended function count and timing data is to be collected for all WebSphere MQ queues, and that a detail trace of a transaction that accesses WebSphere MQ resources will provide information about various function calls, including the OPEN, CLOSE, GET, PUT, and PUT1 calls, for each queue.

  If YES is specified, FILEXPND=YES must also be specified. See “FILEXPND=NO | YES” on page 131. See also the MQFOFF parameter.

**MQFOFF=0 | nn**

This parameter specifies the displacement into the 48-byte WebSphere MQ queue name that begins the 8-byte name to be used by MainView for CICS, if MQFILE=YES is also specified.

This number is relative to zero. The default setting of 0 means the 8-byte file name is the first 8 bytes.

*Tip*

Specify a value for MQFOFF that ensures unique 8-byte queue names are used by MainView for CICS. For example, for the following queue names, you could specify MQFOFF=13 to use the last 8-bytes of the names:

- MQ.MQSYSA.CICS1.Q1LOG
- MQ.MQSYSA.CICS1.Q2LOG
- MQ.MQSYSA.CICS1.Q2MSG
- MQ.MQSYSA.CICS2.Q1LOG
- MQ.MQSYSA.CICS2.Q2LOG
- MQ.MQSYSA.CICS2.Q2MSG

**NTGRAPH=NO | YES**

This parameter specifies whether non terminal attached transactions are included in the Graph Global System display and LTX (Service Level Response) CMRSTATS records.

- **NO** specifies that only terminal attached transactions are included.
YES specifies that terminal-attached and non terminal-attached transactions are included.

**STRACSZ=25 | nnn**

This parameter specifies the maximum number of megabytes (MB) in the CICS data space to be used for summary tracing.

Valid values are 0 to 1024. A value of 0 means no summary tracing will be performed.

The sum of the STRACSZ and DTRACSZ values cannot exceed 1024 MB.

**TASKKILL=NO | YES**

This parameter specifies whether the task kill function is to be initiated.

This option is checked only at PLTPI time or on the first INIT request and is ignored on subsequent INIT requests.

The task kill function can be controlled by using the FST2 transaction KON, KOFF and KRLD commands, or from the CREGAGT view by using the KON, KOF and KRL line commands.

**USENETNM=NO | YES**

This parameter specifies whether the netname should be used in MainView for CICS displays and tracing.

- NO specifies that the netname is not used in displays and tracing.
- YES specifies that the netname appears in the History Summary and CICS Trace Entries displays. In addition, CICS traces can be initiated based upon the netname.

**WAITINT=100 | nnn**

This parameter specifies the frequency, in hundredths of a second, with which MainView for CICS checks for:

- CICS regions that have started or stopped
- BBI-SS PAS termination requests

Valid values are 010 to 999.
Databases and Fourth-Generation Languages (4GLs)

This section describes parameters for the databases and 4GLs that can be monitored by MainView for CICS. The default values for the parameters are underlined.

Note
CMRSOPT no longer includes parameters for the following databases and 4GLs:

- Adabas
- DATACOM/DB
- GENER/OL
- Natural
- SUPRA
- SYS2K

Support for these applications is provided by user exits in the BBSAMP data set. For a list of BBSAMP members, see “BBSAMP data set members” on page 249.

CSP=YES | NO , CSPPGM=CSP0001 | xxxxxxxxx

CSP specifies monitoring of the IBM Cross System Product (CSP) panel.

CSPPGM

CSPPGM specifies the name of the IBM-supplied CSP control/driver program specified in the PCT entry for the CSP main transaction. If CSPPGM is used, CSP also must be specified.

DB2=YES | NO

DB2 specifies monitoring of a DB2 system. Values are as follows:

- YES specifies that DB2 plans are monitored and that detail traces will include DB2 SQL calls.
- NO specifies that DB2 plans are not monitored.

DMS=YES | NO

DMS specifies monitoring of the DMS (Development Management System) panel.
DMSOFF=124 | 114

DMSOFF specifies the offset of the DMS program name in the transaction work area for specific releases of DMS.

The value is hexadecimal 024 when used in the CMRSOPT macro. On the CMRSOPT view, the value is decimal notation 036.

IDEAL=NO | YES

IDEAL specifies monitoring of the CA-IDEAL panel.

MANTIS=NO | n. n[.MANPGM=MANTIS | xxxxxxx][.MANPGM2=MANTIS2 | xxxxxxx][.MANPGM3=MANTIS3 | xxxxxxx]

MANTIS specifies monitoring of the Mantis panel.

Define n.n as one of the following versions of Mantis:

- MANTIS=4.0
- MANTIS=4.2 (also valid for version 4.25)
- MANTIS=5.2 (also valid for version 5.25)
- MANTIS=5.6
- MANTIS=6.0

MANPGM

MANPGM specifies the name of the distributed Mantis module, where xxxxxxx is the name specified in the PROGRAM parameter of the PCT entry for the Mantis transaction. If MANPGM is used, MANTIS must also be specified.

MANPGM2

MANPGM2 specifies the name of an additional, optional Mantis module.

MANPGM3

MANPGM3 specifies the name of an additional, optional Mantis module.
MILLEN=NO | YES[, MILPGM1=MILLMDIO | xxxxxxxx] [, MILPGM2=M2LLMDIO | xxxxxxxx]

MILLEN specifies monitoring of the Millennium panel. Values are as follows:
NO specifies that millennium monitoring is not activated.
YES specifies that millennium monitoring is activated.

MILPGM1

MILPGM1 specifies the name of the distributed Millennium module, where xxxxxxxx is the name specified in the PROGRAM parameter of the PCT entry for the Millennium transaction. If MILPGM1 or MILGM2 is used, MILLEN also must be specified.

MILPGM2

MILPGM2 specifies the name of an additional, optional Millennium module.

PCS=NO | YES[, PCSPGM=DOCSCMAN | xxxxxxxx] [, PCSOFF=0120 | nnnn]

PCS specifies monitoring of the PCS (Patient Care System) panel. If PCSPGM or PCSOFF is used, PCS also must be specified.

PCSPGM

PCSPGM specifies the name of the distributed PCS, where xxxxxxxx is the name specified in the PROGRAM parameter of the PCT entry for the PCS transaction.

PCSOFF

PCSOFF specifies the offset of the screen name (TWASCNAM) in the PCS transaction work area (TWA). Obtain the offset from an Assembly listing of DOCSCMAN. Specify the PCSOFF value as a four-byte hexadecimal with leading zeroes.

The value is hexadecimal 120 when used in the CMRSOPT macro. On the CMRSOPT view, the value is decimal notation 288.

UFO=NO | n. n[, UFGPM=UFOFASTM | xxxxxxxx]

UFO specifies monitoring of the UFO panel.

Define n.n as one of the following versions of UFO:
- UFO=2.4
- UFO=2.5
- UFO=2.6
- UFO=3.0 (version 3.0 and later)

**Note**

UFO version 3.0 and later is supported by the UFO accounting exit. To activate this support, CMRUFOX must be specified in the UFOINIT parameter ACCEXIT (see your UFO customization guide for more information). For this support to be available, UFO version 3.0 must be at maintenance level 3.0.94 or PTFs UF30052 and UF30071 must be applied.

**UFOPGM**

UFOPGM specifies the name of the distributed UFO module, where xxxxxxxx is the name that is specified in the PROGRAM parameter of the PCT entry for the UFO transaction. If UFOPGM is used, UFO also must be specified.

**Additional data types**

This section describes parameters that can be used to specify monitoring of additional data types. The default values for the parameters are underlined.

**BMS=NO|YES**

BMS activates an intercept to collect data from the BMS map set.

**DLICLK1=1 | n**

DLICLK1 identifies the first of four DBCTL clocks to be monitored. Subsequent user clocks must be contiguous. A value of 0 indicates no DBCTL measurement.

**DLICLKN=0 | n**

DLICLKN identifies the last DBCTL user clock to be monitored. The default is 0, or the DLICLK1 value + 3, if DLICLK1 is specified.
**DLICNT1=** \(1 \mid n\)

DLICNT1 identifies the first of 10 user counters to be applied to the DBCTL clocks. Subsequent user counters must be contiguous. A value of 0 indicates no DBCTL measurement will occur.

**DLICNTN=** \(1 \mid n\)

DLICNTN identifies the last of 10 user counters to be applied to the DBCTL clocks. The default is 1 or the DLICNT1 value + 9, if DLICNT1 is specified.

**PSB=** \(YES \mid NO\)

PSB indicates whether DBCTL PSB data is collected.

**SERVICE=(x1,...,x17)**

SERVICE sets a system-wide range of transaction response time service levels that are used by the CMRSTATS type CA record. The record is only created if LTX= is specified in CMRSOPT. Values are expressed in tenths of a second.

Valid values are 1 to 999. Service levels must be specified in ascending order. The following shows the default value:

SERVICE=(5,10,15,20,25,30,40,50,60,70,80,90,100,200,300,600,900)

Up to 17 service levels can be specified. When fewer than 17 levels are specified, the CMRSOPT macro increments the remaining unspecified values by 10.

**SMF data recording**

This section describes parameters that are used to set the interval to request CICS to write the specified SMF 110 Type x'0002' and Type x'0B02' (statistic) records to the SMF data set. The default values for the parameters are underlined.

This process is performed by the MainView for CICS FIC2 transaction. For more information about how to use these records, see the MainView for CICS PERFORMANCE REPORTER Data Reference and MainView for CICS PERFORMANCE REPORTER User Guide.
T6F=NO | YES

T6F specifies whether MainView for CICS global performance (CC) records are written to the SMF data set.

CSA=0 | nnn

CSA specifies the interval in minutes for recording CICS system statistics to the SMF data set. Includes dispatcher statistics.

Valid values are 0 to 999. If 0 is specified, CICS system statistics are not recorded.

DCT=0 | nnn

DCT specifies the interval to record destination control table statistics to the SMF data set.

Valid values are 0 to 999 minutes. If 0 is specified, destination control statistics are not recorded.

DLZ=0 | nnn

DLZ specifies the interval to record DBCTL system statistics to the SMF data set.

Valid values are 0 to 999 minutes. If 0 is specified, DBCTL system statistics are not recorded.

DMP=0 | nnn

DMP specifies the dump statistics collection interval. These statistics include system and transaction dump information.

Valid values are 0 to 999 minutes. If 0 is specified, dump statistics are not recorded.

The CICS request is:

EXEC CICS PERFORM STATISTICS REORD SYS_DUMP TRANDUMP

ENQ=0 | nnn

ENQ specifies enqueue statistics collection interval.

Valid values are 0 to 999 minutes. If 0 is specified, statistics are not recorded.
**FCT**=0 | nnn

FCT specifies the interval to record file control statistics to the SMF data set. Valid values are 0 to 999 minutes. If 0 is specified, file control statistics are not recorded.

**FEPI**=0 | nnn

FEPI specifies the FEPI statistics collection interval. Valid values are 0 to 999 minutes. If 0 is specified, FEPI statistics are not recorded.

**GNI**=0 | nnn

GNI specifies the interval to record global network statistics to the SMF data set. Valid values are 0 to 999 minutes. If 0 is specified, global network statistics are not recorded.

**IRC**=0 | nnn

IRC specifies the interval to record IRC/ISC (inter-region and intersystem connection) statistics to the SMF data set. Valid values are 0 to 999 minutes. If 0 is specified, IRC/ISC statistics are not recorded.

**JCT**=0 | nnn

JCT specifies the interval to record journal control table statistics to the SMF data set. Includes log stream statistics. Valid values are 0 to 999 minutes. If 0 is specified, journal control table statistics are not recorded.

**LTX**=0 | nnn

LTX specifies the interval to record service level statistics to the SMF data set. Valid values are 0 to 999 minutes. If 0 is specified, service level statistics are not recorded.
**MON=0 | nnn**

MON specifies the monitor statistics collection interval.

Valid values are 0 to 999 minutes. If 0 is specified, monitor statistics are not recorded.

**OTHR=0 | nnn**

OTHR specifies the other statistics collection interval. These statistics include recovery manager, statistics domain, and table manager information.

Valid values are 0 to 999 minutes. If 0 is specified, the recovery manager, statistics domain, or table manager statistics are not recorded.

The CICS request is:

EXEC CICS PERFORM STATISTICS RECORD RECOVERY STATS TABLEMGR

**PAM=0 | nnn**

PAM specifies the interval to record DSA or page allocation map statistics to the SMF data set.

Valid values are 0 to 999 minutes. If 0 is specified, DSA or page allocation map statistics are not recorded.

**PCT=0 | nnn**

PCT specifies the interval to record program control table statistics to the SMF data set.

Valid values are 0 to 999 minutes. If 0 is specified, program control table statistics are not recorded.

**POL=0 | nnn**

POL specifies the interval to record Local Shared Resource (LSR) pool statistics to the SMF data set.

Valid values are 0 to 999 minutes. If 0 is specified, LSR pool statistics are not recorded.
**PPT=0 | nnn**

PPT specifies the interval to record processing program table statistics to the SMF data set. Includes program auto-install statistics.

Valid values are 0 to 999 minutes. If 0 is specified, processing program table statistics are not recorded.

**RCT=0 | nnn**

RCT specifies the interval to record DB2 Plan activity to the SMF data set.

Valid values are 0 to 999 minutes. If 0 is specified, DB2 Plan activity statistics are not recorded.

**STI=0 | nnn**

STI specifies the interval to record System Initialization Table (SIT) statistics to the SMF data set.

Valid values are 0 to 999 minutes. If 0 is specified, SIT statistics are not recorded.

**TCT=0 | nnn**

TCT specifies the interval to record Terminal Control Table (TCT) statistics to the SMF data set.

Valid values are 0 to 999 minutes. If 0 is specified, TCT statistics are not recorded.

**TST=0 | nnn**

TST specifies the interval to record temporary storage table statistics to the SMF data set.

Valid values are 0 to 999 minutes. If 0 is specified, temporary storage table statistics are not recorded.

**WEB=0 | nnn**

WEB specifies the web statistics collection interval.

Statistics produced include:

<table>
<thead>
<tr>
<th>TCPIPSERVICE</th>
<th>JVMPROGRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>JVMPROFILE</td>
<td>BEAN</td>
</tr>
<tr>
<td>IPCONN</td>
<td>MQCONN</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>JVMPOOL</td>
</tr>
</tbody>
</table>
Valid values are 0 to 999 minutes. If 0 is specified, statistics for these items are not recorded.

---

**Note**

If you plan to use CICS statistic records for reporting, you can set the SMF parameters to the appropriate interval in minutes and set T6F to YES in CMRUSOPT.

---

## Modifying the CMRSOPT with the macro

Use the following procedure to modify the CMRSOPT using the macro:

1. Copy member CMRASM from BBSAMP to the UBBSAMP data set, and make sure the new member name conforms to your site’s naming conventions.

2. Read the comments in the member that was created in step 1, and then modify the JCL accordingly.

Keep the following information in mind:

- CMRSOPT must be assembled and linked into BBLINK or an authorized load library that is concatenated in front of BBLINK.
- CMRSOPT must be linked non-reentrant.
- You must include the BBSAMP data set in the assembler SYSLIB concatenation.

3. Modify the sample CMRUSOPT according to your site’s monitoring requirements.

A sample CMRUSOPT is in a member of the same name in your BBSAMP data set. CMRUSOPT contains the default specifications and it is used to create the distributed module CMRSOPT. You should edit this member for your site’s monitoring requirements.

---

**Note**

If you want to set the same monitoring conditions for all your CICS regions, name the load module CMRSOPT. If you need different monitoring conditions for each region, rename the load module to the name of the CICS region with which it is associated (for example, CICSPROD). CMRSOPT is the default name if there are no load modules with a region name. BMC recommends that you use the CMRSOPT administration views instead of editing the CMRUSOPT module.
The CMRSOPT module contains three types of statements:

- TYPE=INITIAL
- TYPE=resource
- TYPE=FINAL

Each statement includes a group of parameters that specify characteristic monitoring conditions. The following sections describe each that is parameter associated with the three CMRSOPT statements.

4 Submit the JCL and check the job output to make sure all steps are completed with a return code of 0.

---

**Note**

S106 abends can occur if the newly link-edited module causes the BBLINK data set to enter secondary extents and the active BBI-SS PAS attempts to load the module. If this situation happens, stop the BBI-SS PAS and restart it.

---

**TYPE=INITIAL statement (default conditions)**

The TYPE=INITIAL statement includes parameters that set the default global monitoring conditions for all CICS regions.

Parameters are grouped by:

- default monitoring conditions
- databases and fourth-generation languages (4GLs)
- additional data types
- SMF monitoring conditions

The individual parameters are described in “CMRSOPT parameters” on page 127.

**TYPE=resource statement (optional)**

The TYPE=resource statement identifies:

- Monitored resources displayed in online full-screen graph displays
- Type of monitor area used
Default maximum response time (service level)

Multiple TYPE=resource statements can be specified in CMROPT. The TYPE=resource statements included as examples in the CMRUSOPT sample member monitor only files and transactions. These statements are commented out. Unless you change these statements, no graph data is collected.

The TYPE=resource statement specifies the type of resource that is monitored. Valid parameters used with this statement are as follows:

**Table 17: Resource types**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Resource type</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE=FILE</td>
<td>files and databases</td>
</tr>
<tr>
<td>TYPE=TRAN</td>
<td>transactions</td>
</tr>
<tr>
<td>TYPE=TERM</td>
<td>terminals</td>
</tr>
<tr>
<td>TYPE=PANEL</td>
<td>4GL panels</td>
</tr>
<tr>
<td>TYPE=MAP</td>
<td>BMS map set names</td>
</tr>
<tr>
<td>TYPE=PSB</td>
<td>DBCTL PSB names</td>
</tr>
</tbody>
</table>

Additional parameters included with TYPE=resource statements specify how these resources are monitored.

Typically, several parameters are used with the TYPE=resource statement, as shown in the following example:

```
TYPE=resource,NAME=group,MONITOR=class,MAXRESP=nnn
```

The parameters are described in the following list:

**NAME=ALL | name**

This parameter names the monitored resource group.

ALL specifies that all resources corresponding to the defined TYPE are monitored. As each new resource is executed, it is added dynamically to the monitor table. Virtual storage is acquired for that resource’s statistical counts. If ALL is specified, it must precede all other entries for the same resource type.

name identifies the specific or generic name of a resource to be monitored. This name can be used to set specific service levels or to produce a single monitored entry for a group of generic entries.

The resource names must be specified in ascending sequence following the NAME=ALL parameter, if it is present. Generic entries include minus signs (-) to
indicate positions not included in the comparison. If a generic resource name matches a specific entry, the resource statistics are recorded in the specific entry.

**MONITOR=MEDIUM | DL/I | FILE | LONG**

This parameter defines the monitoring class. Storage for monitoring classes is acquired in extended private storage for the BBI-SS PAS address space. The following estimates are per monitored resource. Valid values and the amount of virtual storage required for each class are as follows:

- **MEDIUM** specifies to keep timings only (1358 bytes).
- **DL/I** specifies to keep timings and usage counts plus DBCTL timings and counts (3428 bytes).
- **FILE** is used only with file monitor (228 bytes).
- **LONG** specifies to keep timings and usage counts (2758 bytes).

**MAXRESP= nnn**

This parameter sets an individual response time service level for a resource defined with the TYPE=resource statement. It overrides the default service level set by the MAXRESP parameter of the TYPE=INITIAL statement.

It applies only to the resource type being used. Valid values are 1 to 999 and represent tenths of a second; for example, 245 is 24.5 seconds. When MONITOR=FILE is specified, the value represents milliseconds.

The following examples show the use of TYPE= resource statements for CICS transactions:

```
CMRSOPT TYPE=TRAN,NAME=ALL,MAXRESP=10,MONITOR=MEDIUM
All transactions are added dynamically to the monitor table, given a service level of 1.0 seconds, and use a medium monitor area.

CMRSOPT TYPE=TRAN,NAME=C---,MAXRESP=15,MONITOR=MEDIUM
All transactions that begin with C are combined and stored into a single entry (C---) with a service level of 1.5 seconds, and use a medium monitor area.

CMRSOPT TYPE=TRAN,NAME=CEMT,MAXRESP=20,MONITOR=LONG
CEMT monitoring statistics are collected with a service level of 2.0 seconds and a long monitor area.

CMRSOPT TYPE=TRAN,NAME=MENU,MAXRESP=5,MONITOR=DL/I
```
MENU monitoring statistics are collected with a service level of 0.5 seconds and a DBCTL monitor area.

**TYPE=FINAL statement (required)**

The TYPE=FINAL statement ends option generation. A TYPE=FINAL statement must be included in CMRUSOPT.

**Resetting the CMRSOPT**

CMRSOPT can be reset dynamically with the RESET command as follows:

The following command resets monitoring options for all CICS regions:

```
.RESET PARM CMRSOPT ALL
```

The following command resets monitoring options that apply to a specific CICS region.

```
.RESET PARM CMRSOPT cicsName
```

See the *MainView Administration Guide* for more information about the RESET command and its parameters.

You can also reset the CMRSOPT with the SRL command on the CREAGT view. See “Commands available with the CREAGT view” on page 268.

Data collection is stopped and restarted during CMRSOPT reset.
Setting problem thresholds (CMRPRBT)

The problem threshold table, CMRPRBT, has entries that consist of a message number followed by the thresholds that trigger the display of that message.

The threshold table is used by the:

- Problem display and CREGPRB view
  Every time a user presses Enter in the Problem display or the CREGPRB view, the thresholds in the table are checked and messages that exceed the thresholds are displayed. The Problem display and CREGPRB view do not check thresholds for messages FT542 through FT547 and FT605 through FT614.

- MainView for CICS monitors, such as @PRBn and @TSKn
  The @PRBn monitors only use the thresholds in the problem threshold table. Other monitors, such as @TSKn, only use the severe threshold if the WLIM keyword is not specified. An active monitor that uses the table checks the thresholds for its associated message and issues the message to the journal log when a threshold is exceeded. For information about the relationship between messages, monitors, and WLIM, see "Keyword parameters" in the MainView for CICS Monitors Guide.
Default thresholds for the distributed problem threshold table are specified in BBSAMP member CMRUPRBT and in the distributed load module CMRPRBT. No modifications are required, but you should review the thresholds to see if they work for your environment. You can customize thresholds, but do not add messages to or delete messages from the table.

The CMRPRBT table can be maintained as an administrative function by using the CMRPRBT view, or it can be assembled and loaded based on the parameters that are provided by the CMRPRBT macro. All features that are provided by the CMRPRBT macro are supported in the administrative views.

When CMRPRBT is maintained as an administrative function, data is stored in member CMRPRBxx of the BBIPARM data set and data is maintained by using the CMRPRBT view. To use the administrative function, you must first use the CMRPRBT view to add the first table to BBIPARM.

---

**Note**

BMC recommends that you use the CMRPRBT view to maintain the problem threshold table. For information about using CMRPRBT, see "Administrative views" in the *MainView for CICS Online Services Reference Manual*.

---

The message format in CMRPRBxx is:

message_ID=(information_threshold, warning_threshold, severe_threshold)

For example, the definition for message FT041 could be:

FT041=(32, 100, 200)

When a target CICS connects to the BBI-SS PAS, the CMRPRBT table is loaded into ECSA. MainView for CICS looks for CMRPRBT in the following sequence:

- attempts to locate a CMRPRBxx member in the BBI-SS PAS BBIPARM concatenation; if one is found, it is used
  - CMRPRB00 is the default member name. The BBCFG facility can be used to specify a different suffix, as described in the *MainView Customization Reference*.
- attempts to load module CMRPRBT (default location is in BBLINK)

### Modifying the CMRPRBT macro

All MainView for CICS problem messages are located in the BBMLIB data set with a separate member for each message.
The members are named by message number.

Only the messages in CMRPRBT can have their thresholds modified. MainView for CICS issues a number of problem messages that do not require thresholds (for example, **FT056 CURRENTLY AT ACTIVE MAXIMUM TASK CONDITION**). These messages are not listed in CMRPRBT. The CMRPRBT default thresholds are specified in BBSAMP member CMRUPRBT.

**To modify CMRPRBT**

1. Edit sample member CMRUPRBT in your BBSAMP data set to set message thresholds.

2. Edit the TARGET parameter to include the CICS regions whose messages you want to set thresholds for:

   ```
   TARGET=(name1, name2, ..., nameN, | *)
   ```

   Thresholds apply to the messages generated by the CICS regions specified with the TARGET parameter. An asterisk (*) indicates that message thresholds apply to all unspecified CICS regions.

   If a TARGET= parameter is not included in CMRPRBT, the first TARGET parameter is used to set thresholds for any unspecified regions. For example, if TARGET statements are in the following order, the thresholds for CICS1 and CICS2 apply to CICS3 because it is an unspecified region:

   ```
   CMRPRBT  TARGET=(CICS1,CICS2),
   FT033=(200,300,400),
   FT034=(10,40,60),
   ....
   CMRPRBT  TARGET=(CICSTST,CICSPROD),
   FT033=(100,200,300),
   FT034=(80,100,200),
   ....
   ```

3. Edit the FT$n$ message statements to change the default message thresholds.

   The format of the FT$n$ statement is as follows:

   ```
   FT$n$=(aaaaaa, bbbbbb, cccccc)
   ```

   The variable, FT$n$, represents the message number. The values within parentheses, aaaaaa, bbbbbb, and cccccc, represent thresholds as follows:

   - **aaaaaa** represents an informational message threshold.
   - **bbbbb** represents a warning message threshold.
   - **ccccc** represents a severe message threshold.
Threshold values for count messages are specified as integers. The following example demonstrates how message thresholds are specified for counts:

\[ \text{FT041} = (32, 100, 200) \]

- An informational message is issued when the task uses more than 32 KB of DSA.
- A warning message is issued when the task uses more than 100 KB of DSA.
- A severe message is issued when the task uses more than 200 KB of DSA.

Percent and time thresholds are expressed as integers without a decimal point. A zero is added as the low-order number. The following example demonstrates how message thresholds are specified for percent thresholds:

\[ \text{FT046} = (500, 750, 900) \]

- An informational message is issued when CICS uses more than 50.0% of CPU resources.
- A warning message is issued when CICS uses more than 75.0% of the CPU.
- A severe message is issued when CICS uses more than 90.0% of the CPU.

4 (optional) Decrease the CPU utilization of the CREGION view and the interval recorder by limiting the scope of region problem count collection.

To stop CREGION and interval recorder collection of problem counts for storage violations (FT066 and FT088), add an 'N' as the fourth parameter to their Problem Threshold Table entries for the FT066 message. An example of the MACRO parameter is as follows:

\[ \text{FT066} = (0, 10, 20, N), \text{ STORAGE VIOLATIONS X} \]

Similarly, to stop collection of problem counts for program loads (FT099), add an 'N' as the fourth parameter to the entries for the FT099 message. An example of the MACRO parameter is as follows:

\[ \text{FT099} = (25, 50, 100, N), \text{ TIMES PROGRAM Fetched X} \]

5 (optional) Change the default color for a type of message by placing the name of a color after the equal sign in the following statements:

\[ \text{SEVERE=RED,} \\
\text{WARN=BLUE,} \\
\text{INFO=GREEN,} \]

MainView for CICS supports the following colors:
6 \textit{(optional)} Exclude messages from being displayed with the PROBLEM service if you think they represent insignificant events:

a Locate the following statements:
\begin{verbatim}
XTRAN=(' ',' ',' ',' ',' ',' ',' ',' ',' ',' ')
XFILE=(' ',' ',' ',' ',' ',' ',' ',' ',' ',' ')
\end{verbatim}

The XTRAN and XFILE statements specify which transactions or CICS files should be excluded from appearing with the PROBLEM service, even if their thresholds are exceeded.

b Enter the names of up to 10 transactions or files within the parentheses of the appropriate XTRAN or XFILE statement, and include single quotation marks around each transaction or file name and separate them with a comma.

Both statements support the "+" and "*" wildcards (for example, 'file\_name1','file\_name*').

7 Use sample JCL in BBSAMP member CMRASM to assemble and link the CMRUPRBT source module:

a Copy member CMRASM from BBSAMP to the UBBSAMP data set. (Make sure the new member name conforms to your site's naming conventions.)

b Read the comments in the member that was created in the previous step, and then modify the JCL accordingly.

Keep the following issues in mind:

- CMRPRBT must be assembled and linked into BBLINK or a user-authorized load library that is concatenated in front of BBLINK. A user-authorized load library is recommended so that future maintenance does not override your site's customizations.

- CMRPRBT must be linked reentrant.

- You must include the BBSAMP data set in the assembler SYSLIB concatenation.

c Submit the JCL, and check the job output to make sure all steps are completed with a return code of 0.
Note
S106 abends can occur if the newly link-edited module causes the BBLINK data set to enter secondary extents and the active BBI-SS PAS then attempts to load the module. If this situation happens, stop the BBI-SS PAS and restart it.

Resetting the CMRPRBRT macro

The CMRPRBRT table can be reset dynamically with the RESET command, or with the PRL action command in the CREGAGT view.

See “Commands available with the CREGAGT view” on page 268.

The following command resets monitoring options for all CICS regions:

`.RESET PARM CMRPRBRT`

The following command resets monitoring options that apply to a specific CICS region:

`.RESET PARM CMRPRBRT cicsName`

Refer to the MainView Administration Guide for a description of the RESET control command and supported parameters.

Implementing the task kill function

MainView for CICS provides an optional task kill function that operates within a target CICS region. The task kill function uses a set of user-defined thresholds and can terminate tasks that exceed those thresholds.

Task kill modes

The Task Kill function can run in one of two modes:

- Warn-only mode
- Kill mode
Warn-only mode

In warn-only mode, MainView for CICS uses the kill thresholds to determine if a message is issued when any of the kill thresholds are exceeded and the task is not killed. Once a resource threshold is exceeded in a task, MainView for CICS does not monitor that resource again in that task. You can turn on warn mode by specifying YES on the WARN option, which also requires that you specify CON, LOG, or BOTH on the Write Message option (WTO). You can use this mode to test and determine if the kill thresholds are set appropriately.

Kill mode

In kill mode, the default, you can define a set of kill thresholds with a set of optional warning thresholds. To use the warning thresholds, you must turn on the Warning Thresh option by specifying YES in the administration dialog. You must also specify CON, LOG, or BOTH on the Write Message option (WTO). MainView for CICS checks the warning thresholds first and issues a warning message when the resource warning threshold is exceeded. After the warning threshold for a resource is exceeded, MainView for CICS uses the kill threshold for that resource to determine if the task should be abended at the next time it gets control.

**Note**

If you do not turn on the warning threshold option or, if you turn on the option but do not set the warning thresholds, MainView for CICS uses the values set by the kill thresholds to stop tasks without any warning.

Thresholds and options for task kill function

Thresholds can be set for the following specifications:

- Storage total, ABOVE, BELOW
- CPU time
- File I/O
- DB2 calls
- DBCTL calls
- other DB calls
- temporary storage access (TSQ)
- transient data access (TDQ)
- MQS calls
- elapsed time

Tasks are terminated with default or user-defined task specific abend codes. You can also set options to control the following functions:
Sending a WTO message when a task exceeded its thresholds

Bypassing certain tasks in specific conditions

Creating a dump when the task is abended because one of the resource thresholds has been exceeded

Run in Warn only mode

Run in Kill mode with or without warning thresholds

You can specify the task kill thresholds, options, and abend codes from the CMRTTHTHR view, which stores these parameters in the BBPARM member CMRTTHnn for the BBI-SS PAS. During the initialization of the task kill exit, the BBI-SS PAS reads the settings in the CMRTTHnn member and the parameters are passed, using the EXCI facility, to the task kill exit in CICS where they are stored in ECSA storage.

Note

CMRTTH00 is the default member name. You can use the BBCFG facility to set a different suffix, as described in the MainView Customization Reference.

For more information about using the CMRTTHTHR view, see the MainView for CICS Online Services Reference Manual and the online Help for CMRTTHTHR.

Task kill limitations and considerations

This topic describes task kill limitations and considerations.

The task kill function gets information from the following sources:

- the CICS monitor record (for local file, elapsed time, and CPU time)
- the MainView for CICS task's TIE data area (for DB2, DBCTL, MQS, FILE, Temporary Storage, Transient Data, and other DB calls)
- the CICS storage manager control blocks for all storage-related thresholds

Therefore, if the MainView for CICS Extractor is not running or if the CICS Monitoring Function (CMP) is not active, the effects of the task kill function are limited.

The task kill function gets control at two CICS global exit points: XEIIN and XRMIOUT. The task has to issue an EXEC CICS or return after an RMI call in order for the task kill function to get control and check the thresholds that are specified in CMRTTHTHR. Because of this limitation, a task might not be killed at the
time the threshold is exceeded; it will be killed the next time the kill exit gets control and checks the kill thresholds.

- The programs that provide the task kill function are subject to discard and create processing that is employed by the TERM and INIT functions for auto maintenance. The task kill function is turned off (if it is on) when a TERM function is issued and restarted when the INIT function is issued.

- The file call threshold includes remote files if the extractor is active. If the extractor is not active and MN=ON, the kill exit counts only local file access because it uses the CICS task monitoring area (TMA) field, which is not updated for remote file access.

- When the extractor is active and TSQ and TDQ monitoring are active, the threshold includes local and remote access; otherwise, when only the extractor is active, only local access is included.

- The Task Kill Exit is not supported in a CPSM CMAS or WUI region. Attempts to start the Task Kill Exit will not be processed.

- The Option Warn and Warning Thresh functions are mutually exclusive. The kill thresholds are always used in WARN-only mode.

- MainView for CICS uses the warning thresholds only when the option Warning Thresh is set to YES. In addition, it is used only when you also specify a kill threshold for the resource.

- When you set the Warning Thresh to YES, when a warning threshold for a resource is exceeded, MainView for CICS uses the kill threshold of the same resource the next time it gets control.

   - This means before a task is killed by the exit it might have multiple warning messages such as one for each monitored resource or just one warning message for the resource that has exceeded the Kill threshold.

   - There is also a possibility MainView for CICS kills the task at the next CICS EXEC right after issuing a warning for the resource (depending on the value of the resource).

   - It could also delay killing a task depending on when the exit gets control. The exit does not get control until the task schedules an EXEC CICS.

   The resource might exceed both warning and kill thresholds before the exit gets control but the warning threshold is checked first and the kill threshold is not checked until the next EXEC CICS, therefore delaying the kill.
Transactions that are excluded from task kill processing

The task kill function will not kill the following tasks and transactions:

- a system task
- a transaction that is currently executing a DFH* program
- a transaction that is currently executing one of the EZACIC* programs (CICS Socket Exits)
- a transaction that is currently executing a EYU* (IBM CPSM) program
- a transaction that is currently executing a EQA* (IBM DEBUG TOOL) program
- a transaction that has already abended
- a transaction that is abending or dumping
- BMC transactions (JNL2, FST2, BMCE, FIC2, BCRT, SMN2 and FCD2)

The task kill function will not kill a task for certain thresholds or options if the MainView for CICS Extractor is not running or if the CICS Monitoring Function (CMP) is not active, as described earlier and in the online Help for the CMRTTHR views and related dialogs.

You can also exclude other transactions from resource management by using the XDEBUG and the Kill Mirror options under CMRTTHR:

- Use the XDEBUG option to exclude transactions that are started under CEDF, CEDX, or the IBM Debug Tool.
  
  If the IBM Debug tool starts debugging an active task, that task will not be under resource limitation, even if it was under resource limitation previously.

  Excluding transactions that are being debugged by the IBM Debug Tool involves extra overhead. Unless you are using the IBM Debug Tool, you should not specify XDEBUG=IDBG or XDEBUG=ALL

- The Kill Mirror option excludes transactions that are running the DFHMIRS* program when the kill exit gets control.

  If the DFHMIRS* program does not schedule any EXEC CICS, it will not be killed by the exit, no matter what is specified in this option.

For more information about these options, see the online Help.
Starting the task kill function

The task kill function can be started as follows:

- automatically at PLTP startup by using the CMRSOPT parameter TASKKILL=YES.

  This parameter is checked by MainView for CICS during CICS PLTP processing or when the first INIT request is processed. The parameter is ignored on all subsequent INIT requests. The TASKKILL=NO | YES parameter is described in “TASKKILL=NO | YES” on page 135.

  **Note**  
  BMC recommends that you use the TASKKILL=YES parameter in CMRSOPT to start the task kill function during PLT if MainView for CICS is in the PLT.

- manually by using the FST2 KON transaction command, as described in “Starting the task kill function” on page 235

- by using the KON line command from the CREGAGT view, as described in “CREGAGT view commands” on page 267

Stopping the task kill function

The task kill function is inactive by default. The task kill function can be stopped as follows:

- manually by using the FST2 KOFF transaction command, as described in “Stopping the task kill function” on page 236

- by using the KOF line command from the CREGAGT view, as described in Commands available with the CREGAGT view on page 268

Setting the task kill parameters

Use the CMRTTHHR family of views to set the task kill abend codes, thresholds, and options.

For a detailed description of these parameters, see the online Help for the CMRTTHHR views and related dialogs. For detailed information about the CMRTTHHR view, see the MainView for CICS Online Services Reference Guide.
Reloading changed task kill parameters

Changes made to the task kill parameters in the CMRTTHHR view do not take effect until the CMRTTHnn table is reloaded.

The CMRTTHnn table can be reloaded as follows:

- by using the FST2 KRLD transaction command as described in “Reloading the task kill function” on page 236
- by using the KRL command from the CREGAGT view, as described in “Commands available with the CREGAGT view” on page 268

Note
Reset thresholds do not apply to active tasks.

Implementing program timing

The Program Timing function tracks CPU use by each program that executes for a single task.

A call count and an accumulation of the CPU time for each program is maintained. The CPU time includes the processor time for all TCB modes for which the task was dispatched (RLS SRB time is not included). The call count is the number of times the program received control through a LINK or an XCTL. The count is not incremented when a program returns to a calling program or to CICS.

Program Timing accumulates the CPU time for all programs that are accessed by a transaction only when the Applications Programming Interface (API) LINK, XCTL and RETURN commands are used to pass or transfer control. CPU time used by CICS on behalf of a program’s processing request (like file I/O, database accesses, storage gets/frees, and so on) is attributed to the program making the request. CPU time that is used by CICS to process a LINK or XCTL request is attributed to the program intended to receive control. A couple examples of CPU accumulation follow:

- When program A links to program B, CICS accesses the PPT for the program’s processing parameters. CICS may have to install a program definition if one does not exist and the Program Auto-Install Facility is active. CICS may also have to LOAD the program into storage before actually giving control to it. The CPU time that is used to perform these tasks is accumulated to program B.

- When program A links to program B, a load failure might occur if B is not available. The minute the link to B is issued, the CPU clock for program B is...
started. In this case, program B will be reported with a use count and CPU time even though program B never received control.

Program timing information is maintained in data areas that are established by the MainView for CICS data extractor, so the extractor must be active. See “Using CMRDETL data sets” on page 77 for information about the extractor.

Program Timing entries are identified by using the exclamation point (!) as the type identifier. If you use ! as an identifier for a database through the MainView for CICS CMRCUEX user exit, change it to another value. See “User exit interface (CMRCUEX)” on page 255 for more information.

Activating program timing

To activate the program timing function at MainView for CICS startup in the CICS target, display the CMRSOPT administration view and set the PGM CPU Timing parameter field to **YES**.

This parameter is checked during PLT initialization when MainView for CICS is in the PLT, or on the first INIT request and is ignored on subsequent INIT requests. The parameter is not available in the CMRSOPT macro.

The Program Timing function can be turned on or off by using the CREGAGT view or the FST2 transaction. The CREGAGT view also provides status information on the Program Timing function.

The line commands for the CREGAGT view are as follows:

- **PGY** turns on the Program Timing function
- **PGN** turns off the Program Timing function

**Figure 16: CREGAGT view**

```
12APR2005  07:56:58 ------ MainView WINDOW INTERFACE (V4.2.04) ----------------
COMMAND  ===>  SCROLL ===> PAGE
CURR WIN ===> 1        ALT WIN ===>
>W1 =CREGAGT==========(ALL======*=======)12APR2005==07:56:55====MVCICS===D===17
CMD Target   Intvl SMF  PAS  Region  Extract TskKill AO CICS Enrgzer CMRDTL  PG
--- Name     Time- ID   ID   Status  Status  Status  Status  Status  Write  Tim
BCVCD410 07:56 SJSC DVPC INACTIV N/A     N/A     N/A     N/A    N/A
PGY BCVCPC64E 07:56 SJSE PP5B CONNECT ACTIVE ACTIVE UNAVAIL UNAVAIL ON     ON
BCVCDC30 07:56 SJSC DVPC CONNECT ACTIVE Active UNAVAIL UNAVAIL ON     ON
BCVCDO40 07:56 SJSC DVPC CONNECT ACTIVE ACTIVE UNAVAIL UNAVAIL ON     ON
```

The FST2 transaction commands are as follows:

- **PGY** or **PGON** turns on the Program Timing function
- **PGN** or **PGOFF** turns off the function
See “FST2 transaction” on page 263 for information about the FST2 transaction.

Program Timing information for active tasks can be viewed with the TASKPGM view; history data can be viewed by using the CHISTDIO and CHPROG views. The history views also indicate whether the function was stopped or interrupted (stopped and restarted) while a task was active.

Figure 17: TASKPGM view—active task information

Figure 18: CHISTDIO view
A Yes value in the **Pgm. Timing Incmplt** field indicates program timing was stopped (and possibly restarted) while that particular task was being timed.

**Figure 19: CHPROG view--historical data**

<table>
<thead>
<tr>
<th>Task Num</th>
<th>CICS</th>
<th>Lcl Task</th>
<th>Lcl Task Tran Program</th>
<th>Count</th>
<th>Time</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>196</td>
<td>BCVCD630</td>
<td>12APR2005</td>
<td>06:46:00 FIC2 CMRROLL3</td>
<td>1</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td>196</td>
<td>BCVCD630</td>
<td>12APR2005</td>
<td>06:47:00 FIC2 CMRSLOG3</td>
<td>1</td>
<td>0.0008</td>
<td>0.0008</td>
</tr>
<tr>
<td>197</td>
<td>BCVCD630</td>
<td>12APR2005</td>
<td>06:47:00 FIC2 CMRSLOG3</td>
<td>1</td>
<td>0.0009</td>
<td>0.0009</td>
</tr>
<tr>
<td>198</td>
<td>BCVCD630</td>
<td>12APR2005</td>
<td>06:48:00 FIC2 CMRROLL3</td>
<td>1</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td>198</td>
<td>BCVCD630</td>
<td>12APR2005</td>
<td>06:48:00 FIC2 CMRSLOG3</td>
<td>1</td>
<td>0.0008</td>
<td>0.0008</td>
</tr>
<tr>
<td>199</td>
<td>BCVCD630</td>
<td>12APR2005</td>
<td>06:49:00 FIC2 CMRROLL3</td>
<td>1</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td>199</td>
<td>BCVCD630</td>
<td>12APR2005</td>
<td>06:49:00 FIC2 CMRSLOG3</td>
<td>1</td>
<td>0.0079</td>
<td>0.0079</td>
</tr>
<tr>
<td>200</td>
<td>BCVCD630</td>
<td>12APR2005</td>
<td>06:50:00 FIC2 CMRROLL3</td>
<td>1</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td>200</td>
<td>BCVCD630</td>
<td>12APR2005</td>
<td>06:50:00 FIC2 CMRSLOG3</td>
<td>1</td>
<td>0.0009</td>
<td>0.0009</td>
</tr>
<tr>
<td>201</td>
<td>BCVCD630</td>
<td>12APR2005</td>
<td>06:51:00 FIC2 CMRROLL3</td>
<td>1</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td>201</td>
<td>BCVCD630</td>
<td>12APR2005</td>
<td>06:51:00 FIC2 CMRSLOG3</td>
<td>1</td>
<td>0.0009</td>
<td>0.0009</td>
</tr>
<tr>
<td>202</td>
<td>BCVCD630</td>
<td>12APR2005</td>
<td>06:52:00 FIC2 CMRROLL3</td>
<td>1</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>202</td>
<td>BCVCD630</td>
<td>12APR2005</td>
<td>06:52:00 FIC2 CMRSLOG3</td>
<td>1</td>
<td>0.0009</td>
<td>0.0009</td>
</tr>
</tbody>
</table>

See the **MainView for CICS Online Services Reference Manual** for information about the TASKPGM view.

**Notes on Program Timing operations**

Program Timing does not require file expansion to be active.

See “FILEXPND=NO | YES” on page 131).  

- If Program Timing is active when the extractor shuts down, it will be restarted when the extractor restarts.

- If Program Timing is active when MainView for CICS terminates within a target CICS, it restarts when MainView for CICS initializes within that same target CICS.

- If Program Timing is active when MainView for CICS terminates as a result of the BBI-SS PAS terminating, it restarts when MainView for CICS initializes as a result of the BBI-SS PAS restarting.

- If Program Timing is active when the target CICS shuts down, it will not be started when the CICS restarts.

- Tasks already active when Program Timing is started are not timed.
Monitoring sessions and connections

The session/connection feature collects session information (functions and function timing) for DTP (Distributed Transaction Processing) only for all session activity across LU 6.1, APPC and MRO connections.

Additionally, it collects connection usage information by activity type (see below) for all activity across LU 6.1, APPC, MRO, and TCP/IP connections.

The following table shows the data that is collected for each activity type and connection type:

<table>
<thead>
<tr>
<th>Activity Type</th>
<th>Connection Type</th>
<th>Connection Data?</th>
<th>SessionData?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LU61</td>
<td>MRO</td>
<td>APPC</td>
</tr>
<tr>
<td>DBCTL</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>DPL</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>DTP</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>File</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>IC</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>TD</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>TS</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

Data is maintained in the same resource entries in CMRDETL that are used for file activity, database activity, program timing, and so on, and therefore has an effect on the MAXFILE parameter.

The left parenthesis ( is used as the one-byte character identification of the session entries.

The right parenthesis ) is used as the one-byte character identification of the connection entries.

Session monitoring

This topic provides session monitoring information.

- CHISTDIO displays the number of sessions, and total counts and times for a task

- CHSESS provides general history information about the usage of individual sessions by a transaction
CHISTDN shows the session activity counts and elapsed time by command for the entire task

CHISTDN1-5 shows the session activity counts and elapsed time by command for an individual session

TASKSES provides count and timing information for session activity within a conversation to a connected CICS system

For more information about these views, see the *MainView for CICS Online Services Reference Manual*.

### Connection monitoring

Connection monitoring records activity type data.

The types of activities monitored across a connection are:

- **distributed transaction processing (DTP)**
  
  DTP refers to a transaction on one CICS talking to a transaction on another CICS. The XEIIN and XEIOUT exits are used for DTP data collection.

- **distributed program link (DPL)**
  
  DPL refers to a program running on one CICS that does an EXEC CICS LINK to a program defined as remote, or the EXEC command supplies a SYSID. Connection timing for CICS TS 3.1 and earlier uses the XEIIN and XEIOUT exit points. CICS TS 3.2 and later uses the XPCREQC exit point.

- **transient data (TD) request function shipping**
  
  TD queue requests can be function shipped to another CICS if the TD queue is defined as remote, or a SYSID is specified on the EXEC CICS statement. TD request processing is done with a CSMx transaction on the remote system. TD function shipping data is trapped with the XEIIN and the XTDEREQC exit points.

- **temporary storage (TS) request function shipping**
  
  TS queue requests can be function shipped to another CICS if the TS queue is defined as remote (or indirect), or a SYSID is specified on the EXEC CICS statement. TS request processing is done with a CSMx transaction on the remote system. TS function shipping data is trapped via the XEIIN and the XTSEREQC exit points.

- **file request function shipping**
  
  File requests can be function shipped to another CICS if the file is defined as remote, or a SYSID is specified on the EXEC CICS statement. File request processing is done with a CSMx transaction on the remote system.
IBM IMS/DBCTL request function shipping

Database access is done with the EXEC DLI or CALLDLI commands specifying a PSB. A DFHPSBxx table specifies the SYSID where the PSB is located. The shipped request runs under a CSMx transaction on the remote side. The XDLIPRE and XDLIPOST global exit points are used to gather data.

To gather connection timing for remote DBCTL requests, PSB=YES must be specified in the CMRSOPT.

interval control request function shipping

Interval control requests can be function shipped to another CICS if a SYSID is specified on the EXEC CICS statement. Interval control request processing is done with a CSMx transaction on the remote system. Interval control function shipping data is trapped with the XEIN and the XICEREQC exit points.

For each connection used by a task, the total time and count for all activity is accumulated, along with each individual activity type’s time and count. The time represents the elapsed time from when a request leaves the sending CICS until the request returns.

Connection monitoring views

Like session monitoring, connection monitoring is controlled with the FST2 transaction or the CREGAGT view.

For a description, see “Starting and stopping session/connection monitoring” on page 168.

Connection data is displayed in the following views:

- TASKCON is the primary connection usage view. It provides usage information for all connections used by all tasks, as well as hyperlinks to connection related detail views.
  CHCONN provides similar information for historical data.

- TASKCOND displays detail information about a specific CICS connection for a specific CICS task.
  CHCONND provides similar information for historical data.

- TASKCONx displays the counts and times for a specific activity type. The type of data displayed depends on the function type selected:
  - TASKCONLtype DPL (PL)
  - TASKCONPtype DTP (TP)
  - TASKCONFtype FSFL (FL)
— TASKCON $/type FSTD (TD)
— TASKCON $/type FSTS (TS)
— TASKCON $/type FSIC (IC)
— TASKCON $/type FSDC (DC)
— TASKCON $/type other (OT)

The CHCONN$ views provide similar information for historical data. For example, CHCONNL provides historical DPL data.

- Hyperlinks on the Connectn field of the CHISTDIO view display the CHISTDO view which provides detail connection information.

For more information about these views, see the MainView for CICS Online Services Reference Manual.

Starting and stopping session/connection monitoring

To activate the session/connection monitoring function at MainView for CICS startup in the CICS target, display the CMRSOPT administration view and set the SESSION Data parameter field to YES.

This parameter is checked during PLT initialization when MainView for CICS is in the PLT, or on the first INIT request and is ignored on subsequent INIT requests. The parameter is not available in the CMRSOPT macro.

Session/connection monitoring can also be controlled with the FST2 transaction or the CREGAGT view:

- CSON activates the session/connection data collection (FST2 only)
- CSOFF deactivates the session/connection data collection (FST2 only)
- CSY activates the session/connection data collection (FST2 and CREGAGT)
- CSN deactivates the session/connection data collection (FST2 and CREGAGT)

The CREGAGT view also provides status information about the session/connection monitoring function.
Monitoring transient data (TD) queues

The transient data feature collects function and function timing information for transient data (TD) queues that are accessed by a task.

MainView for CICS collects data individually by TD queue name. The monitored functions include reads, writes, and deletes.

MainView for CICS maintains the collected data in the same resource entries in CMRDETL that are used for file activity, database activity, program timing, and so on, and therefore has an effect on the MAXFILE parameter.

The one-byte forward slash (/) identifies the transient data entries.

The following views display TD information for a task:

<table>
<thead>
<tr>
<th>View name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTASKDIO</td>
<td>displays the number of TD queues and the total function count and total and average time</td>
</tr>
<tr>
<td>CHISTDIO</td>
<td></td>
</tr>
<tr>
<td>TASKTDQ</td>
<td>displays in tabular form the count and timing information for each TD queue</td>
</tr>
<tr>
<td>CHTDQ</td>
<td></td>
</tr>
<tr>
<td>TASKTDQD</td>
<td>displays detail function count and timing information for a single TD queue</td>
</tr>
<tr>
<td>CHTDQD</td>
<td></td>
</tr>
<tr>
<td>CTASKDJ</td>
<td>displays detail function and count information for all TD queues</td>
</tr>
<tr>
<td>CHISTDJ</td>
<td></td>
</tr>
<tr>
<td>CHISTDJ1 through 5</td>
<td></td>
</tr>
</tbody>
</table>

For more information about these views, see the MainView for CICS Online Services Reference Manual.

Starting and stopping transient data (TD) queue monitoring

To activate the transient data (TD) queue monitoring function at MainView for CICS start up in the CICS target, display the CMRSOPT administration view and set the Transient Data Queue parameter field to YES.

This parameter is checked during PLT initialization when MainView for CICS is in the PLT, or on the first INIT request; the parameter is ignored on subsequent INIT requests. This parameter is not available in the CMRSOPT macro.

You can also control transient data queue monitoring with the FST2 transaction or the CREGAGT view:
Monitoring temporary storage (TS) queues

The temporary storage feature collects function and function timing information for temporary storage (TS) queues that are accessed by a task.

MainView for CICS collects data individually by TS queue name. The monitored functions include reads, writes, and deletes.

MainView for CICS maintains the collected data in the same resource entries in CMRDETL that are used for file activity, database activity, program timing, and so on, and therefore has an effect on the MAXFILE parameter.

The one-byte backward slash (\) identifies the temporary storage entries.

The following views display TS information for a task:

<table>
<thead>
<tr>
<th>View name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTASKDIO</td>
<td>displays the number of TS queues and the total function count and total</td>
</tr>
<tr>
<td>CHISTDIO</td>
<td>and average time</td>
</tr>
<tr>
<td>TASKTSQ</td>
<td>displays in tabular form the count and timing information for each TS queue</td>
</tr>
<tr>
<td>CHTSQ</td>
<td></td>
</tr>
<tr>
<td>TASKTSQD</td>
<td>displays detail function count and timing information for a single TS queue</td>
</tr>
<tr>
<td>CHTSQD</td>
<td></td>
</tr>
<tr>
<td>CTASKDK</td>
<td>displays detail function and count information for all TS queues</td>
</tr>
<tr>
<td>CHISTDK</td>
<td></td>
</tr>
<tr>
<td>CHISTDK1 through 5</td>
<td></td>
</tr>
</tbody>
</table>

For more information about these views, see the MainView for CICS Online Services Reference Manual.
Starting and stopping temporary storage (TS) queue monitoring

To activate the temporary storage (TS) queue monitoring function at MainView for CICS startup in the CICS target, display the CMRSOPT administration view and set the **Temporary Storage Queue parameter** field to **YES**.

Because temporary storage queue names can be as long as 16 bytes, an offset parameter is also available from the CMRSOPT administration view. This parameter designates the offset into the 16-byte queue name to where the 8-byte key is extracted. The 8-byte key designates the queue name from which MainView for CICS collects the temporary storage data. Valid offset values are from 0 through 9; 0 is the default.

These parameters are checked during PLT initialization when MainView for CICS is in the PLT, or on the first INIT request and are ignored on subsequent INIT requests. These parameters are not available in the CMRSOPT macro.

You can control temporary storage monitoring with the FST2 transaction or the CREGAGT view:

- **TSON**: activate temporary storage queue collection (FST2 only)
- **TSOF**: deactivate temporary storage queue collection (FST2 only)
- **TSY**: activate temporary storage queue collection (FST2 and CREGAGT)
- **TSN**: deactivate temporary storage queue collection (FST2 and CREGAGT)

The CREGAGT view also provides status information about the temporary storage queue monitoring function.

Monitoring the CICS Transaction Gateway

MainView for CICS supports the CICS Transaction Gateway for z/OS version 7.1 and later in remote mode.

To use MainView for CICS to monitor the CICS Transaction Gateway (CTG), you must modify both the MainView for CICS and CTG environments.

Modify the MainView for CICS environment as described in the following sections:

- “Update the PAS startup JCL” on page 172
- “Set the DYNTGT parameter in member BBISSPnn” on page 173
“Set the CTG monitoring parameters in member CMRBEXnn” on page 173

Modify the CTG environment as described in the following sections:

“Update the CTG environment file” on page 176

“Update the ctg.ini configuration file” on page 177

MainView for CICS modifications

Make the following modifications to your MainView for CICS environment.

Update the PAS startup JCL

All of the data sets mentioned in this section must be APF authorized.

Update the PAS startup JCL to include the following libraries in the STEPLIB/JOBLIB concatenation.

- CICS Transaction Gateway Statistics API Runtime library
  — provided by IBM as an SMP/E PDSE during installation of CTG
  — DSN format: hlq.SCTGDLL

- IBM C Language Environment Runtime Library
  — provided by IBM as an SMP/E PDSE during installation of LE
  — DSN format: hlq.SCEERUN2

- IBM TCP/IP network management interface program EZBNMIFR
  — provided by IBM as an SMP/E PDSE during installation of TCP/IP
  — DSN format: hlq.CSSLIB

If the above data sets are in the LNKLST, you do not need to add them to the PAS JCL.

Update the PAS startup JCL to include the following library in the STEPLIB, JOBLIB, or BBLOAD that has BBLINK in the concatenation.

- MainView for CICS PDSE that contains the interface to the CTG Statistics API
— provided by BMC as an SMP/E PDSE during installation of MainView for CICS

— DSN format: hlq.TOSZLINK

If BBLINK is in the LNKLST and not in the PAS startup JCL, TOSZLINK must also be in LNKLST. If a BBLOAD DD is in the PAS startup JCL, BBLINK and TOSZLINK must also be in the BBLOAD concatenation.

### Set the DYNTGT parameter in member BBISSPnn

If ACONCICS=NO (the default) is specified in the BBISSPnn member in UBBPARM, specify DYNTGT=YES for all BBI-SS PASs that are monitoring:

- CICS Transaction Gateway
- any CICS region that CTG is using

### Set the CTG monitoring parameters in member CMRBEXnn

Update your CMRBEXnn member with the global monitoring parameter CTGMON and target monitoring parameter CTGNAME.

The CTGMON parameter controls whether CTG monitoring is on or off.

CTGMON=YES, MAXGATE=16, DSPLIM=2048

CTGMON=NO | YES

Specifies whether CTG monitoring should be active. Default is NO.

The CTGMON parameter options are described below.

**MAXGATE=nnn**

Indicates the maximum number of gateways the PAS will manage. The value must be in range of 1-100. Default is 20.

**DSPLIM=nnnn**

Indicates the size in megabytes of the home data space used by the PAS to manage communication with the gateway. The value must be in range of 100-2048. Default is 2048.
There can only be one CTGMON parameter in CMRBEXnn (any additional CTGMON parameters are ignored).

See “Starting and resetting CTG monitoring” on page 180 for information about resetting parameters.

Define the CTGs to be monitored with CTNAME parameter and its options.

**CTNAME = ccccccccc**

The CTNAME parameter identifies the gateway. It provides the JOBNAME of a CICS Transaction Gateway for the z/OS daemon. The daemon must run on the same z/OS image as the BBI-SS PAS.

**Example**

```
CTNAME=BCVDWBTG,PORT=24010,SSID=WBPC,DSPSIZE=1,IPSTACK=DC$TCPC
```

CTNAME=cccccccc can be any 8-character name when CTGHOST is specified, but ccccccccc must be unique within the CMRBEXnn member and it cannot be an existing job name.

The CTNAME parameter options are described below.

**CTGHOST=hostname**

This keyword lets MainView for CICS collect gateway statistics from daemons that run on platforms other than z/OS.

Provides the host name or IP address of the CTG daemon. The host name can be up to 255 characters and span multiple lines. If the name spans multiple lines, the first part of the name must extend into column 80 and the continuation must begin in column one (blank spaces are treated as delimiters).

**Note**

For gateway daemons on platforms other than z/OS, MainView for CICS collects statistics data but not task performance data or CTG history data. Therefore, the following views, which display task performance data and CTG history data, will not contain meaningful data for those gateways:

- CTGSERV, CTGSRV*, CTGCICST, CTGCCS*, CTGHIST*, CTGHIC*  

The following views will display meaningful data for all gateways:

- CTGSERV, CTGSERVZ, CTGSERVQ, CTGSERVD, CTGSRVL, CTGCICS,  
- CTGCICSZ, CTGCICSQ, CTGCICSD, CTGCICSD CTGLSRV, CTGLSRVZ,  
- CTGLSRVQ, CTGLSRV
**PORT=nnnnn**

Provides the port number of the gateway for the CICS Transaction Gateway Statistics API. The values are in range of 1-65535. Default is 2980.

Specify the same statistics port number in the gateway `ctg.ini` file, as described in “Update the ctg.ini configuration file” on page 177.

The port number must be unique.

**SSID=cccc**

Provides the name of the BBI-SS PAS that will manage the gateway.

**DSPSIZE=nnn**

Provides the size in megabytes of the local data space used to hold transaction performance data from the gateway. The value must be in the range of 1-100. Default is 1.

All data in the data space is lost when:

- the BBI-SS PAS is recycled
- CTG monitoring is turned off
- the monitored gateway is recycled

You have the option to store CTG history data in a VSAM data set and have it managed in the same manner as CMRDETL data by using the CMRDETL view. MainView for CICS looks for a CTG history data set definition in BBPARM member `CMRDTLnn` (defined in the BBI-SS PAS JCL) which matches the CTG target. If a definition is found, it is used; if no definition is found, MainView for CICS creates a data space by using the DSPSIZE value. For information about creating a CTG history data set definition, see “Using the CMRDETL view” on page 80.
Note

- Setting the DSPSIZE value too high might have an adverse effect on performance.

- CTG transaction performance data is completely separate from CMRDETL data, and must reside in its own VSAM data set. For information about displaying CTG transaction performance data, see "CICS Transaction Gateway" in the MainView for CICS Online Services Reference Manual. Also see the MainView for CICS PERFORMANCE REPORTER User Guide and MainView for CICS PERFORMANCE REPORTER Data Reference for more information about batch reporting.

**IPSTACK=xxxxxxxx**

Enables a single BBI-SS PAS to manage CICS Transaction Gateways that use different IP stacks on the same LPAR. xxxxxxxx is the IP stack name that is specified in the _BPXK_SETIBMOPT_TRANSPORT environment variable in the CTG environment file. Up to sixteen CTG jobs (as specified on the CTGNAME parameter) can have unique IP stacks. If more than sixteen unique stacks are specified, the next connection request with a unique IPSTACK value will fail.

IPSTACK is supported only if the ctg.ini file contains the statistics API protocol handler definition (protocol@statsapi) that was introduced in CTG version 7.2, rather than the statsport parameter. If an invalid combination of IPSTACK and statsport parameters is specified, the gateway connection fails, MainView for CICS issues the message FT686E, and does not attempt to connect again until the gateway or the BBI-SS PAS is recycled.

**CICS Transaction Gateway modifications**

Make the following modifications to your CTG environment.

**Update the CTG environment file**

The CICS Transaction Gateway must point to the UNIX file system where the MainView for CICS monitoring code resides.

The file system:

- is provided by BMC as an SMP/E file system during installation of MainView for CICS in a data set named hlq.BDRHFS
should be treated as a load library and be shared only by systems that receive
MainView for CICS maintenance at the same time

should not be combined with any other data set, including any other BMC
product file

has to be mounted and accessible to the CICS Transaction Gateway address space

Update the CTG environment file to include the path to hlq.BDRHFS, which
contains the code that implements the MainView for CICS CTG request exit.

The environment file is a PDS member pointed to by the setting of the STDENV
DD statement in the startup JCL for the CTG daemon.

Update the CLASSPATH to include the path to the UNIX file (BDRHFS) that
contains the MainView for CICS Oracle Java exit.

Format: CLASSPATH=classpath for mvcics.jar; classpath2...

Update the LIBPATH to include the path to the UNIX file (BDRHFS) that contains
the MainView for CICS C++ code.

Format: LIBPATH=libpath for mvcics libraries; libpath2...

The UNIX file must be mounted and accessible to the gateway.

In the following example, notice that the CLASSPATH and LIBPATH statements
refer to the same directory:

CLASSPATH=/var/BCVM/CMR62/bmc/bdr/v62/MVCICSctg.jar:
/usr/lpp/ctg/classes/ctgclient.jar:
/usr/lpp/ctg/classes/ctgserver.jar:
/usr/lpp/ctg/classes/ctgsamples.jar
LIBPATH=/var/BCVM/CMR62/bmc/bdr/v62/

**Update the ctg.ini configuration file**

The ctg.ini file is an HFS file defined to the gateway by the CICSCLI= parameter in
the environment file.

Update the ctg.ini file as follows:

- Make sure that APPLID=xxxxxxxx is specified in the PRODUCT section so that
each CTG has an unique APPLID.

- Add the name of the MainView for CICS Java exit by using the requestexits
parameter in the GATEWAY section.

  The format is requestexits=MVCICSctgMonitor

Chapter 7  Optional implementation procedures  177
Make sure the port number specified in UBBPARM member CMRBEXnn matches the port number specified in one of the following parameters:

— statsport= parameter for CTG version 7.1 or later

— port= parameter on the statistics API protocol handler definition (protocol@statsapi) for CTG version 7.2 or later (required if IPSTACK is specified in CMRBEXnn)

Figure 20 on page 178 provides an example of the GATEWAY section in a ctg.ini file for CTG version 7.n.

**Figure 20: Sample GATEWAY section in a ctg.ini file for CTG version 7.n**

```
SECTION GATEWAY
  initconnect=1
  maxconnect=100
  initworker=1
  nonames=on
  statsport=22012
  statint=030000
  stateod=000000
  protocol@tcp.handler=com.ibm.ctg.server.TCPHandler
  protocol@tcp.parameters=port=22002;\n    connecttimeout=2000;\n    idletimeout=600000;\n    pingfrequency=60000
  workertimeout=10000
  closetimeout=10000
  log@info.dest=console
  log@error.dest=console
  requestexits=MVCICSctgMonitor
ENDSECTION
```

Figure 21 on page 178 provides an example of the GATEWAY section in a ctg.ini file for CTG version 8.n.

**Figure 21: Sample GATEWAY section in a ctg.ini file for CTG version 8.n**

```
SECTION GATEWAY
  initconnect=1
  maxconnect=100
  initworker=1
  protocol@statapi.handler=com.ibm.ctg.server.RestrictedTCPHandler
  protocol@statapi.parameters=connecttimeout=2000;port=2980;bind=:maxconn=5;
    statint=030000
    stateod=000000
  protocol@tcp.handler=com.ibm.ctg.server.TCPHandler
  protocol@tcp.parameters=connecttimeout=2000;port=2006;idletimeout=600000;
    pingfrequency=60000
    workertimeout=10000
    closetimeout=10000
    log@info.dest=console
    log@error.dest=console
    requestexits=MVCICSctgMonitor
ENDSECTION
```

Additional information about setting up the CTG server can be found in the *CICS Transaction Gateway: z/OS Administration manual.*
Redirect gateway request exit messages (optional)

By default, the messages that are issued by the MVCICS gateway request exit are written to standard output (STDOUT), along with messages from the CTG daemon.

For CTG version 7.1 and later, you can redirect the messages to a Hierarchical File System (HFS), a data set, or a DD name defined in the CTG startup procedure. To redirect the messages, specify the following startup option in the CTG startup procedure:

-j-DMVCICSctg.out=outputDestination

_outputDestination_can be one of the following output destinations:

<table>
<thead>
<tr>
<th>Output destination</th>
<th>Syntax</th>
</tr>
</thead>
</table>
| HFS                | /abcd/efgh/filename.txt  
Example:  
-j-DMVCICSctg.out=/var/BCVM/BCVD/BCVWSB/bcvdwbtg.log |
| JCL DDNAME         | //DD:ddname  
Example:  
-j-DMVCICSctg.out=//DD:CTGDBG |
| pre-allocated data set | /’dsnqual1.dsnqual2.dsnqual3  
Example:  
-j-DMVCICSctg.out=//’BCVM.CTG.LOGDATA(BCVDBTG)’  
The data set must be pre-allocated.  
Note: The syntax requires the data set name to be enclosed in single quotations marks, however, in JCL the data set name must enclosed in two sets of single quotation marks. |

Example

To write the messages to a data set member, specify the new startup option on the EXEC statement in the CTG startup JCL:

//CTG EXEC PGM=CTGBATCH,REGION=&REG,ACCT=(5611),TIME=1439,  
PARM='&LEOPT.&CTGHOME/bin/ctgstart-j-DMVCICSctg.out=/*  
’BCVM.BCVDWBTG.LOGDATA(BCVDBTG)’ -noinput' Be sure to type a continuation character in column 72 if needed and enclose the data set name in two sets of single quotation marks.
Known issue for CTG 7.1

If you use MainView for CICS to monitor CTG 7.1 gateways, make sure that MainView for CICS has access to the IBM CTG Statistics API function openRemoteGatewayConnection, which was introduced in CTG 7.2 and is required by this enhancement.

- If you monitor only CTG 7.1 gateways, apply IBM PTF UK38084 to your CTG 7.1 environment (the PTF provides the missing API function).

- If you monitor CTG 7.1 gateways and CTG 7.2 or 8.0 gateways, instead of applying the IBM PTF, you can specify the latest version (7.2 or 8.0) of the CTG Statistics API runtime library, hlq.SCTGDLL, in the STEPLIB or JOBLIB concatenation in the BBI-SS PAS startup JCL (or in the LNKLST).

Failure to take one of the above actions will result in the following error when the MainView for CICS task CMRCTGH initially attempts to communicate with any gateway:

CEE3561S External function openRemoteGatewayConnection was not found in DLL CTGSTATS

This error causes CMRCTGH to stop processing, which disables monitoring for all versions of CTG. A CEEDUMP might also be produced.

Starting and resetting CTG monitoring

Once CTG monitoring is active, you can reset the current monitoring parameters by using the RESET command with the Parm option from an z/OS console, or with the LOG display.

To reset the parameters from an z/OS console, use the MODIFY command:

F pasname,RESET PARM,CMRCTG

where pasname is the BBI-SS PAS stcname.

Use the following command to reset the parameters from the LOG display:

.RESET PARM CMRCTG

When the command is issued, the current CTG parameter values are refreshed with the values in the parameter file, CMRBEXnn. With this command you can:

- turn CTG monitoring support on and off, based on the setting of the CTGMON parameter
increase the maximum number of gateways that can be monitored by a PAS, 
based on the setting of the MAXGATE parameter (this value cannot be decreased 
while the PAS is active)

increase the home data space size, based on the setting of the DSPLIM parameter 
(this value cannot be decreased while the PAS is active)

add or remove gateways from those monitored by the PAS, based on the setting of 
the CTGNAME parameter

Do not use the RESET command to change the PORT and DSPSIZE parameters for 
an existing CTGNAME. To change these parameter, turn CTGMON off and make 
the changes for an existing CTGNAME, then turn CTGMON back on.

See the MainView Administration Guide for more information about the RESET 
command and its parameters.

See "CICS Transaction Gateway" in the MainView for CICS Online Services 
Reference Manual for a description of the views that provide status and performance 
information about the active gateways.

Console interface to the gateway request exit

For CTG version 7.2 and later, when MVCICS CTG monitoring is active, you can 
control the gateway request exit by using the following CTG console interface 
command:

/F ctgname,APPL=RMEXIT,COMMAND=CMR command

where:

ctgname is the name of the CTG daemon JOB/STC to process the request.

CMR command is one of the following commands:

- CMR QUIESCE requests the exit to stop processing transaction data. This 
  command does not terminate the MVCICS gateway request exit; the exit simply 
  idles until a CMR RESUME command is received.

- CMR RESUME requests the exit to resume processing transaction data.
Example
To stop collecting task performance data with the MVCICS gateway request exit, use the following command:

/F BCVDWBTG,APPL=RMEXIT,COMMAND=CMR QUIESCE

To resume collecting task performance data with the MVCICS gateway request exit, use the following command:

/F BCVDWBTG,APPL=RMEXIT,COMMAND=CMR RESUME

Defining history parameters in BBIPROF

Several parameters affect how many records are displayed in the following history views:

- **CHIST and CHISTINT** from the CHISTORY selection dialog
- **CTGHIST** from the CTGHSTRY selection dialog

These parameters, defined below, can be overridden by defining them in a BBIPROF library member. A sample BDR#DEF can be found in the BBSAMP data set.

For MainView terminal sessions, BBIPROF is allocated in the MainView CLIST. Each user can have their own BDR#DEF member by concatenating their user BBIPROF in the MainView CLIST.

For MainView Explorer sessions, a BBIPROF DD statement that references the library containing the BDR#DEF member can be added to the MainView Explorer STC procedure, and that member will be used by all MainView Explorer users.

BDR#DEF contains parameters in the form: KEYWORD VALUE. They can be in any order, except TIMESTAMP which must be the first entry. The following is a sample definition:

```
TIMESTAMP 2007/07/04
MAXREC 10000 :
DEFREC 5000 :
MAXEXC 10000 :
DEFEXC 2000 :
MAXMSG YES;
EXCMSG YES;
```

Definitions are as follows:

- **TIMESTAMP** can consist of any string. The time stamp must appear on the first line.
- **MAXREC** specifies the maximum number of records to return per PAS. The default is 10000.

- **DEFREC** specifies the default number of records to return per PAS if the value is not specified in the CHISTORY view. The default is 5000.

  **Note**

  The MAXREC and DEFREC values are divided by the number of targets in the current context to determine the maximum or default number of records per target for that PAS that are to be returned.

- **MAXEXC** specifies the maximum number of records to read that do not satisfy the parameter set’s filter condition per CICS. The default is 10000.

- **DEFEXC** specifies the number of records to exclude per CICS. The default is 2000.

- **MAXMSG** specifies whether the following message will be displayed when the maximum record count has been reached:

  BBKT6E04I The maximum record count of xxxx has been reached for cccccc at record for yyyy/mm/dd hh:mm:ss

  The variables are as follows:

  - *xxxx* is the value in the **Record Count** field.
  - *ccccc* is the **affected system**.
  - *yyyy/mm/dd* and *hh:mm:ss* are the date and time of the record when the record count is reached.

  Setting MAXMSG to YES means the message will be displayed. The default value is YES.

- **EXCMSG** specifies whether the following message will be displayed when the maximum excluded record count has been reached:

  BBKT6E03I The exclude record count of xxxx has been reached for cccccc at record for yyyy/mm/dd hh:mm:ss

  The variables are as follows:

  - *xxxx* is the value in the **Record Count** field.
  - *ccccc* is the **affected system**.
  - *yyyy/mm/dd* and *hh:mm:ss* are the date and time of the record when the record count is reached.

  Setting EXCMSG to YES means the message will be displayed. The default value is YES.
CTGHIST provides similar messages with message number prefixes BBKRC0* instead of BBKT6E*.

Monitoring SAP transactions

This section describes optional steps for implementing SAP transaction monitoring. Using the CMRCUEX user exit program, MainView for CICS can receive SAP data from the SAPSTEC statistical monitoring exit. To implement SAP transaction monitoring, you must:

- modify the CMRSOPT macro
- add code to the SAPSTEC exit
- activate the SAPSTEC exit

For details about using the CMRCUEX user exit, see “User exit interface (CMRCUEX)” on page 255.

Modifying CMRSOPT

For the collection of SAP data to work properly, you must specify TYPE=PANEL in the CMRSOPT macro.

For a description of this CMRSOPT parameter, see “TYPE=resource statement (optional)” on page 146.

Note
The CMRSOPT view does not support the TYPE parameter; the CMRSOPT macro must be used for SAP transaction monitoring.

Adding code to SAPSTEC

MainView for CICS provides sample code that enables data to be passed from SAPSTEC to CMRCUEX.

The code is contained in these BBSAMP members:
Member | Description
--- | ---
SAPCMROT | passes the SAP internal transaction and report IDs from SAPSTEC through the CMRCUEX user exit program to MainView for CICS
SAPCMRTW | terminates any previously initiated clocks
SAPCMRWT | captures work area waits and issues a call to MainView for CICS to start timing the wait

Each of these members contains Assembler code and instructions for adding the code to the SAPSTEC exit.

In addition, member SAPCMROW contains a complete sample of the SAPSTEC exit.

### Activating the SAPSTEC exit

After you have added the code from the BBSAMP members to SAPSTEC, you are ready to activate the SAP statistical monitoring exit. For details on activating SAPSTEC, refer to your SAP documentation.

### Customizing MainView AutoOPERATOR for CICS

This section describes optional steps for customizing MainView AutoOPERATOR for CICS to your site’s requirements.

The steps described in this section are optional. The full range of MainView AutoOPERATOR for CICS functions are available whether or not you choose to perform the following steps:

- Customize terminal types eligible for broadcast messages.
- Limit CICS transient data queues eligible for rule processing.

### Customizing terminal types eligible for broadcast messages (CAOTTAB)

When the MainView AutoOPERATOR for CICS BROADCAST application is invoked, a list of all terminals eligible for the BROADCAST function is displayed.
The following criteria must be met before a terminal is eligible for broadcasting:

- It must have a CICS status of INSERVICE.
- If it is a VTAM terminal, it must be currently acquired.
- Its model specification must match the specifications of the supplied terminal table.

Because the performance of terminals varies by model, you may need to customize the MainView AutoOPERATOR for CICS terminal table to include the terminal types used at your site.

The CAOTTAB macro generates the MainView AutoOPERATOR for CICS terminal table. The default table is distributed by BMC Software using the following CAOTTAB macro invocation:

```
CAOTTAB (L3277,R3277)
```

Any terminal type that can be specified for the CICS terminal definition operand TERMMODL can be specified for CAOTTAB.

The member TTABJCL in your BBSAMP data set contains sample JCL that can be used to assemble and link CAOTTAB.

**To modify CAOTTAB**

1. Copy TTABJCL from BBSAMP to UBBSAMP, and make sure the new member name conforms to your site’s naming conventions.
2. Read the comments in the member that was created in the previous step, and then modify the JCL accordingly.
3. Modify the sample invocation of the CAOTTAB macro to make all terminal types eligible for BROADCAST messages.
4. Submit the newly created JCL, and check the output to ensure that all steps are completed with a return code of 0.

**Note**

S106 abends can occur if the newly link-edited module causes the BBLINK data set to enter secondary extents and the active BBI-SS PAS attempts to load the module. If this situation happens, stop the BBI-SS PAS and restart it.
Limiting CICS transient data queues eligible for rule processing (CAODTAB)

By default, MainView AutoOPERATOR for CICS intercepts any data written to any CICS extra-partition transient data destination.

If customized to do so, MainView AutoOPERATOR for CICS can also intercept any data written to intrapartition destinations. It then runs its rule processor definitions to determine if a particular message should be intercepted and acted upon.

The default may be acceptable in a CICS system that does not send large amounts of user data to its transient data destinations. But if there is a large volume of data traffic, performance can be impacted.

If your CICS regions have a large volume of transient data traffic and you want to limit the names of the CICS transient data destinations that are eligible for the rule processor message interception, use the CAODTAB macro to generate a CICS transient data name table that can define which queue names are part of the table and that can:

- include only selected intrapartition transient data queue destinations
- include only selected extrapartition transient data queue destinations
- include both selected extrapartition and intrapartition transient data queue destinations
- exclude only selected intrapartition transient data queue destinations
- exclude only selected extrapartition transient data queue destinations
- exclude both selected extrapartition and intrapartition transient data queue destinations

Any CICS extrapartition or intrapartition transient data destination name defined at your site can be specified for CAODTAB.

Using the CAODTAB macro

The member DTABJCL in prefix.BBSAMP contains sample JCL which you can use to assemble and link CAODTAB:

1. Copy DTABJCL from prefix.BBSAMP to prefix.UBBSAMP. Make sure the new member name conforms to site naming conventions.

2. Edit the member created in Step 1 on page 187 by reading the comments in the member and then modifying the JCL accordingly.
3 Modify the sample invocation of the CAODTAB macro so that it uses applicable parameters from “CAODTAB macro parameters” on page 188.

4 Submit the newly created JCL, and check the output to ensure that all steps complete with return code 0.

CAODTAB macro parameters

The following list describes input parameters for the CAODTAB macro:

Table 18: Input parameters for the CAODTAB macro

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Possible values</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXTRA</td>
<td>(YES</td>
</tr>
<tr>
<td>INTRA</td>
<td>(YES</td>
</tr>
<tr>
<td>QUEUES</td>
<td>[(name1),(name2),(namen)...]</td>
</tr>
</tbody>
</table>

The QUEUES= parameter is optional. Use it when you want to explicitly include or exclude specific queues by queue name.

Examples

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

Example

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

EXTRA=(YES, INCLUDE), INTRA=(NO), QUEUES=(QUE1, QUE2, QUE3)

The CICS transient data name table will be an inclusion table for extrapartition transient data messages from queues named QUE1, QUE2, and QUE3. If no queue names are specified (for example, if QUEUES=() or if the queues parameter is not used), all extrapartition transient data messages are included.

Example

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

EXTRA=(NO), INTRA=(YES, INCLUDE), QUEUES=(QUE4, QUE5, QUE6)

The CICS transient data name table will be an inclusion table for intrapartition transient data messages from queues named QUE4, QUE5, and QUE6. If no queue names are specified (for example, if QUEUES=() or if the queues parameter is not used), all intrapartition transient data messages are included.
Example
To specify that both intrapartition and extrapartition transient data messages will be eligible for processing by the Rule Processor:

\[ \text{EXTRA=(YES, INCLUDE), INTRA=(YES, INCLUDE), QUEUES=(QUE7, QUE8)} \]

The CICS transient data name table will be an inclusion table for both extrapartition and intrapartition transient data messages for queues named QUE7 and QUE8. If no queue names are specified (for example, if \(\text{QUEUES=}()\) or if the queues parameter is not used), all extrapartition and intrapartition transient data messages are included.

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

\[ \text{EXTRA=(NO), INTRA=(NO)} \]

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

\[ \text{EXTRA=(NO), INTRA=(YES, INCLUDE)} \]

The CICS transient data name table will be an inclusion table for all intrapartition transient data messages from all queues.

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

\[ \text{EXTRA=(NO), INTRA=(YES, EXCLUDE)} \]

The CICS transient data name table will be an exclusion table for intrapartition transient data messages. In this example, since the \(\text{QUEUES=}\) parameter is not used, no intrapartition queues are explicitly excluded. Thus, all intrapartition queues are included.
MainView for CICS provides the capability to monitor and record the details of a CICS transaction in the form of a trace. Trace data is stored on VSAM log data sets and can be viewed with the online trace facility.

*Note*

For information about displaying application trace data online and how to start a trace, see "Application Trace Facility" in the *MainView for CICS Online Services Reference Manual*.

Logging trace data requires an allocated trace directory and a trace log data set. This chapter describes how to prepare a trace directory and trace log data sets. This chapter also includes a procedure to declare the trace defaults that appear in the MainView for CICS online trace request dialogs.

Preparing the trace directory can be done automatically by MainView Customization as described in the *MainView Customization Reference*, or manually as described in this chapter. Trace log data sets can be allocated in advance, or they can be dynamically allocated at the time of the trace request.

**Preparing a trace directory**

For trace logging to occur, a trace directory must be allocated and initialized before the BBI-SS PAS is initialized.

This section describes how to prepare a trace log directory by using batch jobs from sample members that are located in the BBSAMP and BBPARM data sets.

*Note*

If a security management system is installed at your site, you might need to grant the BBI-SS PAS authority to dynamically allocate trace log data sets.
Defining and initializing a trace directory data set

Each BBI-SS PAS has one trace directory.

A trace directory is a VSAM linear data set containing one entry for each trace log data set. Each entry indicates the date and time when the trace log data set was created, its current status, the trace target, and other related information. Entries can be added or removed from the trace directory to move trace logs to different systems.

To define and initialize a trace directory data set

1. Edit member JXT001 in your site’s BBSAMP data set.
2. Add your job card and edit the statements to meet your site’s naming conventions.
3. Submit the job.

Identifying the trace directory to BBI

Use the following procedure to identify the trace directory to BBI.

1. Edit member BBIISP00 from your site’s BBPARM data set.
2. Find and modify the following statement:
   \[ TRDIR=dsn \]

   Define \( dsn \) as the name of the trace directory data set. This name has no default. Before any trace can be started with trace logging, the directory must be allocated and initialized before BBI-SS PAS initialization. BBSAMP member JXT001 creates the trace directory.

Automating cleanup of the trace directory

If you use MainView for CICS to start many traces that perform logging tasks on a regular basis, you can reach the maximum number of trace entries that can be stored in the trace directory. Reaching the maximum prevents the BBI-SS PAS from starting new traces until you delete the completed entries.

While you can use online history trace administration commands to delete the entries, in high-use environments, this manual process can become tedious. Use the following procedure to implement automatic clean up of the trace directory.
To implement automatic clean up of the trace directory

1 Edit member BBIISP00 from your site’s BBPARM data set.

2 Find and modify the following statement:

```plaintext
TLDSAGE=nnn
TLDSAVE=(nnn,DELDSN)
```

Define *nnn* as the number of days (between 1 and 999) after which, MainView for CICS automatically deletes a trace directory entry from the trace directory. When you specify the DELDSN keyword, MainView for CICS also deletes the trace log data set.

The deletion process occurs during the following events:

- When the BBI-SS PAS initializes
- Daily after midnight

When you specify TLDSAGE, MainView for CICS recalls any migrated trace directories during BBI-SS PAS initialization. For this purpose, BMC recommends specifying a value of at least TLDSAGE=999 because traces that perform logging tasks will fail when the trace directory is migrated.

The following messages appear in the BBI-SS PAS journal and job logs to notify you that traces have been deleted:

```
BBQTDL06I Deleting trace entries older than xxx days
BBQTDL08I Deleted xxx entries from the trace directory
```

The following example shows messages that are also written to the BBI-SS PAS journal when you specify the DELDSN keyword:

```
TL6800I IDC0550I ENTRY (D) BOLTSM.BCVTM67C.L1.T1220.V01.DATA DELETED
TL6800I IDC0550I ENTRY (C) BOLTSM.BCVTM67C.L1.T1220.V01 DELETED
```

Verifying trace directory entries

Trace directory entries are not updated automatically by events occurring outside of the BBI-SS PAS, such as data set deletion or archival. Occasionally, you might need to synchronize a trace directory with the actual status of the trace log data sets.

1 Edit member JXT003 in your site’s BBSAMP data set to verify, purge, or print directory entries.

JXT003 checks for the existence of a trace log data set in the system catalog.
2 Add your job card and edit the statements to meet your site’s naming conventions.

3 Specify a processing option with the PARM statement:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blank</td>
<td>If PARM is not specified (default), an uncataloged entry is marked as INV (INVALID).</td>
</tr>
<tr>
<td>ARCVOL</td>
<td>If ARCVOL (Archive Volser) is specified, this value is matched against the volser in the system catalog for each entry in the directory. If matched, the data set is not verified, which allows you to bypass recalling all trace log data sets from archives.</td>
</tr>
<tr>
<td>LIST</td>
<td>lists the directory entries that are changed. If NOVERIFY is specified or implied, all entries are listed (equivalent to LISTALL).</td>
</tr>
<tr>
<td>LISTALL</td>
<td>lists all entries</td>
</tr>
<tr>
<td>NOLIST</td>
<td>does not list changed entries</td>
</tr>
<tr>
<td>PURGE</td>
<td>deletes any data sets in the directory that are invalid trace data sets</td>
</tr>
<tr>
<td>NOPURGE</td>
<td>does not delete invalid data sets (marked as INV (INVALID) in the directory)</td>
</tr>
<tr>
<td>VERIFY</td>
<td>verifies each of the entries in the trace directory.</td>
</tr>
<tr>
<td>NOVERIFY</td>
<td>does not verify entries in the trace directory</td>
</tr>
<tr>
<td>WRITE</td>
<td>updates trace directory with status changes</td>
</tr>
<tr>
<td>NOWRITE</td>
<td>does not update trace directory with changes detected</td>
</tr>
</tbody>
</table>

4 Submit the job.

**Note**

Verifying trace directories might take an extended period of time because every entry in the trace directory is allocated dynamically and read to verify its current status.

---

### Creating trace log data sets

You can define trace logs manually or let the BBI-SS PAS allocate them dynamically when needed.

This section describes the steps to manually create a trace log.
If you want to learn how to allocate a trace log dynamically, see "Trace facility views" in the MainView for CICS Online Services Reference Manual. It explains how to create log data sets by completing the fields of an online trace facility menu.

Sample member JXT011 located in the BBSAMP data set contains JCL to create a trace log data set. The IDCAMS DEFINE command specifies the characteristics of the data set that is used as a trace log.

Figure 22 on page 195 shows an example of JXT011. You must edit your copy of JXT011 before submitting the batch job to create the trace log. Make the appropriate changes indicated by the comments in this example.

Figure 22: Example of member JXT011 that creates trace log data sets

```assembler
//JXT011 JOB
//* TO CUSTOMIZE:
//* (1) COMPLETE JOB CARD INFORMATION
//* (2) DEFINE NAME OF TRACE LOG DESIRED (MUST END WITH .V01)
//* (3) PROVIDE PROPER VOL VALUE FOR THE DEFINE STATEMENT
//*
//* STEP 1 - DEFINE TRACE LOG DATA SET
//*
//STEP1 EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFINE CL(NAME($PREFIX.$USID.TLDS.V01)-
VOL(VOLSER)-
TRK(10 0)-
SHR(2 3)-
LINEAR)
```

Specifying trace defaults

The data extractor must be active to perform summary or detail tracing.

A detail trace will use the CICS global user exits XEIIN and XEIOUT to trace all EXEC CICS API and SPI calls that are issued by a transaction. A detail trace will trace only transactions that started after the trace unless it is an in-flight trace. A summary trace will trace all of the transaction history records that were written after the trace is started, even if the transaction was started before the summary trace.

Keywords that are specified in member CMRBEX00 of the BBPARM data set define the default values of MainView for CICS trace requests. These values appear in the fields of the online trace request data entry dialogs as the option defaults.
This section describes CMRBEX00 keywords and CMRSOPT keywords that define trace option defaults for:

- all trace requests
- detail trace request
- trace logging

The following CMRSOPT option affects only summary tracing:

- STRACSIZ=nnn determines the size of the trace data space reserved for summary trace.

The following CMRSOPT options also affect detail tracing:

- DB2=YES must be specified to trace DB2 SQL calls
- MQFILE=YES must be specified to trace WebSphere MQ calls
- PSB=YES must be specified to trace of DLI calls
- DLOGPROC=TRSIZE determines the size of the detail trace buffers
- DTRACSIZ=nnn determines the size of the trace data space reserved for detail trace.

For more information about CMRSOPT, see “Setting monitoring conditions (CMRSOPT)” on page 125.

**General trace options**

When the first trace (summary or detail, including in-flight traces) is started for a CICS region, a data space that is specific to that CICS is allocated to manage the trace data.

The data space is destroyed when the last trace for the CICS region is purged. The size of the data space is determined by the STRACSIZ and DTRACSIZ options of the CMRSOPT macro. The approximate size of the data space is as follows:

\[
\text{Data\_Space\_Size} = 2 \times (\text{STRACSIZ} + \text{DTRACSIZ}) + \text{System\_Overhead}
\]
Doubling the STRACSIZ and DTRACSIZ values provides space for trace logging buffers. The System_Overhead value consists of the space required to hold trace tables, linked list elements, and so on. If the calculated value is greater than the maximum allowable size for a data space (2 GB), the STRACSIZ and DTRACSIZ values are adjusted downward until the total size of the data space equals 2 GB.

The following keywords in CMRBEX00 set the default values of the trace display buffer size (STORAGE option) and trace duration (TRTIME option). These options apply to every type of trace and appear on the Start CICS Trace Request dialog.

**STORAGE=500 | nnn**

STORAGE specifies the size of the display buffer in kilobytes. The default is 500 KB. This value overrides the value defined in member BBIISP00. The STORAGE value should be at least four times the specified TRSIZE value.

The display buffer defined by the STORAGE option is used to view trace data for CICS tasks that have issued a CICS SYNCPOINT request. A SYNCPOINT occurs automatically during task termination; it may also be explicitly invoked by an active task. The display buffer contains general task information for both summary and detail traces. For detail traces, the buffer also contains detail event records that identify each EXEC CICS command invoked by the task being traced.

The STORAGE option defines the amount of storage within the Extended Private Area that is to be GETMAINed when a trace is selected for viewing.

---

**Note**

The STORAGE option applies only to the full-screen display of trace data. This option still appears on the windows-mode Start CICS Trace Request dialog, but any change that you make to the value is ignored.

**TRTIME=null | n | hh:mm:ss**

Define $n$ as the maximum trace duration in minutes (1 to 32,000) or $hh:mm:ss$ as the time at which tracing is to stop. The default is null; traces do not have a maximum duration.

---

**Note**

The STOP field of the Start CICS Trace Request data entry dialog is primed with the TRTIME value. If TRTIME is not specified, the STOP field remains blank.
**TRLIM=1000K | nnnnnnnnn**

TRLIM specifies the maximum amount of data space storage that is allocated to a single trace. Storage is expressed in either kilobytes (xxxxK) or bytes (nnnnnnnn).

If the K suffix is not specified with the TRLIM value, the storage size is interpreted as nnnnnnnnn bytes. The default is 1000 KB. Valid values are 1-16383 KB.

If TRLIM is not specified, the default maximum storage limit is the product of the default TRBUFF and TRSIZE values:

\[ \text{Storage\_Limit} = (\text{TRBUFF} \times \text{TRSIZE}) \]

**Note**

If the TRSIZE and TRBUFF values are changed on the CICS Detail Trace Options dialog and (TRSIZE * TRBUFF) exceeds the TRLIM value specified here, the following message is displayed when users try to start the trace:

**BUFFERS EXCEED LIMIT**

---

**Detail trace options**

A pool of buffers is maintained for each active detail trace.

The buffers temporarily store trace data for viewing with the online trace services. The buffer pool is obtained from a CICS-specific trace data space that belongs to the BBI-SS PAS address space. The CMRSOPT option (DTRACSIZ * 2) defines how much of the data space is to be used for detail trace buffers. When that space is exhausted, the trace facility will not start another detail trace for the CICS region.

CMRBEX00 parameters or an online Start Trace request specify the size and number of trace buffer pools. The buffer pools are assigned to the trace being started.

CMRBEX00 keyword values appear in fields of the online Start Trace dialog as the defaults.

**TRBUFF=10 | nnn**

TRBUFF specifies the number of buffers that are allocated for a detail trace request. The default value in the distributed CMRBEX00 member is 10. Valid values are 1 to 999.
The number of trace buffers should be at least equal to two times the average number of active trace requests plus 10%. The TRBUFF value divided by 2 is the maximum number of concurrent tasks that can be traced.

If another value is entered from the CICS Detail Trace Options dialog, it overrides the value that is specified with the CMRBEX00 TRBUFF keyword.

---

Note

If a traced event requires more than two buffers to hold the data and the trace is not being logged, some data might be lost.

---

**TRSIZE=50K | nnnK**

TRSIZE can be used in one of two ways:

- TRSIZE * TRBUF is the size of the buffer pool reserved for this detail trace in the trace data space.
  
  For example, TRSIZE=50K and TRBUF=10 will reserve 500 KB of storage in the trace data space for this trace.

- If DLOGPROC=TRSIZE in CMRSOPT, the TRSIZE value is also the size of the trace buffer for each task that is being traced.; otherwise, the buffer size is 48 KB.

The default value in the distributed CMRBEX00 member is 50 KB. Valid values are 1-999 KB, but the number is rounded to a multiple of 4 KB.

If the TRSIZE field is specified with another value from the CICS Detail Trace Options dialog, it overrides the CMRBEX00 TRSIZE keyword.

---

Note

The default TRBUFF and TRSIZE values shown here are as distributed in member CMRBEX00 of the BBPARM data set. If you do not create a customized version of CMRBEX00 in the UBBPARM data set, these default values (10 and 50K) remain in effect.

However, if you create a customized version of CMRBEX00 but do not specify a TRBUFF or TRSIZE value, the internal defaults of 20 and 400K take effect. In addition, if the resulting TRLIM value (TRBUFF * TRSIZE) is less than 8000 KB, (20 * 400 KB), the value is increased to 8000 KB.

---

**Trace logging options**

The following keywords define the defaults to allocate a trace log data set.
TRPREFIX=**highLevel**

TRPREFIX defines the high-level prefix of trace log data set names if the Log DSN field of the Start CICS Trace Request panel is specified without quotation marks.

If TRPREFIX is left blank, the ID of the user requesting the trace is the high-level qualifier of the trace log data set name.

**Note**
MainView for CICS generates a trace log data set name with TRPREFIX and the generated name cannot exceed 44 bytes. If the generated name exceeds 44 bytes, MainView for CICS excludes the date (mmmdd) node and, if necessary, the time (Thhmm) node from the generated name.

TRREUSE=Y | N

TRREUSE specifies whether the data within a trace log can be overwritten if it is not reset. The default is Y.

N (NO) indicates that data is not overwritten.

If the request specifies a 1 for the number of logs and N is defined for TRREUSE, data is not recorded. If the request specifies a 1 and Y is defined for TRREUSE, previously recorded data is overwritten.

TRVOLS=(vol1,vol2...vol7)

Define *voln* as the ID of the default volumes that are allocated for trace log data sets. Up to seven volumes can be specified. The default is SYSDA.

TRCYL=3 | nnn

TRCYL defines the default primary cylinder allocation (CYLS option) for trace log data sets. The default is 3. Valid values are 1 to 999.

TRSUFX=D | suffix

TRSUFX specifies the default suffix added to the trace cluster DSN (Data DSN Suffix option) to make the data set name for the data component. The default is D.

TRSMSSCL=name
TRSMSSCL defines the default name of the SMS storage class for trace log data set allocation. This keyword has no default value.

**TRSMSDCL= name**

TRSMSDCL defines the default name of the SMS data class for trace log data set allocation. This keyword has no default value.

**TRSMSMCL= name**

TRSMSMCL defines the default name of the SMS management class for trace log data set allocation. This keyword has no default value.
Implementing security

To secure MainView for CICS, BMC recommends that you use your existing site external security manager that uses the z/OS system authorization facility (SAF) interface.

SAF passes security requests to external security managers (ESMs). SAF security supports the IBM RACF product and the CA Technologies CA-ACF2 and CA-Top Secret products.

Profile names define product resources that are secured by an ESM through the SAF interface. The following books provide the information that you need to secure MainView for CICS resources with an ESM:

- MainView Security Reference Manual lists the SAF entity names for MainView for CICS services, views, and actions.

- MainView Security Guide describes how to use Plex Manager security views to manage security parameter and resource class property members in the BBSECURE data set.

MainView for CICS provides a proprietary security macro (CMRSECU) that can protect action line commands from full-screen displays and from some window-mode views; however CMRSECU can protect only full-screen display access.

When a full-screen display is replaced by a window-mode view, the full-screen display is no longer supported. The action parameter table (Table 19 on page 206) is not updated when more actions are available for window-mode views.

This section documents the macro for compatibility purposes. If you are a new MainView for CICS user, you should not use the CMRSECU security method. In addition, current users should migrate to ESM security protection.

**Note**

Support for the CMRSECU macro might be withdrawn in a future release of MainView for CICS.
Modifying the CMRSECU macro

You can use the CMRSECU macro that is found in your BBSAMP data set to authorize security for MainView for CICS and MainViewAutoOPERATOR for CICS.

The CMRSECU macro assembles and links the CMRSECU module and creates a table of security definitions.

To modify the CMRSECU macro

1. Copy member CMRASM from BBSAMP to the UBBSAMP data set, and make sure the new member name conforms to your site’s naming conventions.

2. Read the comments in the member that was created in Step 1 on page 204, and then modify the JCL accordingly.

Keep the following issues in mind:

- CMRSECU must be assembled and linked into BBLINK or an authorized load library that is concatenated in front of BBLINK.
- CMRSECU must be linked reentrant.

3. Modify the sample CMRSECU macro according to your site’s security requirements.

A sample of CMRSECU is in member CMRUSECU of your BBSAMP data set. The CMRSECU macro can contain up to three different TYPE statements:

- The CMRSECU TYPE=INITIAL statement defines security for each
  —MainView for CICS function or action service
  —MainView AutoOPERATOR for CICS action service
- The CMRSECU TYPE=ENTRY statement further defines security by user ID. For example, specific CICS services can be restricted to individual users.
- A single CMRSECU TYPE=FINAL statement ends the input.

Multiple sets of TYPE=INITIAL and TYPE=ENTRY statements can be specified in the CMRSECU macro. The TYPE statements and their parameters are described in the following sections.

4. Submit the JCL, and check the job output to make sure all steps are completed with a return code of 0.
S106 abends can occur if the newly link-edited module causes the BBLINK data set to enter secondary extents and the active BBI-SS PAS attempts to load the module. If this situation happens, stop the BBI-SS PAS and restart it.

**CMRSECU TYPE=INITIAL statement (required)**

The CMRSECU TYPE=INITIAL statement defines security for each:

- **MainView for CICS** function or action service
- **MainView AutoOPERATOR for CICS** action service

To specify which target or targets the security definitions apply to, type

```
TYPE=INITIAL,TARGET=[(target1,target2,...,targetn.) | *]
```

**Note**
The asterisk (*) indicates that security definitions apply to all non specified targets. If an * is not specified and a nonspecified target is referred to online, the first CMRSECU TYPE=INITIAL statement is used to define function security for that target. Refer to BBSAMP member CMRUSECU for an example.

To set function security, enter **YES** or **NO** for each function parameter:

- **YES** indicates the function is protected. Only operator IDs with YES specified in the CMRSECU TYPE=ENTRY statement are allowed to access that function (see “CMRSECU TYPE=ENTRY statement (required)” on page 210.

- **NO** indicates the function is not secured. All operator IDs that are allowed access to MainView for CICS or MainView AutoOPERATOR for CICS are also allowed access to the function.

If the function is protected and an unauthorized operator tries to access it, an error message is issued to the operator and logged.

Display parameters apply to MainView for CICS services; if you do not have MainView for CICS installed, the parameters in the Display Parameter list do not apply. Action parameters apply to MainView for CICS functions and MainView AutoOPERATOR for CICS IMFEXEC CICS commands. For IMFEXEC CICS syntax requirements, refer to the MainView AutoOPERATOR Advanced Automation Guide.
Note

Two function parameters are listed for MainView for CICS services that display information and perform actions: one secures access to the display and the other secures the service actions. For example, TASK is both a display and an action service. Specifying TASK=NO allows unlimited access to the TASK statistics and MEMORY displays only. Specifying ALTTASK=YES secures the action services related to tasks.

Action parameters for the CMRSECU TYPE=INITIAL statement are listed in Table 19 on page 206. Display parameters for the CMRSECU TYPE=INITIAL statement are listed in Table 20 on page 208.

The default for each parameter is NO.

**Table 19: Action parameters for the CMRSECU macro**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>MainView for CICS service description</th>
<th>MainView AutoOPERATOR for CICS IMFEXEC command</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTAID=NO</td>
<td>YES</td>
<td>specifies whether to secure the change capability for the Automatic Initiate Descriptor statistics display and the CIAD x views</td>
</tr>
<tr>
<td>ALTCLASS=NO</td>
<td>YES</td>
<td>specifies whether to secure the change capability for the classes display</td>
</tr>
<tr>
<td>ALTCONN=NO</td>
<td>YES</td>
<td>specifies whether to secure the status change capability in the CONNECT service and the CCONN x views The expanded displays are included.</td>
</tr>
<tr>
<td>ALTDEST=NO</td>
<td>YES</td>
<td>specifies the status change capability for the DEST service</td>
</tr>
<tr>
<td>ALTER=NO</td>
<td>YES</td>
<td>specifies whether to secure the change capability for the virtual storage address display</td>
</tr>
<tr>
<td>Parameter</td>
<td>MainView for CICS service description</td>
<td>MainView AutoOPERATOR for CICS IMFEXEC command</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| ALTFILE=NO | YES                                                                                                    | specifies whether to secure the status change capability in the FILE, DDIR, DSNAMES, and DATATABLE services; and the CFILE x views | IMFEXEC CICS ALLOC  
IMFEXEC CICS CLOSE FILE  
IMFEXEC CICS DISABLE FILE  
IMFEXEC CICS ENABLE FILE  
IMFEXEC CICS FREE  
IMFEXEC CICS OPEN FILE  
IMFEXEC CICS RECOVERDB  
IMFEXEC CICS STARTDB  
IMFEXEC CICS STOPDB       |
| ALTICES=NO | YES                                                                                                     | specifies whether to secure the change capability in the ICE display and the CICES x views                     | IMFEXEC CICS PURGE ICE     |
| ALTPROG=NO | YES                                                                                                     | specifies whether to secure the status change capability in the PROGRAM and REMOTES services, and the CPROG x views | IMFEXEC CICS DISABLE PROGRAM  
IMFEXEC CICS DROP  
IMFEXEC CICS ENABLE PROGRAM  
IMFEXEC CICS LOAD PROGRAM  
IMFEXEC CICS NEWCOPY PROGRAM     |
| ALTTASK=NO | YES                                                                                                     | specifies whether to secure the task kill capability in the TASK, ENQUEUE, TERMINAL, CONSOLES, and CONNXPND services; and in the TASK x TASKDSA x, TASKLCK x, and TASKFIL x views | IMFEXEC CICS KILL     |
| ALTTERM=NO | YES                                                                                                     | specifies whether to secure the status change capability in the TERMINAL and CONSOLES services, and the CTERM x views | IMFEXEC CICS ACQUIRE  
IMFEXEC CICS INSERVE  
IMFEXEC CICS OUTSERVE  
IMFEXEC CICS RELEASE  
IMFEXEC CICS SPURGE         |
| ALTTRAN=NO | YES                                                                                                     | specifies whether to secure the status change capability in the TRAN display and the CTRAN x views            | IMFEXEC CICS ENABLE TRAN  
IMFEXEC CICS DISABLE TRAN     |
<p>| ALTTSUT=NO | YES                                                                                                     | specifies whether to secure the status change capability in the TSUT display and the TSQUEUE x views         | IMFEXEC CICS PURGE TSUT   |
| SETDMPD=NO | YES                                                                                                     | specifies whether to secure the change capability of the dump data set fields in the CREGSYS view               | IMFEXEC CICS ALTER     |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>MainView for CICS service description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SETJVMP=NO</td>
<td>YES specifies whether to secure the change capability of the CJVMPL view</td>
</tr>
<tr>
<td>SETSYS=NO</td>
<td>YES specifies whether to secure the change capability of the system settings fields in the CREGSYS view</td>
</tr>
<tr>
<td>SETTCPS=NO</td>
<td>YES specifies whether to secure the change capability of the CTCPSRV view</td>
</tr>
<tr>
<td>SETTCLA=NO</td>
<td>YES specifies whether to secure the change capability of the CTRNCLA view</td>
</tr>
</tbody>
</table>

**Note**
In the CTRNCLA view, modifying the 'Max Act' value of transaction class DFHTCL00 is the same as modifying the 'Max Tasks' field and is secured by the SETSYS=parameter. Modifications of other classes of transactions in the CTRNCLA view are secured by the SETTCLA parameter.

Table 20: Display parameters for the CMRSECU macro

<table>
<thead>
<tr>
<th>Parameter</th>
<th>MainView for CICS service description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABEND=NO</td>
<td>YES specifies whether to secure the abend status display</td>
</tr>
<tr>
<td>AID=NO</td>
<td>YES specifies whether to secure the Automatic Initiate Descriptor (AID) statistics display</td>
</tr>
<tr>
<td>CLASS=NO</td>
<td>YES specifies whether to secure the class maximum statistics display</td>
</tr>
<tr>
<td>CONNECT=NO</td>
<td>YES specifies whether to secure the connection statistics display The expanded displays are included.</td>
</tr>
<tr>
<td>DB2SYSP=NO</td>
<td>YES specifies whether to secure the DB2 system parameters and DBCTL displays</td>
</tr>
<tr>
<td>DB2TASK=NO</td>
<td>YES specifies whether to secure the DB2 task statistics and DBCTASK displays</td>
</tr>
<tr>
<td>DEST=NO</td>
<td>YES specifies whether to secure the destination ID information display</td>
</tr>
<tr>
<td>DLI=NO</td>
<td>YES specifies whether to secure the DBCTL statistics display</td>
</tr>
<tr>
<td>DUMP=NO</td>
<td>YES specifies whether to secure the virtual storage address display</td>
</tr>
<tr>
<td>ENQUEUE=NO</td>
<td>YES specifies whether to secure the enqueued resources display</td>
</tr>
<tr>
<td>EXITS=NO</td>
<td>YES specifies whether to secure the task-related and global user exits display</td>
</tr>
<tr>
<td>FILE=NO</td>
<td>YES specifies whether to secure the files statistics display in the DBCTL data base display, the file dsnames displays, and the CICS data tables displays The expanded displays are included.</td>
</tr>
</tbody>
</table>

CMRSECU TYPE=INITIAL statement (required)
<table>
<thead>
<tr>
<th>Parameter</th>
<th>MainView for CICS service description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HISTORY=NO</td>
<td>YES</td>
</tr>
<tr>
<td>ICES=NO</td>
<td>YES</td>
</tr>
<tr>
<td>JOURNAL=NO</td>
<td>YES</td>
</tr>
<tr>
<td>MONITOR=NO</td>
<td>YES</td>
</tr>
<tr>
<td>NUCLEUS=NO</td>
<td>YES</td>
</tr>
<tr>
<td>PLAN=NO</td>
<td>YES</td>
</tr>
<tr>
<td>PPST=NO</td>
<td>YES</td>
</tr>
<tr>
<td>PROBLEM=NO</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>If PROBLEM=YES, you must specify OPID=xxxxxxxx in the TYPE=ENTRY statement. The OPID= specification must equal the AUTOID or USRID specified in BBPARM member BBIISP00.</td>
</tr>
<tr>
<td>PROGRAM=NO</td>
<td>YES</td>
</tr>
<tr>
<td>PSB=NO</td>
<td>YES</td>
</tr>
<tr>
<td>REVIEW=NO</td>
<td>YES</td>
</tr>
<tr>
<td>SHARE=NO</td>
<td>YES</td>
</tr>
<tr>
<td>SUBPOOL=NO</td>
<td>YES</td>
</tr>
<tr>
<td>SUFFIX=NO</td>
<td>YES</td>
</tr>
<tr>
<td>SUMMARY=NO</td>
<td>YES</td>
</tr>
<tr>
<td>TASK=NO</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>The expanded displays are included.</td>
</tr>
<tr>
<td>TEMP=NO</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>The expanded displays are included.</td>
</tr>
<tr>
<td>TERM=NO</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>The expanded displays are included.</td>
</tr>
<tr>
<td>TIOT=NO</td>
<td>YES</td>
</tr>
<tr>
<td>TRAN=NO</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>The expanded displays are included.</td>
</tr>
<tr>
<td>TSUT=NO</td>
<td>YES</td>
</tr>
<tr>
<td>VTAM=NO</td>
<td>YES</td>
</tr>
</tbody>
</table>
The CMRSECU TYPE=ENTRY statement and its parameters are used to qualify access to each of the following services that has been secured through the CMRSECU TYPE=INITIAL statement:

- MainView for CICS function or action service
- MainView AutoOPERATOR for CICS action service

One CMRSECU TYPE=ENTRY statement is required for each operator ID that requires access to the previously secured function or service.

If the function is protected and an unauthorized operator tries to access it, an error message is issued to the operator and logged.

Specify **YES** or **NO** for each function or action service that has been secured by CMRSECU TYPE=INITIAL where

- **YES** indicates the operator can access the function or service that has been secured.
- **NO** indicates the operator cannot access the function or service that has been secured.

Parameters for the CMRSECU TYPE=ENTRY statement are:

**OPID=xxxxxxxx | ***

OPID specifies an operator ID, where:

- xxxxxxxx is a:
  - TSO user ID for access through a terminal session
  - EXCP user ID for access through an EXCP session
  - VTAM user ID for access through a VTAM session
- * is a generic qualifier for an operator ID (for example, A6*).

The display and action parameters are the same as the CMRSECU TYPE=INITIAL parameters.

Action parameters for the CMRSECU TYPE=INITIAL statement are listed in Table 19 on page 206. Display parameters for the CMRSECU TYPE=INITIAL statement are listed in Table 20 on page 208.
CMRSECU TYPE=FINAL statement (required)

The CMRSECU TYPE=FINAL statement has one parameter and ends this option generation; an example follows:

```
CMRSECU TYPE=FINAL
```

Resetting the CMRSECU macro

To reset the CMRSECU table dynamically, issue the following command:

```
:RESET PARM CMRSECU
```

See the MainView Administration Guide for a description of the RESET control command and supported parameters.
Common tasks

This chapter summarizes some of the common tasks that are involved in customizing MainView for CICS.

Use the checklists that are provided in this chapter to guide you through each task and help you find more information.

Adding a CICS region

Use the following procedure to add a CICS region.

1. Depending on the method that is used to control CICS to BBI-SS PAS communications (see “BBI-SS PAS and CICS target connection” on page 59), select one of the following methods of adding the target to the JNT:
   - If you are using the AutoConnect facility, you do not need to add the target to the JNT.
   - If you are using dynamic targets, add and install a new target definition (TGTDEF) for the CICS region (see "Managing Targets and Target Contexts" in the MainView Administration Guide).
   - If you are using static targets, update member BBIJNT00 of the UBBPARM data set with a new TARGET entry for the CICS region (see the "Define BBI-SS PAS Suffixes and Target System Parameters" step in the MainView Customization Reference).

2. If you want to collect CICS task performance history data (T6E), ensure that you have a valid CMRDETL definition for the CICS region (see “Using CMRDETL data sets” on page 77).

3. If you want to automatically start MainView for CICS monitors, update UBBPARM member BBIISP00 to add a target entry for the CICS region (see the "Specify BBI Interval Services Parameters" step in the MainView Customization Reference).
4 Modify the CICS startup procedures by following steps 1-7 in the “Completing implementation” on page 110.

Upgrading CICS regions

This topic provides information on upgrading CICS regions.

If you have implemented the steps as described in “Completing implementation” on page 110, you do not need to do anything additional on the CICS side as long as you use the same JCL, CSD, PLT, SIT and MCT in CICS.

If any of these have changed, follow the appropriate steps in “Completing implementation” on page 110.

You can use the same CMRDETL data sets for the upgraded CICS.

Starting communication between a CICS region and BBI-SS PAS

To activate communication between a CICS region and the BBI-SS PAS, you must use one of the following CICS transactions:

- **FST2**
  This transaction can be issued from any device or invoked from a MainView AutoOPERATOR EXEC. It activates communication only when issued without any operands.
  See “FST2 transaction” on page 263.

- **CREGAGT view**
  See “Commands available with the CREGAGT view” on page 268


Starting and stopping task data monitoring

To start and stop task data monitoring (the extractor) in a CICS region, you must perform the following task.
This procedure provides full global and transaction level data collection, which enables you to view data in the online Graph services and Summary display.

Use one of the following:

- **CREGAGT view**
  See “CREGAGT view commands” on page 267.

- **QON and QOF action commands**
  See “CREGAGT view commands” on page 267.

- **FST2 QON | QOFF**
  This transaction can be issued from any device or invoked from a MainView AutoOPERATOR EXEC. Use this transaction with the appropriate parameter to start (QON) or stop (QOFF) full data collection.
  Communication between the CICS region and the BBI-SS PAS remains active even though data collection has stopped.
  See “FST2 transaction” on page 263.

- **SMN2**
  This transaction can be issued only from a terminal. It displays a menu to establish communication and task data collection. SMN2 also can be used to stop task data collection. Communication cannot be stopped after it is started.

### Starting data collection for other products

Use the following procedure to start collecting data for fourth-generation language (4GL) or third-party database product.

1. Use the CMRSOPT parameters (with either the CMRSOPT macro or the CMRSOPT view), or sample exits that are supplied by MainView for CICS.

2. Reset the CMRSOPT if you are using CMRSOPT view, or assemble and link your modified source module (CMRUSOPT) then reset the CMRSOPT.

   See “Setting monitoring conditions (CMRSOPT)” on page 125.

### Adding or deleting graph data

Use the following procedure to add or delete resources to be graphed.
1 Modify CMROPT options.

See “TYPE=resource statement (optional)” on page 146. Also note the approximate storage requirements that are discussed in that section.

2 Assemble and link your modified source module (CMRUSOPT) as described in “Setting monitoring conditions (CMROPT)” on page 125.

Setting display and command security

MainView for CICS displays (fullscreen and window) and commands can be protected using site external security manager (RACF, TOPSECRET, ACF2, etc.) or proprietary security.


**Note**

Window views can only be protected by ESM security.

Changing problem thresholds

To change problem thresholds, use one of the following methods:

- See “Setting problem thresholds (CMRPRBT)” on page 150 for information about changing thresholds using the CMRPRBT macro.

- Use the CMRPRBT view as described in the MainView for CICS Online Services Reference Manual.

Preparing CMRDETL recording and archiving

This topic provides information about preparing CMRDETL recording and archiving.

To set up and enable detail file recording and archiving, see “Using CMRDETL data sets” on page 77.
Modifying graph thresholds

Use the following procedure to modify graph thresholds:

1. Set the response time thresholds for graphed resources such as files, transactions, terminals, panels, MAPs, and PSBs.

   The MAXRESP parameter indicates the warning level; the severe level is automatically double the specified value. See “TYPE=FINAL statement (required)” on page 149 to change the MAXRESP parameter.

2. Assemble and link your modified source module (CMRUSOPT) as described in “Setting monitoring conditions (CMRSOPT)” on page 125.

   **Note**

   Thresholds for non resource graph fields are fixed.

   See “Field Cross-Reference for Graphs” in the *MainView for CICS Online Services Reference Manual* for the threshold value for each graphed field.

Adding or deleting global statistical records

Use the following procedure to add or delete global statistical records.

1. Change the default parameters that are specified in CMRSOPT macro or CMRSOPT view to add or delete global statistic record types.

   See “TYPE=INITIAL statement (default conditions)” on page 146.

   Valid CMRSTATS record types are CSA, DCT, DLZ, DMP, ENQ, FCT, FEPI, GNI, IRC, JCT, LTX, MON, OTHR, PAM, PCT, POL, PPT, RCT, STI, TCT, TST, T6F, and WEB.

   The default data collection interval for most record types is either 60 (once each hour) or 0 (the record is not recorded).

   See "MainView for CICS Records" in the appendix of the *MainView for CICS PERFORMANCE REPORTER User Guide* for the format of each global statistics record.

2. Reset the CMRSOPT if you are using CMRSOPT view, or assemble and link your modified source module (CMRUSOPT) and then reset the CMRSOPT.

   See “Setting monitoring conditions (CMRSOPT)” on page 125.
Adjusting for Daylight Saving Time DST

MainView for CICS maintains time-related data in various control blocks on the BBI-SS PAS and the CICS regions that are executing MainView for CICS.

This data is used to calculate several durations that are used by monitors and task performance data. If the BBI-SS PAS is not restarted after the DST change, the data might not be updated with the proper time setting. This situation might result in misleadingly large values after the spring-forward time change or negative values after the fall-back change.

Because of the fall-back time change, the FIC2 transaction (which runs every minute in CICS to collect CMR and CICS statistics) might be set to run in the next hour instead of the next minute. The next scheduled transaction would then run every minute.

The spring-forward change might cause the next FIC2 transaction to run immediately, possibly producing one or two FT003W PREVIOUS EXECUTION OF FIC2 IS STILL ACTIVE messages in the CICS job log.

You can avoid misleading or incorrect time-related data (such as overall response time) for tasks or monitors that are active during the timechange, and for the FIC2 scenarios described in this section. BMC recommends that you stop the MainView for CICS BBI-SS PAS, perform the DST time change, and re-start the BBI-SS PAS. If this is not a concern, you do not have to re-start the BBI-SS PAS.
Additional MainView AutoOPERATOR for CICS functions

This appendix identifies additional MainViewAutoOPERATOR functions that become available when you establish communications between the BBI-SS PAS and CICS.

After you have implemented BBI-SS PAS to CICS communication, you can:

■ use the Rule Processor to capture messages from the CICS transient data queue
■ issue the C and P line commands from the STATUS application’s TASK display
■ issue the CHAP (Change Task Priority) line command from the STATUS application’s TASK display
■ use the BROADCAST application from the CICS Operator Workstation in the BBI-TS
■ use the CICS dependent IMFEXEC CICS commands from an EXEC

The CICS dependent IMFEXEC CICS commands are as follows:

Table 21: CICS dependent IMFEXEC CICS commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACQUIRE TERMINAL</td>
<td>Acquire a VTAM-supported terminal.</td>
</tr>
<tr>
<td>ALLOC</td>
<td>Allocate a data set to a CICS region.</td>
</tr>
<tr>
<td>CEMT</td>
<td>Issue a CICS master terminal command.</td>
</tr>
<tr>
<td>CHAP</td>
<td>Change a task’s priority in a CICS region.</td>
</tr>
<tr>
<td>CLOSE</td>
<td>Close a file in a CICS region.</td>
</tr>
<tr>
<td>DISABLE</td>
<td>Disable a CICS resource.</td>
</tr>
<tr>
<td>DROP</td>
<td>Decrease the use count of a CICS program.</td>
</tr>
<tr>
<td>DUMPDB</td>
<td>Prepare a database for dumping.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ENABLE</td>
<td>Make a CICS resource available for use.</td>
</tr>
<tr>
<td>FREE</td>
<td>Deallocate a data set to a CICS region.</td>
</tr>
<tr>
<td>INSERVE</td>
<td>Place a CICS resource in service.</td>
</tr>
<tr>
<td>KILL TASK (PURGE, FORCEPURGE)</td>
<td>Terminate a task in a CICS region by task number.</td>
</tr>
<tr>
<td>KILL TERM</td>
<td>Terminate a task in a CICS region by term ID.</td>
</tr>
<tr>
<td>LOAD</td>
<td>Load a program in a CICS region.</td>
</tr>
<tr>
<td>NEWCOPY</td>
<td>Load a new version of a program in a CICS region.</td>
</tr>
<tr>
<td>OPEN</td>
<td>Open a file in a CICS region.</td>
</tr>
<tr>
<td>OUTSERVE</td>
<td>Place a CICS resource out of service.</td>
</tr>
<tr>
<td>PURGE</td>
<td>Purge a resource in a CICS region.</td>
</tr>
<tr>
<td>RECOVERYDB</td>
<td>Prepare a database for recovery.</td>
</tr>
<tr>
<td>RELEASE TERMINAL</td>
<td>Release a VTAM-supported terminal.</td>
</tr>
<tr>
<td>STARTDB</td>
<td>Start a database.</td>
</tr>
<tr>
<td>STOPDB</td>
<td>Stop a database.</td>
</tr>
</tbody>
</table>

However, if you choose not to implement BBI-SS PAS to CICS communication, you can still use MainViewAutoOPERATOR to:

- use the STATUS application from the CICS Operator Workstation in the BBI-TS
- issue the K, D, and F line commands from the STATUS application’s TASK display
- use the CICS independent IMFEXEC CICS commands from an EXEC

The CICS independent IMFEXEC commands are as follows:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLOC (LOCAL only)</td>
<td>Allocate a data set to the BBI-SS PAS.</td>
</tr>
<tr>
<td>ALTER</td>
<td>Change CICS task related throttles.</td>
</tr>
<tr>
<td>ALTERVVS</td>
<td>Change virtual storage in the CICS region.</td>
</tr>
<tr>
<td>FREE (LOCAL only)</td>
<td>Deallocate a data set from the BBI-SS PAS.</td>
</tr>
<tr>
<td>KILL TASK (KILL, WITH DUMP, FORCE)</td>
<td>Terminate a task in a CICS region by task number.</td>
</tr>
<tr>
<td>QUERY</td>
<td>Invoke a MainView for CICS service.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>SPURGE</td>
<td>Change the spurge value for CICS transactions.</td>
</tr>
</tbody>
</table>
Deactivating CICS global user exit trace

MainView for CICS and MainView AutoOPERATOR use CICS global user exits to gather information from a region. With CICS trace active, the tracing of exits adds overhead and decreases the amount of user information that is available in the incore trace table. You can turn off the user exit trace entries that are generated by MainView for CICS and MainView AutoOPERATOR. However, doing so also deactivates any other user exit trace entries; other products and user code that include user exits are no longer traced.

To turn off user exit tracing, you can:

- set the SIT parameters SPCTRxx and STNTRxx (where xx is the user exit interface) to suppress component tracing at CICS initialization time:
  - SPCTRUE=OFF (suppress special tracing for user exit interface)
  - STNTRUE=OFF (suppress standard tracing for user exit interface)

- use the CETR transaction
MainView products occasionally start a CICS transaction by using interval control services.

The start of such a transaction must not be prevented from completing because of the existence of any recoverable temporary storage definitions in the CICS temporary storage table (TST).

**Temporary storage prefix CMRI**

If a conflict exists between the MainView temporary storage prefix and an existing temporary storage definition, the prefix that is used by MainView (CMRI) must be changed by applying the following zap:

```
NAME CMRINT2 CMRINT2
VER 00BC  C3D4D9C9
REP 00BC xxxxxxxx  (any nonrecoverable prefix)
```

**Temporary storage queue BMCMVCTQ**

MainView for CICS creates a temporary storage queue named BMCMVCTQ, which it uses to save status information across terminations and initialization of MainView for CICS in an auto maintenance environment.

The queue is created when FST2 TERM is entered, and it remains in the system until CICS is recycled. It is deleted and re-created for subsequent TERM/INIT processes.
To work properly, no conflict can exist between BMCMVCTQ and an existing temporary storage definition. If a conflict exists, the temporary storage queue name can be changed by applying the following zap:

```
NAME CMRINT2 CMRINT2
VER 0200 C2D4C3D4E5C3E3D8
REP 0200 xxxxxxxx (any site-acceptable name)
```

### Enqueue BMCFST2Q

The FST2 transaction issues an enqueue on BMCFST2Q when an FST2 transaction is initiated. Subsequent FST2 transactions will attempt to enqueue on this name before running and will terminate if the enqueue is busy.

If a conflict exists with any enqueue definition or enqueue model, the following zap can be used to change the name of the MainView for CICS enqueue name:

```
NAME OLTCNTL OLTCNTL
VER 003A C2D4C3C6E2E3F2D8
REP 003A xxxxxxxxxxx (any site-acceptable name)
```

### Enqueue BBK1CRTR

The FST2 transaction issues an enqueue on BBK1CRTR when an FST2 LON/LOFF transaction is initiated.

If a conflict exists with any enqueue definition or enqueue model, the following zap can be used to change the name of the MainView for CICS enqueue name:

```
NAME CMRINT2 CMRINT2
VER 01F0 C2C2D2F1C3D9E3D9
REP 01F0 xxxxxxxxxxx (any site-acceptable name)
```

Refer to the following sources to determine whether a conflict exists between the MainView temporary storage prefix and an existing temporary storage definition:

- For MainView AutoOPERATOR, see the "Implementation Considerations" section of the `MainView AutoOPERATOR Customization Guide`.

- For MainView for CICS and MainView AutoOPERATOR, see “General considerations” on page 106.
Enqueue CMRJRNL

The JNL2 transaction issues an enqueue on CMRJRNL when the transaction starts.

This enqueue exists for the life of the JNL2 task and is used to prevent additional JNL2 tasks from initializing.

If a conflict exists with any other site enqueue definition or enqueue model, the following zap can be used to change the name of the MainView for CICS enqueue name:

<table>
<thead>
<tr>
<th>NAME</th>
<th>CMRINT2</th>
<th>CMRINT2</th>
</tr>
</thead>
<tbody>
<tr>
<td>VER</td>
<td>0090</td>
<td>C3D4D9D1D9D5D3</td>
</tr>
<tr>
<td>REP</td>
<td>0090</td>
<td>xxxxxxxxxxxxxxx (any site-acceptable name up to 7 characters)</td>
</tr>
</tbody>
</table>

After the zap is applied, you need to restart the CICS or use the RES command from the CREGAGT view if MainView for CICS in the target CICS is in auto maintenance mode.
Controlling functions manually

This appendix describes how to control the following MainView for CICS functions manually:

- BBI-SS PAS-to-CICS communication
- monitoring
- task kill exit
- completely initiate and terminate MainView for CICS

For instructions about implementing these functions for automatic startup, proceed as follows:

- When implementing MainView AutoOPERATOR only, refer to “Implementing BBI-SS PAS-to-CICS Communication” in the MainView AutoOPERATOR Customization Guide.

- When implementing MainView for CICS and MainView AutoOPERATOR, refer to “Step 4: Modify the PLT” on page 114.

Initializing MainView for CICS

If you modified the PLT, MainView for CICS monitoring starts automatically at CICS startup unless START=NO is specified in the CMRSOPT options table.

This modification starts BBI-SS PAS-to-CICS communication, and optionally starts MainView for CICS monitoring, and the task kill function. If you did not modify the PLT, MainView for CICS can be initialized manually from:

- a CICS terminal
- a z/OS console that is defined to CICS
the CREGAGT view in a BBI-SS terminal session (using windows mode for a static target)

To initialize MainView for CICS from a CICS terminal

1 Log on to a CICS terminal.

2 Type FST2 INIT.

3 Press Enter.

To initialize MainView for CICS from a z/OS console that is defined to CICS, use the MODIFY command to invoke the FST2 transaction by using the INIT option. For example, the following command initializes MainView for CICS in the CICS region with the z/OS job name of CICSPROD:

F CICSPROD,FST2 INIT

To initialize MainView for CICS from a BBI-SS terminal session in windows mode

1 Log on to a BBI-SS terminal session in windows mode and request the CREGAGT view.

2 Type INI on the line for the target CICS.

3 Press Enter.

The INI line command initiates MainView for CICS for static targets only. Dynamic targets must be initialized from the target CICS. For more information on dynamic targets, refer to “BBI-SS PAS and CICS target connection” on page 59.

Terminating MainView for CICS

Terminating MainView for CICS stops all running functions and agents, completely removes the MainView for CICS resources from the target CICS, and stops BBI-SS PAS-to-CICS communication.

To terminate MainView for CICS from a CICS terminal

1 Log on to a CICS terminal.

2 Type FST2 TERM.

3 Press Enter.
To terminate MainView for CICS from a z/OS console that is defined to CICS, use the MODIFY command to invoke the FST2 transaction by using the TERM option. For example, the following command terminates MainView for CICS in the CICS region with the z/OS job name of CICSPROD:

F CICSPROD,FST2 TERM

To terminate MainView for CICS from a BBI-SS terminal session in windows mode

1. Log on to a BBI-SS terminal session in windows mode and request the CREGAGT view.
2. Type TRM on the line for the target CICS.
3. Press Enter.

Automatic termination and initialization of MainView for CICS

When the BBI-SS PAS is terminated, all connected target CICS systems are also terminated—all active functions are stopped and the MainView for CICS resources are removed from each target CICS.

When the BBI-SS PAS is restarted, all previously connected target CICS systems are re-initialized with MainView for CICS and all previously active functions restarted. This re-initialization of MainView for CICS occurs over a COLD or a WARM start of the BBI-SS PAS.

Starting BBI-SS PAS-to-CICS communication

If you modified the PLT, BBI-SS PAS-to-CICS communication starts automatically at CICS startup.

If you did not modify the PLT, you can manually start BBI-SS PAS-to-CICS communication from a CICS terminal or a z/OS console that is defined to CICS.

To start BBI-SS PAS-to-CICS communication from a CICS terminal

1. Log on to a CICS terminal.
2 Start the FST2 transaction by typing `FST2`.

3 Press `Enter`.

To start BBI-SS PAS-to-CICS communication from a z/OS console that is defined to CICS, use the MODIFY command to invoke the FST2 transaction in the CICS region. For example, the following command invokes the FST2 transaction in the CICS region with the z/OS job name of CICSPROD:

```
F CICSPROD,FST2
```

**Stopping BBI–SS PAS–to–CICS communications**

After the BBI-SS PAS-to-CICS communication link is started, it can be stopped through a request or a command.

The BBI-SS PAS-to-CICS communication link is stopped and restarted automatically when the BBI-SS PAS is terminated and restarted. The BBI-SS PAS-to-CICS communication link is also stopped during the processing of a TERM request. The TERM request completely removes the MainView for CICS product from the target CICS system. For more information about the TERM request, refer to “Commands available with the CREGAGT view” on page 268.

**Starting MainView for CICS monitoring**

If you modified the PLT, MainView for CICS monitoring starts automatically at CICS startup.

If you did not modify the PLT, or monitoring was stopped after CICS startup, you can start monitoring manually from a CICS terminal, a z/OS console that is defined to CICS, or the CREGAGT view in a BBI-SS terminal session in windows mode.

*Note*

Unless otherwise specified, a request to start MainView for CICS monitoring will start BBI-SS PAS-to-CICS communication if it is not already started.
To start MainView for CICS monitoring from a CICS terminal if communications is not started

1. Log on to a CICS terminal.
2. Type `FST2 QON`.
3. Press Enter.

To start MainView for CICS monitoring from a CICS terminal if communications is already started

1. Log on to a CICS terminal.
2. Type `FST2 QON`.
3. Press Enter.

or

1. Log on to a CICS terminal.
2. Type `SMN2`.
3. In the Extractor Status field, type ON.
4. Press Enter.

To start MainView for CICS monitoring from a z/OS console that is defined to CICS, use the MODIFY command to invoke the FST2 transaction with the QON option. This process is valid regardless of whether or not the BBI-SS PAS to CICS communication is started.

For example, the following command starts BBI-SS PAS-to-CICS communications (if it is not already started) and turns on MainView for CICS monitoring in the CICS region with the z/OS job name of CICSPROD:

```
F CICSPROD,FST2 QON
```

To start MainView for CICS monitoring from a BBI-SS terminal session in windows mode

1. Log on to a BBI-SS terminal session in windows mode and request the CREGAGT view.
2. Type QON on the line for the target CICS.
3. Press Enter.
If BBI-SS PAS-to-CICS communication is active, the QON line command starts MainView for CICS monitoring for both static and dynamic targets. If BBI-SS PAS-to-CICS communications is not started, QON starts BBI-SS PAS-to-CICS communication and MainView for CICS monitoring for static targets only. The BBI-SS PAS-to-CICS communication for a dynamic target must be initiated from the target CICS. For more information about dynamic targets, refer to “BBI-SS PAS and CICS target connection” on page 59.

**Stopping MainView for CICS monitoring**

Use the following procedures to stop MainView for CICS monitoring.

**To stop MainView for CICS monitoring from a CICS terminal**

1. Log on to a CICS terminal.
2. Type `FST2 QOFF`.
3. Press Enter.

**To stop MainView for CICS monitoring from a CICS terminal**

1. Log on to a CICS terminal.
2. Type `SMN2`.
3. In the Extractor Status field, type `OFF`.
4. Press Enter.

To stop MainView for CICS monitoring from a z/OS console that is defined to CICS, use the MODIFY command to invoke the FST2 transaction with the QOFF option. For example, the following command turns off MainView for CICS monitoring in the CICS region with the z/OS job name of CICSPROD:

```
F CICSPROD,FST2 QOFF
```

**To stop MainView for CICS monitoring from a BBI-SS terminal session in windows mode**

1. Log on to a BBI-SS terminal session in windows mode and request the CREGAGT view.
2. Type `QOF` on the line for the target CICS.
3 Press Enter.

Starting the task kill function

The topic provides information about starting the task kill function.

If you specified TASKKILL=YES in the CMRSOPT option table (described on “Starting the task kill function” on page 160), the task kill function is started automatically at CICS startup either at PLT or on the first MainView for CICS INIT request.

You can also control the task kill function manually by issuing the FST2 transaction from a CICS terminal, a z/OS console that is defined to CICS, or from the CREGAGT view in a BBI-SS terminal session in windows mode.

To start the task kill function from a CICS terminal

1 Log on to a CICS terminal.

2 Type FST2 KON.

3 Press Enter.

To start the task kill function from a z/OS console that is defined to CICS, use the MODIFY command to invoke the FST2 transaction with the KON option in the CICS region. This command is valid regardless of whether or not the BBI-SS PAS-to-CICS communication is started. For example, the following command starts BBI-SS PAS-to-CICS communications (if it is not already started) and starts the task kill function in the CICS region with the z/OS job name of CICSPROD:

F CICSPROD,FST2 KON

To start the task kill function from a BBI-SS terminal session in windows mode

1 Log on to a BBI-SS terminal session in windows mode and request the CREGAGT view.

2 Type KON on the line for the target CICS.

3 Press Enter.

Note

If the task kill exit is active and you change the task kill thresholds by using the CMRTTHthr view, use the KRLD command to reload the CMRTTHnn table.

Appendix D Controlling functions manually 235
Stopping the task kill function

Use the following procedures to stop the task kill function.

To stop the task kill function from a CICS terminal

1 Log on to a CICS terminal.
2 Type FST2 KOFF.
3 Press Enter.

To stop the task kill function from a z/OS console that is defined to CICS, use the MODIFY command to invoke the FST2 transaction with the KOFF option in the CICS region. For example, the following command stops the task kill function in the CICS region with the z/OS job name of CICSPROD:

F CICSPROD,FST2 KOFF

To stop the task kill function from a BBI-SS terminal session in windows mode

1 Log on to a BBI-SS terminal session in windows mode and request the CREGAGT view.
2 Type KOF on the line for the target CICS.
3 Press Enter.

Reloading the task kill function

Use the following procedures to reload the task kill function.

To reload the task kill function from a CICS terminal

1 Log on to a CICS terminal.
2 Type FST2 KRLD.
3 Press Enter.

To reload the task kill function from a z/OS console that is defined to CICS, use the MODIFY command to invoke the FST2 transaction with the KRLD option in
the CICS region. For example, the following command reloads the task kill function in the CICS region with the z/OS job name of CICSPROD:

```
F CICSPROD,FST2 KRLD
```

**To reload the task kill function from a BBI-SS terminal session in windows mode**

1. Log on to a BBI-SS terminal session in windows mode and request the CREGAGT view.

2. Type `KRL` on the line for the target CICS.

3. Press `Enter`.

---

**Starting the program timing facility**

Use the following procedures to start the program timing facility.

**To start the Program Timing facility from a CICS terminal**

1. Log on to a CICS terminal.

2. Type `FST2 PGON`.

3. Press `Enter`.

To start the Program Timing facility from a z/OS console that is defined to CICS, use the MODIFY command to invoke the FST2 transaction with the PGON option in the CICS region. For example, the following command starts the facility in the CICS region with the z/OS jobname of CICSPROD:

```
F CICSPROD,FST2 PGON
```

**To start the Program Timing facility from a BBI-SS terminal session in windows mode**

1. Log on to a BBI-SS terminal session in windows mode and request the CREGAGT view.

2. Type `PGY` on the line for the target CICS.

3. Press `Enter`.
Stopping the program timing facility

Use the following procedures to stop the program timing facility.

**To stop the Program Timing facility from a CICS terminal**

1. Log on to a CICS terminal.
2. Type `FST2 PGOF`.
3. Press `Enter`.

To stop the Program Timing facility from a z/OS console that is defined to CICS, use the MODIFY command to invoke the FST2 transaction with the PGOF option in the CICS region. For example, the following command stops the facility in the CICS region with the z/OS jobname of CICSPROD:

```
F CICSPROD,FST2 PGOF
```

**To stop the Program Timing facility from a BBI-SS terminal session in windows mode**

1. Log on to a BBI-SS terminal session in windows mode and request the CREGAGT view.
2. Type `PGN` on the line for the target CICS.
3. Press `Enter`.

Starting and stopping session and connection monitoring

This function can be controlled with the FST2 transaction or the CREGAGT view:

- CSON activates the session\connection data collection (FST2 only)
- CSOFF deactivates the session\connection data collection (FST2 only)
- CSY activates the session\connection data collection (FST2 and CREGAGT)
- CSN deactivates the session\connection data collection (FST2 and CREGAGT)

The following messages indicate the request’s status:
- FT3088I MainView for CICS session collection stopped
- FT3088I MainView for CICS session collection already active
- FT3088I MainView for CICS session collection not started-extractor inactive
- FT3088I MainView for CICS session collection not started-file expansion off
- FT3088I MainView for CICS session collection not started--ENABLEs failed
- FT3088I MainView for CICS session collection stopped but DISABLEs failed
- FT3088I MainView for CICS session collection already inactive
- FT3088I MainView for CICS session collection started

These messages are written to the CICS job log, z/OS console and BBI journal. The data extractor must be active for the session data collection, and file expansion must be on.

### Starting and resetting CICS Transaction Gateway monitoring

Once CTG monitoring is active, you can reset the current monitoring parameters by using the RESET command with the PARM option from a z/OS console, or with the LOG display.

To reset the parameters from a z/OS console, use the MODIFY command:

```
F pasname,RESET PARM,CMRCTG
```

`pasname` is the BBI-SS PAS stcname.

Use the following command to reset the parameters from the LOG display:

```
.RESET PARM CMRCTG
```

When the command is issued, the current CTG parameter values are refreshed with the values in the parameter file, CMRBEXnn. See the *MainView Administration Guide* for more information about the RESET command and its parameters.
Starting and stopping transient data (TD) queue monitoring

Transient data (TD) queue monitoring can be controlled with the FST2 transaction or the CREGAGT view:

- TDON activates the transient data queue collection (FST2 only)
- TDOF deactivates the transient data queue collection (FST2 only)
- TDY activates the transient data queue collection (FST2 and CREGAGT)
- TDN deactivates the transient data queue collection (FST2 and CREGAGT)

The following messages indicate the request’s status:

- FT3089I MainView for CICS TD queue collection started
- FT3089I MainView for CICS TD queue collection stopped
- FT3089I MainView for CICS TD queue collection already active
- FT3089I MainView for CICS TD queue collection already inactive
- FT3089I MainView for CICS TD queue collection not started-Extractor inactive
- FT3089I MainView for CICS TD queue collection not started-file expansion off

These messages are written to the CICS job log, z/OS console, and BBI journal. The data Extractor must be active, and file expansion must be on for transient data queue collection.
Starting and stopping temporary storage (TS) queue monitoring

Temporary storage (TS) queue monitoring can be controlled with the FST2 transaction or the CREGAGT view:

- TSON activate the temporary storage queue collection (FST2 only)
- TSOF deactivate the temporary storage queue collection (FST2 only)
- TSY activate the temporary storage queue collection (FST2 and CREGAGT)
- TSN deactivate the temporary storage queue collection (FST2 and CREGAGT)

The following messages indicate the request's status:

- FT3089I MainView for CICS TS queue collection started
- FT3089I MainView for CICS TS queue collection stopped
- FT3089I MainView for CICS TS queue collection already active
- FT3089I MainView for CICS TS queue collection already inactive
- FT3089I MainView for CICS TS queue collection not started-Extractor inactive
- FT3089I MainView for CICS TS queue collection not started-file expansion off

These messages are written to the CICS job log, z/OS console, and BBI journal. The data Extractor must be active, and file expansion must be on for transient data queue collection.
Resource descriptions and conflicts

This appendix describes the transactions and programs that are used by MainView for CICS and MainView AutoOPERATOR for CICS. It also includes zaps that can be used in the case of name conflicts.

Table 23: Common transactions

<table>
<thead>
<tr>
<th>Tran ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCE</td>
<td>is started by the CICS External Interface on behalf of a request from the BBI-SS. This transaction is not created dynamically.</td>
</tr>
<tr>
<td>FST2</td>
<td>controls the communications, the data collection and other agent functions within the CICS. For more information about the available functions, see “FST2 transaction” on page 263. The FST2 transaction is not created dynamically. <strong>Note:</strong> All activate functions will start BBI-SS communications if not already started.</td>
</tr>
<tr>
<td>FCD2</td>
<td>performs an action request that uses CICS services for completion. It is scheduled once for each request to ensure proper serialization.</td>
</tr>
<tr>
<td>JNL2</td>
<td>monitors the status of the BBI-SS PAS and starts transaction FCD2 when CICS action requests are processed. JNL2 is always shown as active on the TASK display, but most of that time is spent waiting (user ECB). When JNL2 starts, it remains active because it must be present in the system at all times after communication with the BBI-SS PAS is established. JNL2 is assigned a very high internal dispatching priority.</td>
</tr>
<tr>
<td>SMN2</td>
<td>allows manual starting of communications and data collection components. It also allows manual stopping of data collection. Communications, however, remain active once started. It schedules FST2 (FST2 QON, FST2 QOFF) to perform a service and reports on the status of the components.</td>
</tr>
</tbody>
</table>
Table 24: MainView for CICS transactions

<table>
<thead>
<tr>
<th>Tran ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCRT</td>
<td>For CICS 4.1 and later, the long-running transaction BCRT processes GET and SET requests that require CICS SPI commands to collect data and process modifications. For CICS Transaction Server 2.x and later, it also manages, on an interval basis, the queues that control the allocation and reuse of the MainView for CICSTIE (Task Interface Element) space that is used by the data extractor. For user security purposes and other monitoring exits, BCRT should be managed like transaction JNL2.</td>
</tr>
<tr>
<td>FIC2</td>
<td>captures and records MainView for CICS statistical data</td>
</tr>
<tr>
<td></td>
<td>FIC2 runs once each minute unless specified not to run by the FIC2=NO option in CMRSOPnn. FIC2 will not run in a CPSM CMAS or WUI region, no matter what is specified in CMRSOPnn.</td>
</tr>
</tbody>
</table>
Changing the FST2 or BMCE transactions requires the following actions:

1. The CSD definition for the transaction has to be changed and installed. Refer to BBSAMP member CSDOLTDF.

2. If the BBI-SS PAS has static targets, restart it with a cold start; otherwise, a warm start is sufficient.

   **Note**

   While the transaction IDs may change in CICS, the original transaction IDs still appear in online Help. BBSAMP member OLTUMOD0 includes a sample USERMOD to apply the zap by using SMP/E.

---

**Program descriptions**

This section lists the programs that are used by MainView for CICS and MainView AutoOPERATOR for CICS.

Program names that end in a lowercase x are CICS release dependent. The release dependency is provided in the following tables.

### Table 27: CMR and OLT programs

<table>
<thead>
<tr>
<th>Suffix</th>
<th>CICS release</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>CICS Transaction Server 3.1 only</td>
</tr>
<tr>
<td>8</td>
<td>CICS Transaction Server 3.2 only</td>
</tr>
<tr>
<td>4</td>
<td>CICS Transaction Server 4.1 only</td>
</tr>
<tr>
<td>7</td>
<td>CICS Transaction Server 4.2 only</td>
</tr>
<tr>
<td>6</td>
<td>CICS Transaction Server 5.1 only</td>
</tr>
</tbody>
</table>
**Table 28: BBK programs**

<table>
<thead>
<tr>
<th>Suffix</th>
<th>CICS release</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>CICS Transaction Server 3.1 only</td>
</tr>
<tr>
<td>8</td>
<td>CICS Transaction Server 3.2 only</td>
</tr>
<tr>
<td>4</td>
<td>CICS Transaction Server 4.1 only</td>
</tr>
<tr>
<td>7</td>
<td>CICS Transaction Server 4.2 only</td>
</tr>
<tr>
<td>6</td>
<td>CICS Transaction Server 5.1 only</td>
</tr>
</tbody>
</table>

**Table 29: Common programs**

<table>
<thead>
<tr>
<th>Program</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMRINT2</td>
<td>CICS products internal table</td>
</tr>
<tr>
<td>CMRFCMDx</td>
<td>CICS products action service program</td>
</tr>
<tr>
<td>OLTJRNLx</td>
<td>CICS products BBI-SS PAS communications program</td>
</tr>
<tr>
<td>CMRTMAP</td>
<td>CICS products interactive startup BMS mapset</td>
</tr>
<tr>
<td>CMRTMON</td>
<td>CICS products interactive startup program</td>
</tr>
<tr>
<td>CSLOADC</td>
<td>CICS products common service program</td>
</tr>
<tr>
<td>CSLOAD</td>
<td>CICS products common service program</td>
</tr>
<tr>
<td>OLTFCSET</td>
<td>CICS products PLTPI and initialization program</td>
</tr>
<tr>
<td></td>
<td>This program is not created dynamically.</td>
</tr>
<tr>
<td>OLTCNTL</td>
<td>CICS products release and product verification program</td>
</tr>
<tr>
<td>OLTVER</td>
<td>determines the OLTP release level</td>
</tr>
<tr>
<td>OLTFCSETx</td>
<td>CICS products general agent command processor and initialization program</td>
</tr>
<tr>
<td>OLTCCRE</td>
<td>common resource definition create program</td>
</tr>
<tr>
<td>BALPUTLV</td>
<td>PUT maintenance level verification</td>
</tr>
</tbody>
</table>

**Table 30: MainView for CICS programs**

<table>
<thead>
<tr>
<th>Program</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMRVCMR</td>
<td>verification of the presence of MainView for CICS</td>
</tr>
<tr>
<td></td>
<td>This program is not created dynamically.</td>
</tr>
<tr>
<td>CMR9RLSE</td>
<td>release identification table</td>
</tr>
<tr>
<td>BBK1CCRE</td>
<td>MainView for CICS resource definition create program</td>
</tr>
<tr>
<td>BBK1CRTR</td>
<td>EXEC CICS and TIE space control program</td>
</tr>
<tr>
<td>Program</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>BBK1RE3x</td>
<td>CREGSY* data collector</td>
</tr>
<tr>
<td>BBK1RE5x</td>
<td>CJVMPL data collector</td>
</tr>
<tr>
<td>BBK1TBAx</td>
<td>CJVMSRV data collector</td>
</tr>
<tr>
<td>BBK1TBEx</td>
<td>CENQMDL data collector</td>
</tr>
<tr>
<td>BBK1TBFx</td>
<td>CTSMODL data collector</td>
</tr>
<tr>
<td>BBK1TB1x</td>
<td>CDB2CON data collector</td>
</tr>
<tr>
<td>BBK1TB2x</td>
<td>CDB2ENT data collector</td>
</tr>
<tr>
<td>BBK1TB3x</td>
<td>CDB2TRN data collector</td>
</tr>
<tr>
<td>BBK1TB4x</td>
<td>CLIBR data collector</td>
</tr>
<tr>
<td>BBK1TB5x</td>
<td>CREGTDQ data collector</td>
</tr>
<tr>
<td>BBK1TB8x</td>
<td>CMQCONN data collector</td>
</tr>
<tr>
<td>BBK1TEAx</td>
<td>CDUMPCD data collector</td>
</tr>
<tr>
<td>BBK1TEDx</td>
<td>CWEBSRV data collector</td>
</tr>
<tr>
<td>BBK1TEEx</td>
<td>CPIPLNE data collector</td>
</tr>
<tr>
<td>BBK1TEFx</td>
<td>CURIMAP data collector</td>
</tr>
<tr>
<td>BBK1TE2x</td>
<td>CLSRPOL data collector</td>
</tr>
<tr>
<td>BBK1TE3x</td>
<td>CTCP srv data collector</td>
</tr>
<tr>
<td>BBK1TE4x</td>
<td>CTDQ data collector</td>
</tr>
<tr>
<td>BBK1TE5x</td>
<td>CDOCTMP data collector</td>
</tr>
<tr>
<td>BBK1TE6x</td>
<td>CIPCONN data collector</td>
</tr>
<tr>
<td>BBK1TE9x</td>
<td>TSQUECF data collector</td>
</tr>
<tr>
<td>CMRTRUEx</td>
<td>task-related user exit program</td>
</tr>
<tr>
<td>CMRTRUEx</td>
<td>task-related user exit program stub</td>
</tr>
<tr>
<td>CMRKCPXx</td>
<td>data collection global exit program</td>
</tr>
<tr>
<td>CMRCMPXx</td>
<td>monitoring task end global exit program</td>
</tr>
<tr>
<td>CMRXEIOx</td>
<td>task kill function processor</td>
</tr>
<tr>
<td>CMRXEIST</td>
<td>task kill function control</td>
</tr>
<tr>
<td>CMRROLLx</td>
<td>statistics output program</td>
</tr>
<tr>
<td>CMRSLOGx</td>
<td>statistics gathering program</td>
</tr>
<tr>
<td>CMRXTFG</td>
<td>extended master control block</td>
</tr>
<tr>
<td>OLTSCMRx</td>
<td>start and stop data collection control</td>
</tr>
</tbody>
</table>
### Table 31: MainView AutoOPERATOR for CICS programs

<table>
<thead>
<tr>
<th>Program</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLTVCACO</td>
<td>verifies presence of MainView AutoOPERATOR</td>
</tr>
<tr>
<td></td>
<td>This program is not created dynamically.</td>
</tr>
<tr>
<td>OLTSCAO</td>
<td>starts and stops the transient data exit</td>
</tr>
<tr>
<td></td>
<td>This program is not created dynamically until APAR BAO5427 is applied.</td>
</tr>
<tr>
<td>CBQCMD</td>
<td>broadcast program</td>
</tr>
<tr>
<td></td>
<td>This program is not created dynamically until APAR BAO5427 is applied.</td>
</tr>
<tr>
<td>CMRTDPXx</td>
<td>transient data global exit program</td>
</tr>
</tbody>
</table>
BBSAMP data set members

The BBSAMP data set contains sample macros, JCL, and user exit routines that you can copy and modify.

The first section describes common members that are shared by MainView for CICS and MainView AutoOPERATOR. The second section describes BBSAMP members that apply to only MainView for CICS.

Sample members for MainView AutoOPERATOR and MainView for CICS

This topic provides information about sample members for MainView AutoOPERATOR and MainView for CICS.

Table 32 on page 249 describes members of the BBSAMP data set that are used by both MainView AutoOPERATOR and MainView for CICS.

Table 32: Shared BBSAMP data set members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICSRACF</td>
<td>CLIST that provides an interactive dialog to build SAF resource definitions</td>
</tr>
<tr>
<td>CMRASM</td>
<td>sample JCL to assemble and link-edit any of the following modules:</td>
</tr>
<tr>
<td></td>
<td>■ CMRPRBT (MainView for CICS only)</td>
</tr>
<tr>
<td></td>
<td>■ CMRRAPM (MainView for CICS only)</td>
</tr>
<tr>
<td></td>
<td>■ CMRSECU (MainView for CICS and MainView AutoOPERATOR)</td>
</tr>
<tr>
<td></td>
<td>■ CMRSOPT (MainView for CICS only)</td>
</tr>
<tr>
<td></td>
<td>Descriptions of these modules can be found in this table and in Table 33 on page 250.</td>
</tr>
<tr>
<td>Member name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>CMRPLT</td>
<td>macro used to generate correct PLT entries for all CICS regions. This macro is used when the auto maintenance facility is not used.</td>
</tr>
<tr>
<td>CMRSECU</td>
<td>macro that assembles and links the CMRSECU module and creates a table of security definitions</td>
</tr>
<tr>
<td>CMRUSECU</td>
<td>sample to define function security</td>
</tr>
</tbody>
</table>

## Sample members for MainView for CICS

This topic provides information about sample members for MainView for CICS.

Table 33 on page 250 describes members of the BBSAMP data set that are used only by MainView for CICS.

### Table 33: MainView for CICS BBSAMP data set members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BDR#DEF</td>
<td>sample history parameters (CHIST and related views)</td>
</tr>
<tr>
<td>CMR$DATA</td>
<td>assembler DSECT for the MainView for CICS data included in the CICS SMF 110 record</td>
</tr>
<tr>
<td>CMR$DB2</td>
<td>assembler DSECT for the MainView for CICS DB2 data included in the CICS SMF 110 record</td>
</tr>
<tr>
<td>CMR$DMCT</td>
<td>sample Monitor Control Table entry to collect MainView for CICS data for inclusion in the CICS SMF 110 record</td>
</tr>
<tr>
<td>CMR$DSAS</td>
<td>SAS definitions for the MainView for CICS data included in the CICS SMF 110 record</td>
</tr>
<tr>
<td>CMRADAX</td>
<td>sample exit for Adabas version 5 to capture MainView for CICS data after a database call</td>
</tr>
<tr>
<td>CMRADBX</td>
<td>sample exit for Adabas version 5 to capture MainView for CICS before a database call</td>
</tr>
<tr>
<td>CMRBARCC</td>
<td>sample JCL to copy a trace log data set to a sequential data set for archival</td>
</tr>
<tr>
<td>CMRBRLOD</td>
<td>sample JCL to reload an archived copy of a trace to a new trace log data set for online viewing</td>
</tr>
<tr>
<td>CMRCBxxN</td>
<td>counterpart to the CMRCOBBxx sample programs; same processing but for the F7 format record</td>
</tr>
<tr>
<td>CMRCHRT</td>
<td>sample batch report JCL and control statements</td>
</tr>
<tr>
<td>CMRCJCCL</td>
<td>sample JCL program used to decompress detail records</td>
</tr>
<tr>
<td>CMRCOBCN</td>
<td>COBOL format of the F7 history record (counterpart to CMRCOBCP)</td>
</tr>
<tr>
<td>Member name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CMRCOBCP</td>
<td>sample VS/COBOL II copybook containing CMRDETL record definitions for CMRFRED program communication</td>
</tr>
<tr>
<td>CMRCOBJ1</td>
<td>sample JCL to compile and execute VS/COBOL II programs</td>
</tr>
<tr>
<td>CMRCOBJ2</td>
<td>sample JCL to compile and execute VS/COBOL II programs</td>
</tr>
<tr>
<td>CMRCOB nn</td>
<td>VS/COBOL II batch program samples for generating CICS performance statistics reports</td>
</tr>
<tr>
<td>CMRCONV</td>
<td>sample JCL to run the date conversion program for year 2000 support</td>
</tr>
<tr>
<td>CMRCOPY</td>
<td>Files that are produced by previous releases of MainView for CICS must be converted before the data can be processed by version 5.2 or later.</td>
</tr>
<tr>
<td>CMRCSDAL</td>
<td>JCL used to create the CSD resource groups used by MainView for CICS or MainView AutoOPERATOR</td>
</tr>
<tr>
<td>CMRCSDEL</td>
<td>JCL used to delete resources from the BOOLEPPT and BOOLEPCT resource groups</td>
</tr>
<tr>
<td>CMRCSDL A</td>
<td>JCL used to modify the group list that contains the resource groups installed at CICS initialization (MainView AutoOPERATOR only)</td>
</tr>
<tr>
<td>CMRCSDLB</td>
<td>JCL used to modify the group list that contains the resource groups installed at CICS initialization (MainView AutoOPERATOR and MainView for CICS)</td>
</tr>
<tr>
<td>CMRCSDL M</td>
<td>JCL used to modify the group list that contains the resource groups installed at CICS initialization (MainView for CICS only)</td>
</tr>
<tr>
<td>CMRD D TL</td>
<td>sample JCL to delete and define CMRDETL VSAM data sets</td>
</tr>
<tr>
<td>CMRDETL N</td>
<td>PRL format of the F7 history record (counterpart to CMRDETL)</td>
</tr>
<tr>
<td>CMRDFS U M</td>
<td>sample JCL to delete and define the VSAM data sets or sequential data sets that hold summarized records created by the CMRSUMD utility</td>
</tr>
<tr>
<td>CMRD J CL</td>
<td>sample JCL to archive data from the CMRDETL data set</td>
</tr>
<tr>
<td>CMRDL DF</td>
<td>sample JCL to delete and define CMRDETL VSAM files and load the initial record into CMRDETL</td>
</tr>
<tr>
<td>CMRDPARM</td>
<td>sample CMRPURG parameters</td>
</tr>
<tr>
<td>CMR L</td>
<td>sample JCL to produce PRL reports</td>
</tr>
<tr>
<td>CMRL D TL</td>
<td>sample JCL used by the CMRDETL reload utility--CMRLD TL</td>
</tr>
<tr>
<td>CMRMV J CL</td>
<td>sample JCL to produce PRL reports using multiple VSAM data sets</td>
</tr>
<tr>
<td>CMR N2M CT</td>
<td>MCT definition for MainView for CICSDB2 SMF data</td>
</tr>
<tr>
<td>CMR N2 SAS</td>
<td>SAS definition of the MainView for CICSDB2 SMF data</td>
</tr>
<tr>
<td>CM RN D TA</td>
<td>assembler definition of the MainView for CICSSMF data format</td>
</tr>
<tr>
<td>CMR NDB2</td>
<td>assembler definition of the MainView for CICSDB2 SMF data</td>
</tr>
<tr>
<td>CMR N D M CT</td>
<td>MCT definition for MainView for CICSSMF data</td>
</tr>
</tbody>
</table>

Appendix F  BBSAMP data set members 251
<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMRNDASAS</td>
<td>SAS definition of the MainView for CICS SMF data format</td>
</tr>
<tr>
<td>CMRPRBTT</td>
<td>macro to generate the problem threshold definitions table</td>
</tr>
<tr>
<td>CMRPRL n</td>
<td>various examples of Performance Reporting Language (PRL) control statements to generate reports</td>
</tr>
<tr>
<td>CMRPURGE</td>
<td>sample JCL to purge and merge recorder file data</td>
</tr>
<tr>
<td>CMRPX*</td>
<td>sample PRL programs to produce CMRSTATS reports</td>
</tr>
<tr>
<td>CMRQT6E</td>
<td>assembler format of the F7 history record</td>
</tr>
<tr>
<td>CMRRAP</td>
<td>sample JCL to generate the Resource Analysis Report</td>
</tr>
<tr>
<td>CMRRAPM</td>
<td>macro used to define the resource analysis table used by CMRRAPR and CMRCHRT</td>
</tr>
<tr>
<td>CMRREPRO</td>
<td>sample JCL to copy VSAM files to temporary files prior to processing by the CMRCONV date conversion program</td>
</tr>
<tr>
<td>CMRS2AX</td>
<td>sample exit for SYS2K version 12.0 to capture MainView for CICS activity after a database call</td>
</tr>
<tr>
<td>CMRS2BX</td>
<td>sample exit for SYS2K version 12.0 to capture MainView for CICS activity before a database call</td>
</tr>
<tr>
<td>CMRSASxx</td>
<td>various SAS programs to read CMRDETL data and create reports</td>
</tr>
<tr>
<td>CMRSAxxN</td>
<td>counterpart to the CMRSASxx sample programs; same processing but for the F7 format record</td>
</tr>
<tr>
<td>CMRSITR</td>
<td>sample PRL statements to generate the System Initialization Table (SIT) report</td>
</tr>
<tr>
<td>CMRSJCL</td>
<td>sample JCL used to archive data from single CMRDETL data set</td>
</tr>
<tr>
<td>CMRSMLXT</td>
<td>sample SMLXT</td>
</tr>
<tr>
<td>CMRSOPT</td>
<td>sample macro that specifies monitored resources, collection intervals, and response time service levels</td>
</tr>
<tr>
<td>CMRSPARM</td>
<td>sample CMRPURG parameters for single CMRDETL data set</td>
</tr>
<tr>
<td>CMRSTATX</td>
<td>sample JCL to process type 110 records stored in the SMF data set prior to their use by the CICS statistics program, DFHSTUP, and other products that use the same data CMRSTATX removes MainView for CICS records that are added to the SMF data set.</td>
</tr>
<tr>
<td>CMRSTJCL</td>
<td>sample JCL to run CMRSTATS to convert CICS statistical records to CMRSTATS format</td>
</tr>
<tr>
<td>CMRSUMD</td>
<td>sample program to create four workloads of daily summary records</td>
</tr>
<tr>
<td>CMRSUMDx</td>
<td>sample CMRSUMD control statements to create summary records for various intervals</td>
</tr>
<tr>
<td>CMRSUP01</td>
<td>sample pre- and post-call exit points for monitoring SUPRA version 1.3</td>
</tr>
<tr>
<td>CMRSUP02</td>
<td>sample pre- and post-call exit points for monitoring SUPRA version 2.4 or later</td>
</tr>
<tr>
<td>CMRSUPRX</td>
<td>sample exit for monitoring SUPRA version 2.4 or later</td>
</tr>
<tr>
<td>CMRTBTRA</td>
<td>sample JCL to print trace log data sets</td>
</tr>
<tr>
<td>Member name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>CMRTTHHR</td>
<td>macro to convert the task kill exit threshold table for use with the CMRTTHHR view</td>
</tr>
<tr>
<td>CMRUACMP</td>
<td>sample user exit to call the CMRCUEX program</td>
</tr>
<tr>
<td>CMRUAPLT</td>
<td>sample PLT program to start CMRUACMP</td>
</tr>
<tr>
<td>CMRUPRBT</td>
<td>sample to define the problem threshold table</td>
</tr>
<tr>
<td>CMRURAPM</td>
<td>sample to define the resource analysis table</td>
</tr>
<tr>
<td>CMRUSOPT</td>
<td>sample macro options that define the monitoring options table</td>
</tr>
<tr>
<td>CMRUSR01</td>
<td>sample of a user exit to switch CMRDETL data sets when one becomes full and needs to be archived</td>
</tr>
<tr>
<td>CMRUSRSD</td>
<td>sample exit called by CMRSUMD summarization utility to give control to a user-written routine when a CMRDETL record is processed</td>
</tr>
<tr>
<td>CSDCMRDF</td>
<td>contains the DEFINE statements for the required program and transaction resources for the MainView for CICS component. These definitions are for all releases of CICS</td>
</tr>
<tr>
<td>CSDOLTDF</td>
<td>contains the DEFINE statements required for the program and transaction resources used by the MainView for CICS common OLT component (for all releases of CICS)</td>
</tr>
<tr>
<td>DCCTXPR</td>
<td>sample exit to the DATACOM/DB DCCTXPR routine</td>
</tr>
<tr>
<td>GENEROL</td>
<td>sample user exit for inclusion in the GENER/OL user task program</td>
</tr>
<tr>
<td>LPCZZMON</td>
<td>sample modification of the MSA LPCZZMON exit</td>
</tr>
<tr>
<td>NCIIRDCX1</td>
<td>sample user exit to capture the Natural program name (for Natural version 2.2. n)</td>
</tr>
<tr>
<td>OLTUMOD0</td>
<td>sample USERMOD to zap transaction ID by using SMP/E</td>
</tr>
<tr>
<td>SAPCMROT</td>
<td>sample program to store SAP data in the T6E record of the CMRDETL data set. SAPCMROT captures SAP data by calls to the MainView for CICS exit, CMRCUEX, which is described in “User exit interface (CMRCUEX)” on page 255. SAPCMROT must be included in the SAP statistics exit (SAPSTEC) to pass data to MainView for CICS by a CMRCUEX call.</td>
</tr>
<tr>
<td>SAPCMROW</td>
<td>sample of the SAPSTEC exit called by the SAP system to pass information to MainView for CICS by a CMRCUEX call</td>
</tr>
<tr>
<td>SAPCMRTW</td>
<td>member that terminates previously initiated clocks as part of passing SAP data to MainView for CICS. SAPCMRTW must be included in the SAP statistics exit (SAPSTEC) to pass data to MainView for CICS by a CMRCUEX call.</td>
</tr>
<tr>
<td>SAPCMRWT</td>
<td>member that captures data about SAP work area waits. It issues a call to MainView for CICS to begin timing the wait. SAPCMRWT must be included in the SAP statistics exit (SAPSTEC) to pass data to MainView for CICS by a CMRCUEX call.</td>
</tr>
<tr>
<td>SASI6ExN</td>
<td>SAS formats for the F7 history record (counterpart to SASIN6E). Values for x are: C, P, Q, S, X, 1, and 2.</td>
</tr>
<tr>
<td>Member name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SASIN6E1</td>
<td>SAS sample to read MainView for CICS type 6E records. This sample selects all records that have at least one resource entry and selects only the DBCTL resource entries. It reads all fields (unexpanded and expanded) for all resource entries that are identified with a type of &quot;I&quot;.</td>
</tr>
<tr>
<td>SASIN6E2</td>
<td>SAS sample to read MainView for CICS type 6E records. This sample selects all records that have at least one resource entry and selects only the DB2 resource entries. It reads all fields (unexpanded and expanded) for all resource entries that are identified with a type of &quot;2&quot;.</td>
</tr>
<tr>
<td>SASIN6EC</td>
<td>SAS sample to read MainView for CICS type 6E records. This sample selects all records that have at least one resource entry and selects only the file resource entries. It reads all fields (unexpanded and expanded) for all resource entries that are identified with a type of &quot;C&quot; or &quot;P&quot;.</td>
</tr>
<tr>
<td>SASIN6EO</td>
<td>SAS sample to read MainView for CICS type 6E records. This sample reads and prints all data in the connection resource entries. It produces the data for all entries identified with a type of &quot;)&quot;.</td>
</tr>
<tr>
<td>SASIN6EP</td>
<td>SAS sample to read MainView for CICS type 6E records. This sample selects all records that have at least one resource entry and selects only the program timing resource entries. It reads all fields (unexpanded and expanded) for all resource entries that are identified with a type of &quot;!&quot;.</td>
</tr>
<tr>
<td>SASIN6EQ</td>
<td>SAS sample to read MainView for CICS type 6E records. This sample selects all records that have at least one resource entry and selects only the WebSphere MQ resource entries. It reads all fields (unexpanded and expanded) for all resource entries that are identified with a type of &quot;Q&quot;.</td>
</tr>
<tr>
<td>SASIN6ES</td>
<td>SAS sample to read MainView for CICS type 6E records. This sample selects all records that have at least one resource entry and selects only the session resource entries. It reads all fields (unexpanded and expanded) for all resource entries that are identified with a type of &quot;(&quot;.</td>
</tr>
<tr>
<td>SASIN6EX</td>
<td>SAS sample to read MainView for CICS type 6E records. This sample reads all fields in the standard fixed portions and all of the extension areas. This sample does not process any of the resource (file) entries.</td>
</tr>
<tr>
<td>SASINX nn</td>
<td>SAS control statements to process statistical and transaction records. nn is the record type as described in Appendix C of the MainView for CICS PERFORMANCE REPORTER User Guide.</td>
</tr>
<tr>
<td>SASINxx</td>
<td>(obsolete) SAS control statements to process statistical and transaction records</td>
</tr>
<tr>
<td>STATREC</td>
<td>Macro used by programs that reference CMRSTATS data</td>
</tr>
<tr>
<td>THRJCL</td>
<td>sample JCL and instructions for converting a CMRTTHR task kill threshold table</td>
</tr>
<tr>
<td>XNCIRDC1</td>
<td>sample user exit to capture the Natural program name (for Natural version 2.3.1 or later)</td>
</tr>
</tbody>
</table>
User exit interface (CMRCUEX)

The CMRCUEX user exit interface provides several facilities. You can:

- capture statistics from database products other than DBCTL and DB2 in greater detail than normally reported by MainView for CICS
- add user-defined information to transaction detail history records created by MainView for CICS
- replace the initial (PCT) program name with the name of a program running at a lower level, in the same way that MainView for CICS does for supported fourth-generation languages (4GLs)
- alter the netname or transaction name

Initiating database calls

CMRCUEX interfaces to the module that MainView for CICS uses to capture file statistics.

You can use the timing module to capture statistics that are not already captured by MainView for CICS.

For example, if you use a database other than DBCTL or DB2, you can track the time spent in calls to that database on a file name or request type (such as READ or WRITE) level as follows:

**To initiate database calls**

1. Call CMRCUEX with an I parameter to start a clock before executing the database command.

2. Call CMRCUEX with a T parameter to stop the clock after control is returned to your program.
Statistics that are captured by CMRCUEX are available in the FILE graphs, the HISTORY file expansion, and batch reports. The information is stored in resource segment fields T6EFlnn, T6EFNnn, T6EFTnn, and T6EFCnn on the CMRDETL file, where nn is the nnth resource segment in the T6E record.

The maximum number of occurrences of database and resource entries is controlled by the MAXFILE CMRSOPT parameter. The maximum includes MainView for CICS resources that are also being collected (CICS files, DB2 and DBCTL databases, program timing and session conversation data, and so on). When the maximum is reached, CMRCUEX ignores calls to start a clock for new database names and issues a return code of 4.

Adding user data

CMRCUEX also provides the ability to add identifying information to the transaction history detail record.

You can use this information in any way you choose; for example, you might identify transactions by application subsystem so that you can produce a report sorted by this field.

To request that CMRCUEX add user data to the history record, call CMRCUEX with one of the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>add up to 18 characters of data (for CICS TS 3.1 and later) The user data is stored in T6EUDATA on the CMRDETL file.</td>
</tr>
<tr>
<td>W</td>
<td>add up to 64 characters of data (for CICS TS 3.2 and later) The user data is stored in T6EUDAT2 on the CMRDETL file.</td>
</tr>
</tbody>
</table>

You can call CMRCUEX at any time before the end of the transaction.

Initiating program calls

MainView for CICS normally reports the program name that is known to CICS through the PCT.
CMRCUEX allows you to override that program name with the program name you prefer to track.

For instance, 4GL programs sometimes provide user exit points that make the name of the 4GL program actually executing available to the user. Including a call to CMRCUEX in the exit program can pass the program name to MainView for CICS as a substitute for the initial (PCT) program name. To do this substitution, call CMRCUEX with an I parameter to capture the specified program name. If multiple calls are issued, the last program name that was specified is used.

The program information is stored in fields T6EPGNM and T6EPTYPE in the CMRDETL file. Statistics captured for you by CMRCUEX are available in the PANEL graphs, the HISTORY file expansion, and batch reporting.

### Changing the netname or transaction name

MainView for CICS enables you to override the value for the netname or the transaction name with the value that you prefer to track in the MainView for CICS history data.

For a netname change, call CMRCUEX with the V parameter and the desired name. Netname information is stored in the T6ENETNM field in the history record. For a transaction name change, call CMRCUEX with the N parameter and the desired name. Transaction name information is stored in the T6ETRID field in the history record.

### CMRCUEX rules and conventions

CMRCUEX must be link-edited with the user program in the same load module.

Either allow the linkage editor to resolve the external reference to CMRCUEX (by including the BBLINK library in SYSLIB in the link-edit step) or use the linkage editor INCLUDE control statement to link CMRCUEX into the load module. The user program can be linked as AMODE(24) or AMODE(31). CMRCUEX puts itself in 31-bit mode as soon as it starts.

When you use the COBOL interface module, both CMRCCUEX and CMRCUEX must be link-edited in the calling program’s load module. CMRCCUEX calls CMRCUEX in a COBOL environment.
Restrictions

Several restrictions apply to request type identifiers:

- When timing database calls, MainView for CICS uses the following database IDs: A, C, D, G, I, K, P, S, 1, 2
  They should not be used for user-defined database types.

- When replacing program names, MainView for CICS uses these 4GL IDs: A, C, D, G, I, K, L, M, N, S, U
  They should not be used for user-defined program types.

Because MainView for CICS reserves the following identifiers, you should not use them as user-defined types:

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Used by</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>program CPU Timing</td>
</tr>
<tr>
<td>(</td>
<td>session data collection</td>
</tr>
<tr>
<td>)</td>
<td>connection data collection</td>
</tr>
<tr>
<td>\</td>
<td>TD queue data collection</td>
</tr>
<tr>
<td>/</td>
<td>TS queue data collection</td>
</tr>
<tr>
<td>&lt;</td>
<td>reserved</td>
</tr>
<tr>
<td>&gt;</td>
<td>reserved</td>
</tr>
</tbody>
</table>

Sample uses for CMRCUEX

This section provides examples of Assembler and COBOL statements that call CMRCUEX to:

- initiate a database call
- add user data
- replace a program name
- replace a netname
- replace a transaction name
Initiating a database call

This topic provides information about initiating a database call.

Figure 23 on page 259 and Figure 24 on page 259 are examples of Assembler and COBOL statements to initiate timing of a user-defined database call for a database identified as B.

A similar call, with the exception that CUEXCNTL must be T, is made at the completion of the user-defined database call. When control returns from CMRCUEX, all registers are restored except for registers 0, 1, and 15. Register 15 contains the return code.

Figure 23: Sample assembler statements to initiate a database call

MVI CUEXCNTL,C'I' Tell CMRCUEX to start clock
MVI CUEXRQST,C'D' Collect database stats
MVI CUEXTYPE,C'B' Identify this database
MVC CUEXNAME,=V(file name) May contain any identifying info
LA R1,CUEXPARM Pass the parameters to CMRCUEX
L R15,=V(CMRCUEX) Get CMRCUEX's addr from linkage
BALR R14,R15 Transfer control to CMRCUEX

**CUEXPARM**
- **DS** 0F
- **CUEXCNTL** DS C I=start clock, T=stop clock, U=user data
- **CUEXDATA** DS OCL18 Put user data here when CUEXCNTL=U
- **CUEXRQST** DS C 4=4GL, D=database
- **CUEXTYPE** DS C Identifies 4GL or DB (see restrictions above)
- **CUEXNAME** DS CL8 4GL program name or database file name or other identifying information
- **DS** CL8 Unused when starting or stopping clocks

To invoke CMRCUEX from a COBOL program, you must call CMRCCUEX rather than CMRCUEX. CMRCCUEX is an interface program that translates the parameter that is passing protocol from the one used by COBOL to the one expected by CMRCUEX. CMRCCUEX then invokes CMRCUEX. Figure 24 on page 259 shows an example of the processing.

Figure 24: Sample COBOL statements to initiate a database call

01 CMRCUEX-PARAMETERS.
  03 CUEX-INIT-TERM-OR-USER-CALL PIC X.
  03 CUEX-USER-DATA PIC X(18).
  03 CUEX-TIMING-CTL-FIELDS REDEFINES CUEX-USER-DATA.
  05 CUEX-4GL-OR-DB PIC X.
  05 CUEX-4GL-OR-DB-ID PIC X. (See restrictions above)
  05 CUEX-FILE-OR-PGM-NAME PIC X(8).
  05 FILLER PIC X(8).
  ------------
  MOVE 'I' TO CUEX-INIT-TERM-OR-USER-CALL
  MOVE 'D' TO CUEX-4GL-OR-DB
  MOVE 'B' TO CUEX-4GL-OR-DB-ID
  MOVE file name TO CUEX-FILE-OR-PGM-NAME
  CALL 'CMRCCUEX' USING CMRCUEX-PARAMETERS.
Adding data to the user data field

This topic provides information about adding data to the user data field.

Figure 25 on page 260 shows examples of using CMRCUEX to add data to user data fields.

**Figure 25: Sample statements to store user data**

In Assembler:

```
MVI CUEXCTL,C'U'      Indicate you want to store user data
MVC CUEXDATA,userdata  Give it your user data
LA   R1,CUEXCTL           Pass parameters to CMRCUEX
L    R15,=V(CMRCUEX)    Get addr of CMRCUEX from linkage editor
BALR R14,R15            Transfer control to CMRCUEX
```

In COBOL:

```
MOVE 'U' TO CUEX-INIT-TERM-OR-USER-CALL
MOVE userdata TO CUEX-USER-DATA
CALL 'CMRCCUEX' USING CMRCUEX-PARAMETERS.
```

**Note**

It is also possible to supply a field value for T6EUSER in the detail record. Refer to BBSAMP member GENEROL, which utilizes this facility.

Replacing program names

To replace a program name, call CMRCUEX at the time replacement information becomes available.

Figure 26 on page 260 shows examples of Assembler and COBOL statements to replace program names.

**Figure 26: Sample statements to replace a program name**

In Assembler:

```
MVICUEXCTL,C'I'      Must always be 'I'
MVICUEXRST,C'4'      Replace program name
MVCCUEXTYPE,C'J'     User-chosen identifying code
MVCCUEXNAME,pgmname  May contain any identifying info
LA     R1,CUEXPARM             Pass the parameters to CMRCUEX
L    R15,=V(CMRCUEX)    Get CMRCUEX's addr from linkage editor
BALR    R14,R15          Transfer control to CMRCUEX
```

In COBOL:

```
MOVE 'I' TO CUEX-INIT-TERM-OR-USER-CALL
MOVE '4' TO CUEX-4GL-OR-DB
MOVE 'J' TO CUEX-4GL-OR-DB-ID
MOVE pgm name TO CUEX-FILE-OR-PGM-NAME
CALL 'CMRCCUEX' USING CMRCUEX-PARAMETERS.
```
Replacing netnames

To replace a netname, call CMRCUEX at the time the replacement information becomes available.

**Figure 27: Sample statements to replace a netname**

In Assembler:

```
MVI   CUEXECNTL,C'V'      Indicate you want to replace a netname
MVC   CUEXDATA,netname   Give it your user data
LA    R1,CUEXECNTL            Pass parameters to CMRCUEX
L     R15,=V(CMRCUEX)    Get addr of CMRCUEX from linkage editor
BALR  R14,R15            Transfer control to CMRCUEX
```

In COBOL:

```
MOVE 'V' TO CUEX-INIT-TERM-OR-USER-CALL
MOVE netname TO CUEX-USER-DATA
CALL 'CMRCCUEX' USING CMRCUEX-parameters.
```

Replacing transaction names

To replace a transaction name, call CMRCUEX at the time the replacement information becomes available.

**Figure 28: Sample statements to replace a transaction name**

In Assembler:

```
MVI   CUEXECNTL,C'N'              Indicate you want to replace a transaction
MVC   CUEXDATA,transactionname   Give it your user data
LA    R1,CUEXECNTL                    Pass parameters to CMRCUEX
L     R15,=V(CMRCUEX)    Get addr of CMRCUEX from linkage editor
BALR  R14,R15                    Transfer control to CMRCUEX
```

In COBOL:

```
MOVE 'N' TO CUEX-INIT-TERM-OR-USER-CALL
MOVE tranid TO CUEX-USER-DATA
CALL 'CMRCCUEX' USING CMRCUEX-parameters.
```
FST2 transaction

The FST2 transaction is the MainView for CICS and MainView AutoOPERATOR for CICS control transaction.

It is used internally by MainView and you can also use it from a console defined to CICS or a terminal logged onto CICS. With it you can control:

- the MainView extractor
- the task kill agent
- the initialization and termination of the BBI-SS PAS communication
- starting and stopping MainView AutoOPERATOR for CICS
- switching the dual CMRDETL data sets immediately
- forcing the writing of the CMRDETL VSAM buffer
- controlling program CPU timing data collection
- controlling the setting of the SMF return code from the CICS XMNOUT global exit
- activating and deactivating session data collection
- activating and deactivating temporary storage queue usage data collection
- activating and deactivating transient data queue usage data collection

Table 34 on page 264 describes the available FST2 transaction commands and their function.

---

**Note**

If a command is not provided for the FST2 transaction, only BBI-SS communications are established.
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TERM, TRM</td>
<td>Terminates the MainView agents, records the MainView agent processing environment, and removes the MainView agents from the target CICS. Agents such as the extractor, task kill, and MainView AutoOPERATOR for CICS are terminated and their operating status is recorded in the BMCMVCTQ temporary storage queue. All MainView resources are discarded and released and communications with the BBI-SS PAS is terminated.</td>
</tr>
<tr>
<td>INIT, INI</td>
<td>Initializes the MainView agents by creating all resources, loading necessary programs, initiating communications with the BBI-SS PAS, and starting various functions, such as task kill, the extractor, and MainView AutoOPERATOR for CICS. The first time it is used in a target CICS a PLT-type initialization is performed (functions are started based on the presence of resources and settings in the CMRSOPT options table). Subsequent uses within the same running target CICS restores the environment that was recorded by the TERM process. If the CICS target is not connected to a BBI-SS PAS at the time of the TERM function, the subsequent INIT function initiates agents based on the CMRSOPT settings. The INIT command also accepts a 1-4 character name that identifies the BBI-SS PAS to which a communications connection is desired. Identification is specified immediately after the command (for example, FST2 INIT ssss will perform the initialization process and request a connection to the BBI-SS PAS identified by ssss).</td>
</tr>
<tr>
<td>blank</td>
<td>The FST2 transaction entered without a command operand initiates BBI-SS PAS communications. No other MainView agents are started.</td>
</tr>
<tr>
<td>QOFF, QOF</td>
<td>Shuts down the MainView for CICS extractor agent. The exit routines are disabled and the detail files are closed. No resources are discarded or released.</td>
</tr>
<tr>
<td>QON</td>
<td>Starts the MainView for CICS extractor agent.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>RES</td>
<td>Restarts MainView <em>for CICS</em> within the target CICS by performing a FST2 TERM followed by a FST2 INIT process. The TERM process terminates the MainView agents, records the MainView agent processing environment, and removes the MainView agents from the target CICS system. Agents such as the extractor, task kill, and MainView AutoOPERATOR <em>for CICS</em> are terminated and their operating status is recorded in the BMCMVCTQ temporary storage queue. All MainView resources are discarded and released, and communications with the BBI-SS PAS is terminated. The INIT process initializes the MainView agents by:</td>
</tr>
<tr>
<td></td>
<td>■ Creating all resources</td>
</tr>
<tr>
<td></td>
<td>■ Loading necessary programs</td>
</tr>
<tr>
<td></td>
<td>■ Initiating communications with the BBI-SS PAS</td>
</tr>
<tr>
<td></td>
<td>■ Starting various functions (such as task kill, the extractor, and MainViewAutoOPERATOR for CICS) This process restores the environment that the TERM process recorded. If MainView <em>for CICS</em> is already terminated when the operator issues the RES command, the FT920I message is issued and the INIT process is performed. The RES command does not support including a BBI-SS PAS ID for the INIT process.</td>
</tr>
<tr>
<td>SWI</td>
<td>Switch the dual CMRDETL data sets immediately.</td>
</tr>
<tr>
<td>FLU</td>
<td>Force (flush) the writing of the CMRDETL VSAM buffer.</td>
</tr>
<tr>
<td>KOFF, KOF</td>
<td>Terminates the MainView <em>for CICS</em> task kill agent.</td>
</tr>
<tr>
<td>KON</td>
<td>Starts the MainView <em>for CICS</em> task kill agent.</td>
</tr>
<tr>
<td>KRLD, KRL</td>
<td>Reloads the MainView <em>for CICS</em> task kill function. Resources are discarded, released, recreated, and the KOFF and KON functions are performed.</td>
</tr>
<tr>
<td>SYSM OFF, AOFF, AOF</td>
<td>Terminates the MainView AutoOPERATOR <em>for CICS</em> agent.</td>
</tr>
<tr>
<td>SYSM ON, AON</td>
<td>Starts the MainView AutoOPERATOR <em>for CICS</em> agent.</td>
</tr>
<tr>
<td>LOFF, LOF</td>
<td>Terminates the MainView <em>for CICS</em> long running task, BCRT. This command should not be used unless directed by BMC Software's MainView <em>for CICS</em> support.</td>
</tr>
<tr>
<td>LON</td>
<td>Starts the MainView <em>for CICS</em> long running task, BCRT. This command should not be used unless directed by BMC Software's MainView <em>for CICS</em> support.</td>
</tr>
<tr>
<td>PGY, PGON</td>
<td>Starts the MainView <em>for CICS</em> Program CPU Timing facility.</td>
</tr>
<tr>
<td>PGN, PGOF</td>
<td>Stops the MainView <em>for CICS</em> Program CPU Timing facility</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>CSY/ CSON</td>
<td>Activates session/connection activity recording</td>
</tr>
<tr>
<td>CSN/ CSOFF</td>
<td>Deactivates session/connection activity recording</td>
</tr>
<tr>
<td>PON</td>
<td>Sets the XMNOUT return code to initiate SMF writing of the CICS performance records.</td>
</tr>
<tr>
<td>POF/ POFF</td>
<td>Sets the XMNOUT return code to terminate SMF writing of the CICS performance records.</td>
</tr>
<tr>
<td>EON</td>
<td>Sets the XMNOUT return code to initiate SMF writing of the CICS exception records.</td>
</tr>
<tr>
<td>EOF/ EOFF</td>
<td>Sets the XMNOUT return code to terminate SMF writing of the CICS exception records.</td>
</tr>
<tr>
<td>RON</td>
<td>Sets the XMNOUT return code to initiate SMF writing of the CICS resource records.</td>
</tr>
<tr>
<td>ROF/ ROFF</td>
<td>Sets the XMNOUT return code to terminate SMF writing of the CICS resource records.</td>
</tr>
<tr>
<td>TDY/ TDON</td>
<td>Activates transient data (TD) queue activity recording</td>
</tr>
<tr>
<td>TDN/ TDOF</td>
<td>Deactivates transient data (TD) queue activity recording</td>
</tr>
<tr>
<td>TSY/ TSON</td>
<td>Activates temporary storage (TS) queue activity recording</td>
</tr>
<tr>
<td>TSN/ TSOF</td>
<td>Deactivates temporary storage (TS) queue activity recording</td>
</tr>
</tbody>
</table>

**Note**

The SMF related commands—PON, POF, EON, EOF, RON, ROF, CON, COF—modify the return code that is set by the MainView for CICS program enabled at the XMNOUT CICS global exit. The return code determines whether the record is written to the operating system SMF files. Other programs that are enabled at the XMNOUT global exit point after the MainView for CICS program, might also alter this return code.
The CREGAGT view displays the status of MainView for CICS, MainView AutoOPERATOR for CICS, and Energizer for CICS agents and facilities. From this view you can issue commands to control:

- the MainView extractor
- task kill agent
- the initialization and termination of BBI-SS PAS communication
- starting and stopping MainView AutoOPERATOR for CICS
- switching the dual CMRDETL data sets immediately
- forcing the writing of the VSAM CMRDETL buffer
- starting and stopping recording of T6E (CMRDETL) transaction history data to the VSAM file
- controlling the setting of the SMF return code from the MainView for CICS usage of the CICS XMNOUT global exit
- activating and deactivating session data collection
- program CPU timing data collection
- activating and deactivating temporary storage queue usage data collection
- activating and deactivating transient data queue usage data collection
- resetting CMRSOPT (monitoring options)
- resetting CMRPRBT (problem threshold table)
# Commands available with the CREGAGT view

This topic provides a table of commands available with the CREGAGT view.

## Table 35: CREGAGT commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOF</td>
<td>Terminate MainView AutoOPERATOR for CICS (same as the FST2 transaction’s SYSM OFF function).</td>
</tr>
<tr>
<td>AON</td>
<td>Start MainView AutoOPERATOR for CICS (same as the FST2 transaction’s SYSM ON function).</td>
</tr>
<tr>
<td>COF</td>
<td>Deactivate compression of the CICS SMF 110 performance records.</td>
</tr>
<tr>
<td>CON</td>
<td>Activate compression of the CICS SMF 110 performance records.</td>
</tr>
<tr>
<td>CSN</td>
<td>Deactivate session/connection activity recording.</td>
</tr>
<tr>
<td>CSY</td>
<td>Activate session/connection activity recording.</td>
</tr>
<tr>
<td>DOF</td>
<td>Stop recording of T6E (CMRDETL) transaction history data to the VSAM file. Cannot be issued from the FST2 transaction.</td>
</tr>
<tr>
<td>DON</td>
<td>Start recording of T6E (CMRDETL) transaction history data to the VSAM file. Cannot be issued from the FST2 transaction.</td>
</tr>
<tr>
<td>EOF</td>
<td>Set the XMNOUT return code to terminate SMF writing of the CICS exception records.</td>
</tr>
<tr>
<td>EON</td>
<td>Set the XMNOUT return code to initiate SMF writing of the CICS exception records.</td>
</tr>
<tr>
<td>FLU</td>
<td>Force (flush) the writing of the CMRDETL VSAM buffer.</td>
</tr>
<tr>
<td>INI</td>
<td>Initialize MainView for CICS agents. This command initializes the MainView for CICS agents by creating all resources, loading necessary programs, initiating communications with the BBI-SS PAS, and starting various functions, such as task kill, the extractor, and MainView AutoOPERATOR. The first time INI is used, a PLT-type initialization is performed. Subsequent uses restore the environment recorded by the TRM function unless the CICS was not connected to a BBI-SS PAS. In this case, the CMRSOFT settings for agent startup are used.</td>
</tr>
<tr>
<td>KOF</td>
<td>Terminate the task kill function.</td>
</tr>
<tr>
<td>KON</td>
<td>Start the task kill function.</td>
</tr>
<tr>
<td>KRL</td>
<td>Reload the task kill function or initiate it if it is not running.</td>
</tr>
<tr>
<td>LOF</td>
<td>Terminate the MainView for CICS long-running task BCRT (same as the FST2 transaction’s LOFF function). This command should not be used unless directed by BMC Software MainView for CICS Customer Support.</td>
</tr>
<tr>
<td>LON</td>
<td>Start the MainView for CICS long-running task BCRT (same as the FST2 transaction’s LON function). This command should not be used unless directed by BMC Software MainView for CICS Customer Support.</td>
</tr>
<tr>
<td>PGN</td>
<td>Stop the MainView for CICS Program CPU Timing facility.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>PGY</td>
<td>Start the MainView for CICS Program CPU Timing facility.</td>
</tr>
<tr>
<td>POF</td>
<td>Set the XMNOUT return code to terminate SMF writing of the CICS performance records.</td>
</tr>
<tr>
<td>PON</td>
<td>Set the XMNOUT return code to initiate SMF writing of the CICS performance records.</td>
</tr>
<tr>
<td>PRL</td>
<td>Reload the CMRPRBT, problem threshold table.</td>
</tr>
<tr>
<td>QOF</td>
<td>Shut down the MainView for CICS extractor (same as the FST2 transaction’s QOFF function).</td>
</tr>
<tr>
<td>QON</td>
<td>Start the MainView for CICS extractor (same as the FST2 transaction’s QON function).</td>
</tr>
<tr>
<td>RES</td>
<td>Restart MainView for CICS agents. This command is used after MainView for CICS maintenance has been applied to CICS regions only.</td>
</tr>
<tr>
<td>ROF</td>
<td>Set the XMNOUT return code to terminate SMF writing of the CICS resource records.</td>
</tr>
<tr>
<td>RON</td>
<td>Set the XMNOUT return code to initiate SMF writing of the CICS resource records.</td>
</tr>
<tr>
<td>SRL</td>
<td>Reload the CMRSOPT.</td>
</tr>
<tr>
<td>SWI</td>
<td>Switch the dual CMRDETL data sets immediately.</td>
</tr>
<tr>
<td>TDN</td>
<td>Deactivate transient data (TD) queue activity recording.</td>
</tr>
<tr>
<td>TDY</td>
<td>Activate transient data (TD) queue activity recording.</td>
</tr>
<tr>
<td>TRM</td>
<td>Terminate MainView for CICS agents. This command terminates the MainView for CICS agents, records the processing environment, and removes MainView for CICS from the CICS systems. The MainView for CICS extractor and task kill functions are terminated, as well as MainView AutoOPERATOR, and their status recorded in the BMCMVCTQ temporary storage queue. All MainView for CICS resources are discarded and released and communication with the BBI-SS PAS terminated.</td>
</tr>
<tr>
<td>TSN</td>
<td>Deactivate temporary storage (TS) queue activity recording.</td>
</tr>
<tr>
<td>TSY</td>
<td>Activate temporary storage (TS) queue activity recording.</td>
</tr>
</tbody>
</table>
Implementing OLTHCICS to control PLT processing

Some sites might require that the MainView CICS agents be initialized before allowing PLT processing to continue.

The MainView for CICS facilities on the CICS and the MainView AutoOPERATOR for CICS transient data exit will not start processing until communications have been established with the BBI-SS. When the MainView agent initialization is started from the PLT (see “Step 4: Modify the PLT” on page 114 in “Standard implementation procedures” on page 105), the BMC agent initialization occurs concurrently with PLT processing.

You can use an optional PLT program from BMC, OLTHCICS, to delay PLT processing until either communications are established and the BMC MainView agents have initialized, or a time limit has expired.

OLTHCICS waits to verify that the BMC MainView agents have established communications between the CICS region and the BBI-SS. After confirming the communications link, OLTHCICS returns control to the PLT, allowing CICS initialization to continue. By default, OLTHCICS will wait up to 60 seconds for the communications link to be established; you can change the default to any value from 10 to 150 seconds, as explained in “Implementing the OLTHCICS program” on page 272.

---

Note

BMC does not recommend using this procedure because it will delay initializing the CICS regions and prevent useful work from beginning until the BBI-SS is active and communications with the CICS region has begun.

The OLTHCICS program is provided only because some sites have requirements that either MainView for CICS or MainView AutoOPERATOR for CICS is active before allowing processing on the CICS region.

The failure to start the BBI-SS results in a delay of CICS startup of up to 150 seconds (depending on your settings) and each site should carefully consider the implications of delaying PLT processing before using this program.
Implementing the OLTHCICS program

You must use the DFHPLT macro to create a new entry in the PLT to implement the OLTHCICS program.

Insert the new entry after the entry for OLTFSET and before any entries for processing that should not start before the BMC agents are initialized.

Figure 29 on page 272 and Figure 30 on page 272 show two examples of the PLT macro placement, one using the standard DFHPLT macro and the other using the CMRPLT macro.

**Figure 29: Example 1: PLT macro placement with DFHPLT macro**

```
DFHPLT TYPE=INITIAL,SUFFIX=BB
DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
DFHPLT TYPE=ENTRY,PROGRAM=OLTFSET
DFHPLT TYPE=ENTRY,PROGRAM=OLTHCICS
DFHPLT TYPE=FINAL
END
```

**Figure 30: Example 2: PLT macro placement with CMRPLT macro**

```
DFHPLT TYPE=INITIAL,SUFFIX=BB
DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
CMRPLT MODE=AUTO
DFHPLT TYPE=ENTRY,PROGRAM=OLTHCICS
DFHPLT TYPE=FINAL
END
```

If your site is running with program autoinstall active (PGAIPGM=ACTIVE SIT), you do not have to create and install a program definition. Otherwise, you should add a CSD entry for the OLTHCICS program to your BMC installed group, using the following set of attributes:

```
DEFINE PROGRAM(OLTHCICS) GROUP(BMCMVOLT) EXECKEY(CICS)
  LANGUAGE(ASSEMBLER) RELOAD(NO) RESIDENT(NO)
  USAGE(NORMAL) USELPACOPY(NO) DATALOCATION(ANY)
  STATUS(ENABLED) CEDF(NO)
  DESCRIPTION(BMC CICS PLT DELAY PROGRAM)
```

You can control how long the OLTHCICS program waits for communications to be established (before returning control to CICS and allowing PLT processing to continue) with the INITPARM parameter in the SIT. By default, OLTHCICS will wait up to 60 seconds for the communications link to be established; you can change the default to any value from 10 to 150 seconds.
Figure 31 on page 273 shows an example of the INITPARM parameter specification that modifies the wait time to 30 seconds.

**Figure 31: Modify the wait time with the INITPARM parameter**

```
... INITPARM=(OLTHCICS='30') ...
```
Index

Symbols

! type identifier 161
(as session data identifier 165
) as connection data identifier 165
@RESP CREGION monitor 128

A

ACONCICS parameter 59–61, 69
activating SAPSTEC exit 185
Adabas user exit 250
adding CICS regions 213
adding code to SAPSTEC 184
adding graph data 215
affinity factors
  for AutoConnect 70
  examples 72
  how they work 70
ALTAID parameter 205
ALTCLAS parameter 205
ALTCONN parameter 205
ALTDEST parameter 205
ALTER parameter 205
ALTFILE parameter 205
ALTICES parameter 205
ALTPROG parameter 205
ALTTASK parameter 205
ALTTerm parameter 205
ALTTRAN parameter 205
ALTTSUT parameter 205
AOF command 268
AOFF command 263
AON command 263, 268
application trace facility
customizing 191, 199
default options 199
general options 196
logging options 199
trace directory 191
trace log data set 194
Applications Programming Interface (API) 161
archive JCL 84
ARCHIVE step, CMRDJCL 100
archiving
  CMRDETL data sets 96
  CMRUSR01 user exit 94
  MVS utility 101
  with CMRDETL data sets 96, 99
  with CMRUSR01 user exit 94
  with MVS utility 101
ARCHPOST step, CMRDJCL 100
ARCHPRE step, CMRDJCL 100
ARCVOL option 193
authorizing
  actions 205
  external security manager 203
  functions 205
  operator IDs 210
  with external security manage 203
auto maintenance
  starting with CREGAGT view 114
  BBSAMP members 57
  CSD definitions 49
  initiating 49
  messages 57
  overview 47
  resources 49
  using 47
AutoConnect facility
A

ACONCICS parameter 59–61, 69
affinity factors 70
automatic implementation 68
backward compatibility 66
BBI-SS PAS selection 65
initialization parameters 69
overview 65
automatic termination and initialization 231

B

BALPUTLV program 245
basic mapping service (BMS) monitoring 139
BBI-SS PAS
   AutoConnect affinity factors 70
   CICS communication link, automatic start up 114
   CICS communication link, stopping 232
directive to use specific 64
   manual control 231
   security 107
   user ID 107
BBIJNT00 member
   and BMCCIS00 66
   static JNT entries 61
   TYPE=CICS 62
BBIPROF library, history parameters 182
BBISSP00 member
   ACONCICS parameter 59–61, 69
   DYNTGT parameter 63
BBK1CCRE program 245
BBK1CRTR program 106, 109, 245
BBK1RE3x program 245
BBK1RE5x program 245
BBK1TB1x program 245
BBK1TB2x program 245
BBK1TB3x program 245
BBK1TB4x program 245
BBK1TB5x program 245
BBK1TB8x program 245
BBK1TBAx program 245
BBK1TBEx program 245
BBK1TBFx program 245
BBK1TE2x program 245
BBK1TE3Ax program 245
BBK1TE4x program 245
BBK1TE5x program 245
BBK1TE6x program 245
BBK1TE9x program 245
BBK1TEAx program 245
BBK1TEDx program 245
BBK1TEEx program 245
BBK1TEFx program 245
BBMLIB library 57, 151
BBMTXP00 member 63
BBPARM library
   BBMTXP00 member 63
BBSAMP library
   auto maintenance members 57
   JXT001 member 192
   list of members 249
   SMF 110 record member 120
BBSECURE data set 203
BCRT transaction
   and BBK1CRTR 106, 109
description 243
   starting 263
   stopping 263
   ZAP for changing ID 244
BDR#DEF member 182, 250
BMCE transaction
   how started 243
   reconnecting to CICS 114
   started by EXCI 49
   ZAP for changing ID 244
BMCMVCTQ temporary storage queue 225
BMS parameter 139
BROADCAST application 185, 219

C

C++ code 176
CA-ACF2 support 203
CA-IDEAL monitoring 136
CA-TOP SECRET support 203
CAODTAB macro
description 187
CAOTTAB macro 185
CASID parameter 63
CBQCMD program 50, 245
CCICSCMD resource class 107
CEMT INQUIRE ENQMODEL 106, 109
CEMT INQUIRE TSMODEL 106
CEMT monitoring statistics 146
CFLOW view 23
CHCONN views 167
CHISTDIO view 165, 167, 169, 170
CHISTDJ view 169
CHISTDJ1 - 5 view 169
CHISTDK view 170
CHISTDK1 - 5 view 170
CHISTDN view 165
CHISTDNx views 165
CHSESS view 165
CHTDQ view 169
CHTDQD view 169
CHTSQ view 170
CHTSQD view 170
CICS
  releases supported 17
  adding a region 213
  release-dependent programs 245
  security 107
  SMF 110 record 118
  startup JCL 112
  statistical records 107
  surrogate user checking 107
  SYSIN IRCSTRT parameter 117
  upgrading a region 214
CICS data extractor 77
CICS document template 140
CICS independent IMFEXEC commands 219
CICS PLTPI processing 160
CICS Transaction Gateway
  CLASSPATH parameter 176
  CTG environment modifications 176
  MainView (for CICS) modifications 172
  overview 171
  starting and resetting monitoring 180, 239
CICS virtual storage 26
CICSCLI parameter 177
CICSRACF member 249
CJVMPL view 205
CLASSPATH parameter, CTG 176
cloned AORs 65
CMDSEC parameter 107
CMR$2SAS member 121
CMR$DATA member 120, 250
CMR$DB2 member 121, 250
CMR$DMCT member 250
CMR$DSAS member 120, 250
CMR9RLSE program 245
CMRADAX member 250
CMRADBX member 250
CMRAPPL variable 83, 88
CMRARCH1 program 100
CMRARCH2 program 100
CMRASMM member
description 249
use with CMRPRBRT 151
use with CMRSECU 204
use with CMRSoPT 145
CMRBARC member 250
CMRBEX00 member 195
CMRBEXnn member 173, 180, 239
CMRBRLOD member 250
CMRCBxxN member 250
CMRCClUEX program 259
CMRCHRT member 250
CMRClJCL member 250
CMRCMPXx program 245
CMRCOBCN member 250
CMRCOBCP member 250
CMRCOBJn members 250
CMRCONV member 250
CMRCOPY member 250
CMRCSDAL member 57, 250
CMRCSDAEL member 57, 250
CMRCSDLA member 57, 112, 250
CMRCSDLB member 57, 112, 250
CMRCSDLM member 57, 112, 250
CMRCClUEX user exit
  adding user data 256
  CMRCClUEX (for COBOL) 259
  coding conventions 257
  database calls 255
  description 255
  examples of usage 258
  program calls 256
  restrictions 258
  sample user exit 250
CMRDATA member 118
CMRDB2 member 118
CMRDDTL member
  create history data sets 89
description 250
CMRDDETl data sets
  activating 128
data from CMRCClUEX 255, 256
displaying messages 95
dual data sets 95
expanded record format 41
implementing 122
migration considerations 41
purging records 96, 99
size 90
switching data sets 94
<table>
<thead>
<tr>
<th>CMRDETL data sets, archiving</th>
<th>101</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCB characteristics</td>
<td>101</td>
</tr>
<tr>
<td>overview</td>
<td>96</td>
</tr>
<tr>
<td>process</td>
<td>96, 99</td>
</tr>
<tr>
<td>CMRDETL definitions</td>
<td>77</td>
</tr>
<tr>
<td>CMRDETL parameter</td>
<td>128</td>
</tr>
<tr>
<td>CMRDETL view</td>
<td>80</td>
</tr>
<tr>
<td>CMRDETLN member</td>
<td>250</td>
</tr>
<tr>
<td>CMRDFSUM member</td>
<td>250</td>
</tr>
<tr>
<td>CMRDJCL member</td>
<td></td>
</tr>
<tr>
<td>and CMRPURG</td>
<td>99</td>
</tr>
<tr>
<td>ARCHIVE step</td>
<td>100</td>
</tr>
<tr>
<td>ARCHPOST step</td>
<td>100</td>
</tr>
<tr>
<td>ARCHPRE step</td>
<td>100</td>
</tr>
<tr>
<td>description</td>
<td>250</td>
</tr>
<tr>
<td>CMRDLDF member</td>
<td>250</td>
</tr>
<tr>
<td>CMRDParm member</td>
<td>99</td>
</tr>
<tr>
<td>CMRDTL00 member</td>
<td>80, 94</td>
</tr>
<tr>
<td>CMRFCMDx program</td>
<td>245</td>
</tr>
<tr>
<td>CMRINT2 program</td>
<td></td>
</tr>
<tr>
<td>CMRJOBN variable</td>
<td>83, 88</td>
</tr>
<tr>
<td>CMRJRNs enqueue</td>
<td>106, 109, 227</td>
</tr>
<tr>
<td>CMRKLCPx program</td>
<td>245</td>
</tr>
<tr>
<td>CMRL member</td>
<td>250</td>
</tr>
<tr>
<td>CMRLDLT batch utility</td>
<td>101</td>
</tr>
<tr>
<td>CMRLDLT member</td>
<td>250</td>
</tr>
<tr>
<td>CMRMVJCL member</td>
<td>250</td>
</tr>
<tr>
<td>CMRN2MCT member</td>
<td>119, 250</td>
</tr>
<tr>
<td>CMRN2SAS member</td>
<td>119, 250</td>
</tr>
<tr>
<td>CMRNDATA member</td>
<td>119, 250</td>
</tr>
<tr>
<td>CMRNDDB2 member</td>
<td>119, 250</td>
</tr>
<tr>
<td>CMRNDMCT member</td>
<td>119, 250</td>
</tr>
<tr>
<td>CMRNDsAS member</td>
<td>119, 250</td>
</tr>
<tr>
<td>CMRPAS variable</td>
<td>88</td>
</tr>
<tr>
<td>CMRPAS member</td>
<td>250</td>
</tr>
<tr>
<td>CMRPLT macro</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>249</td>
</tr>
<tr>
<td>using</td>
<td>115</td>
</tr>
<tr>
<td>CMRPLTN macro</td>
<td>117</td>
</tr>
<tr>
<td>CMRPRBt macro</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>151, 250</td>
</tr>
<tr>
<td>modifying</td>
<td>151</td>
</tr>
<tr>
<td>resetting</td>
<td>155</td>
</tr>
<tr>
<td>CMRPRBt view</td>
<td>216</td>
</tr>
<tr>
<td>CMRPRLn members</td>
<td>250</td>
</tr>
<tr>
<td>CMRPURG program</td>
<td></td>
</tr>
<tr>
<td>archiving dual CMRDETL data sets</td>
<td>99</td>
</tr>
<tr>
<td>description</td>
<td>96</td>
</tr>
<tr>
<td>CMRPURGE member</td>
<td>96, 250</td>
</tr>
<tr>
<td>CMRPX members</td>
<td>250</td>
</tr>
<tr>
<td>CMRQiT6E member</td>
<td>250</td>
</tr>
<tr>
<td>CMRRAP member</td>
<td>250</td>
</tr>
<tr>
<td>CMRRAPM macro</td>
<td>250</td>
</tr>
<tr>
<td>CMRREL variable</td>
<td>88</td>
</tr>
<tr>
<td>CMRRPRO member</td>
<td>250</td>
</tr>
<tr>
<td>CMRROLLx program</td>
<td>245</td>
</tr>
<tr>
<td>CMRS2AX member</td>
<td>250</td>
</tr>
<tr>
<td>CMRS2BX member</td>
<td>250</td>
</tr>
<tr>
<td>CMRSASxx members</td>
<td>250</td>
</tr>
<tr>
<td>CMRSAxxN members</td>
<td>250</td>
</tr>
<tr>
<td>CMRSECu macro</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>204, 249</td>
</tr>
<tr>
<td>modifying</td>
<td>204</td>
</tr>
<tr>
<td>parameters</td>
<td>205</td>
</tr>
<tr>
<td>resetting</td>
<td>211</td>
</tr>
<tr>
<td>TYPE=ENTRY statement</td>
<td>210</td>
</tr>
<tr>
<td>TYPE=FINAL statement</td>
<td>211</td>
</tr>
<tr>
<td>TYPE=INITIAL statement</td>
<td>205</td>
</tr>
<tr>
<td>CMRSITR member</td>
<td>250</td>
</tr>
<tr>
<td>CMRSJCL member</td>
<td>250</td>
</tr>
<tr>
<td>CMRSLOGx program</td>
<td>245</td>
</tr>
<tr>
<td>CMRSM variable</td>
<td>88</td>
</tr>
<tr>
<td>CMRSMLXT member</td>
<td>250</td>
</tr>
<tr>
<td>CMRsopt macro</td>
<td></td>
</tr>
<tr>
<td>and product initialization</td>
<td>263</td>
</tr>
<tr>
<td>modifying</td>
<td>145, 184</td>
</tr>
<tr>
<td>non-reentrant</td>
<td>145</td>
</tr>
<tr>
<td>parameters</td>
<td></td>
</tr>
<tr>
<td>4GL</td>
<td>136</td>
</tr>
<tr>
<td>additional data types</td>
<td>139</td>
</tr>
<tr>
<td>database</td>
<td>136</td>
</tr>
<tr>
<td>default monitoring conditions</td>
<td>128</td>
</tr>
<tr>
<td>SMF data</td>
<td>140</td>
</tr>
<tr>
<td>resetting</td>
<td>149</td>
</tr>
<tr>
<td>sample member</td>
<td>250</td>
</tr>
<tr>
<td>SRL command</td>
<td>149</td>
</tr>
<tr>
<td>START parameter</td>
<td>229</td>
</tr>
<tr>
<td>type statements</td>
<td>145</td>
</tr>
<tr>
<td>TYPE=FINAL statement</td>
<td>149</td>
</tr>
<tr>
<td>TYPE=INITIAL statement</td>
<td>140, 146</td>
</tr>
<tr>
<td>TYPE=resource statement</td>
<td>146</td>
</tr>
<tr>
<td>CMRsopt view</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>125</td>
</tr>
<tr>
<td>TARGET parameter</td>
<td>127</td>
</tr>
<tr>
<td>CMRSPARM member</td>
<td>250</td>
</tr>
<tr>
<td>CMRSTATS record types</td>
<td>217</td>
</tr>
<tr>
<td>CMRSTATx member</td>
<td>107, 250</td>
</tr>
<tr>
<td>CMRSTJCL member</td>
<td>250</td>
</tr>
<tr>
<td>CMRSUMD member</td>
<td>250</td>
</tr>
<tr>
<td>CMRSUMDx members</td>
<td>250</td>
</tr>
</tbody>
</table>

MainView for CICS Customization Guide
Index

CMRSUP0x members 250
CMRTARG variable 83, 88
CMRTBTRA member 250
CMRTDPX program 245
CMRTMAP program 245
CMRTMON program 245
CMRTRUEX program 245
CMRTTHnn table
  migration consideration 27
  overview 155, 156
reloading 161
CMRTTHR macro 250
CMRTTHR view 155, 156
CMRUACMP member 250
CMRUAPLT member 250
CMRURAPM member 250
CMRUSECU member
description 204, 249
CMRUSOPT macro 145, 250
CMRUSR01 user exit 93, 94, 250
CMRUSRSD member 250
CMRVCMD program
  CSD definition 49
description 245
CMRWKRK1 data set 96
CMRXEIOx program 245
CMRXEIST program 245
CMRXTFG program 245
COBOL report programs 41
COF command 268
COLD start 117, 231
common communication request queue 68
compatibility
  SMF 110 records 107
temporary storage queues 106
CON command 268
CONTINUE parameter 84, 91
conventions, documentation 14
creating a target definition 63
creating BBIJNT00 elements 61
CREGAGT view
commands 268
nenqueue CMRJRNL 227
EXCI function 114
line commands 162
program timing 237
security 107
session monitoring 168, 238
TD queue monitoring 169
temporary storage monitoring 171
CREGION view 151
CREGSYS view 205
CRGRESLOG parameter 128
Cross System Product (CSP) monitoring 136
CSA parameter 140
CSD definitions 49
CSDCMRDF member 57, 250
CSDOLTDF member 57, 244, 250
CSLOAD program 245
CSLOADC program 245
CSLOADDC program 245
CSP parameter 136
CSSY task 77
CSNE task 77
CSOFF command 168, 238, 263
CSN command 168, 238, 263
CSNC task 77
CTASKDJ view 169
CTASKDK view 170
CTCPSRV view 205
CTG environment file 176
tg.ini file 177
CTGHOST option 174
CTGMON parameter 173
CTGNNAME parameter 174
CTRANX views 205
CTRNCLA view 205
customization
  application trace facility 191, 199
  checklists 213, 218
  CMRDETL data sets 77, 101
  monitoring conditions 125, 149
  of product libraries 45
  optional procedures 125
  product libraries 45
  sample tasks 213, 218
  security 107, 203, 211
  special considerations 106, 107
  standard procedures 47, 105
D

data collection
  automatic start up 114
  controlled by QON and QOF 77
  manual control 232
data extractor
  exclusions 77
  program timing function 161
Data Set Create Option 89
Databases
  capturing statistics 255
  monitoring 136
DATACOM user exit 250
DB2
  monitoring 136
  plan activity 128
  securing plan display 205
  securing system parameters 205
DBCTL monitoring 118, 120
DCCTXPR member 250
DCT parameter 140
default monitoring conditions 128
DEEXEC parameter 182
DEFREC parameter 182
deleting graph data 215
detail trace options 198
Development Management System (DMS)
  monitoring 136
DFHSEXC1 group 114
DFHMCT user clocks 139
DFHPG message 57
DFHRPL library 43, 112
DFHSTUP program 107
DFHXCOPT macro 122
DFHXM message 57
directive to use specific BBI-SS PAS 66
distributed libraries 45
distributed program link (DPL) 166
distributed transaction processing (DTP) 165
DL/I monitoring 118
DLCCLK1 parameter 139
DLCCLKN parameter 139
DLCNT1 parameter 139
DLCNTN parameter 139
DLOGPROC parameter 128
DLZ parameter 140
DMP parameter 140
DMS parameter 136
DMSOFF parameter 136
DOCTEMPLATE, CICS 140
documentation information 13
DOF command 268
DON command 268
DSPLIM option 173
DSPSIZE parameter 174
DTRACSIZ parameter 128, 196, 198
dual CMRDETL data sets 90
DUMP view 26
dynamic JNT 66
DYNAMIC parameter 128
Dynamic Target Definition (TGTDEF) elements 59
dynamic targets
  definitions 64
  enabling 63
  implementing 62
Dynamic Targets facility 65
dynamically defined targets 63
DYNTGT parameter 63

E
electronic documentation 13
emergency restart 117
ENQ model 106, 109
ENQ parameter 140
enqueues
  BMCFST2Q 226
  CMRJRNL 227
  name conflicts 106, 109
EOF command 263, 268
EOFF command 263
EON command 263, 268
ESM. See external security manager 107
EXCI facility
  and auto maintenance 49
  connection and session definitions 49, 114
  redirecting requests 114
EXCLUDE parameter 128
excluding messages from PROBLEM service 151
EXCMSP parameter 182
external security manager (ESM)
  CICS considerations 107
  overview 203
extractor agent shutdown 263
Extractor Status
  turning off 234
  turning on 232
EZBNMIFR program 172

F

FCD2 transaction
  description 243
  ZAP for changing ID 244
FCM1 transaction
  description 243
  ZAP for changing ID 244
FCT parameter 140
FEPI parameter 140
FIC2 parameter 128
FIC2 transaction
  description 243
  SMF data recording 140
  ZAP for changing ID 244
file requests function shipping 166
FILEXPND parameter 128, 164
FLU command 263, 268
fourth-generation languages (4GL)
  description 136
  monitoring 136
  passing program names through CMRCUEX 256
FROM-DATE statement 102
FST2 transaction
KON command 235
  starting the BBI-SS PAS 114
  and auto maintenance 49
  and enqueue name conflicts 106, 109
  and program timing 162
  and session monitoring 238
BBKICRTR enqueue 226
BMCFST2Q enqueue 226
  commands 243
INIT parameter 229
KOF command 236
KRLD command 161, 236
PGON command 237
QOFF command 100, 234
QON command 100, 232
  session monitoring 168
  SWITCH command 99
temporary storage monitoring 171
TERM command 106, 225
transient data queue monitoring 169
ZAP for changing ID 244
FTnnnn message statements
  editing 151
  format 151

G

GENEROL user exit 250
global statistical records 217
Global User Exits (GLUE) 77
GNI parameter 140
GPERIOD parameter 128
graph thresholds 217

H

Help
  online 13
history file allocation 84
History File Specification dialog
  fields 84
history parameters, BBIPROF 182
HISTORY service, securing displays 205
home dataspace 173, 180

I

IDCAMS command 128
IDCAMS REPRO utility 101, 104
IDEAL parameter 136
IMFEXEC CICS command 205, 219
implementation
  application trace facility 191, 199
  checklists 213, 218
  CMRDETL data sets 77, 101
  general considerations 106
  monitoring conditions 125, 149
  of product libraries 45
  optional procedures 125
  product libraries 45
  sample tasks 213, 218
  security 107, 203, 211
  special considerations 107
  standard procedures 47, 105
IMS/DBCTL request function shipping 166
IN-TARGET statement 102
INCLUDE parameter 128
INI command 268
INI line command 114, 229
INIT command 263
INIT function
  and task kill 157
  how it works 47
initializing MainView (for CICS)
  automatically 231
  from terminal session in windows mode 229
  manually 229
initializing MainView agents 263
INITPARM parameter 64
interval control requests function shipping 166
interval control services 225
interval recorder collection 151
IPSTACK parameter 174
IRC
  access method 114
  function 117
  parameter 140

J

JCL members

CMRASM 145, 151, 204, 249
CMRBARC 250
CMRBRLOD 250
CMRCHRT 250
CMRCJCL 250
CMROBJn 250
CMRCNV 250
CMRCOPY 250
CMRCSDEL 57, 250
CMRCSDAL 57, 250
CMRCSDDL 57, 250
CMRCSDLM 57, 250
CMRDDTL 250
CMRDJSUM 250
CMRJCL 99, 250
CMRJCL 250
CMRL 250
CMRLDLF 250
CMRMVJCL 250
CMRPURGE 96, 250
CMRRAP 250
CMRREPRO 250
CMRSJCL 250
CMRSTATX 250
CMRSJCL 250
CMRTBTRA 250
JXT0001 192
JXT003 193
JXT011 194
list of 249
TTABJCL 185
TTHRJCL 250
JCT parameter 140
JNL2 transaction
  description 243
  enqueue definition conflicts 227
  enqueue on CMRJRNL 106, 109
  ZAP for changing ID 244
Jobname Node Table (JNT)
  description 59
  static entries 61
JXT001 JCL member 192
JXT003 JCL member 193
JXT011 JCL member 194

K

kill function. See task kill function 155, 156
KOF command 268
KOF line command 236
KOFF command 263
KOFF line command 236
KON command 263, 268
KON line command 235
KRL command 263, 268
KRL line command 236
KRLD command 263
KRLD line command 236

L
left parenthesis as session data identifier 165
LOF command 268
LOFF command 263
LON command 263, 268
LPCZZMION member 250
LTX parameter 140

M
macros
  CAODTAB 187
  CAOTTAB 185
  CMRPRBT 151
  CMRSECU 204
  list of 249
MainView agents 263
MainView AutoOPERATOR (for CICS)
  BROADCAST application 185
  CAODTAB macro 187
  customizing 185
  functions provided by customization 219
  optional customization 185
  Rule Processor 187
  starting 263
  support for 18
  TDQs eligible for rule processing 187
  terminal types eligible for broadcast 185
  terminating 263
MANPGM parameter 136
MANTIS monitoring 136
MANTIS parameter 136
manual control
  BBI-SS PAS communication 231
  monitoring 232
manually creating history data sets 89
manually starting MainView (for CICS) 229
manually terminating MainView (for CICS) 230
MAXEXC parameter 182
MAXFILE parameter 165, 169, 170, 255
MAXGATE option 173
maximum size parameter 79
MAXLOST parameter 128
MAXLOSTT parameter 128
MAXMSG parameter 182
MAXREC parameter 182
MAXRESP parameter 128, 146, 217
messages
  data set switching 95
  library 151
  modifying with CMRPRBT 150, 151
  session monitoring 238
  setting thresholds 150, 151
  TD queue monitoring 240, 241
MESSAGES General Services option 57
MESSAGES service 95
migration considerations
  any release 41
  audit data for parameter updates 37
  batch report programs 41
  BBI-SS PAS release 43
  CICS support 17
  CMRCUEX 41
  CMRDETL records 41
  CMRSTATS records 38
  CTG support 41
  MainView AutoOPERATOR support 18
  monitor messages 39
  new views 33, 42
  release-specific 17, 22, 32, 37
  task kill exit threshold table 27
  updated views 42
MILLEN parameter 136
Millennium monitoring 136
MNCONV parameter 77, 117
MNEXC parameter 77, 117
MNFREQ parameter 77
MNPER parameter 77, 117
MNSYNC parameter 77
MODIFY command to invoke FST2 transaction 232
MON parameter 140
MONITOR parameter 146
monitoring
4GLs 136
automatic start up 114
BMS 139
CA-IDEAL 136
CSP 136
customization 125, 149
databases 136
DB2 136
default conditions 128
DFHMCT user clocks 139
DMS 136
implementation 125, 149
MANTIS 136
manual control 232
Millennium 136
overview 125
response time service levels 139
SAP 184
SMF data 140
specifying resources 146
stopping 234
monitoring conditions, setting 125
Monitoring Control Table (MCT)
DB2 data 121
description 118
modifying 118
MQFOFF parameter 128
MSG command 57

N
NAME parameter 146
Natural user exit 250
NCIRDCX1 member 250
netname, changing 257
NEWSIT parameter 117
NTGRAPH parameter 128

O
OLTCCRE program 245
OLTCNTL program 245
OLTFSET program 49, 114, 245
OLTFSETx program 245
OLTJRN XLx program 245
OLTSCAO program 245
OLTSCMRx program 245
OLTUMOD0 member 250
OLTVCAO program 245
OLTVER program 245
online Help 13
OTHR parameter 140
OUT-TARGET statement 102

P
PAM parameter 140
PARM statement and trace 193
PCS parameter 136
PCSOFF parameter 136
PCT parameter 140
percent and time thresholds 151
Performance Reporting Language (PRL) 95
PGN command 162, 263, 268
PGOF command 238, 263
PGON command 237, 263
PGY command 162, 263, 268
Plex Manager
security views 203
TGTDEF view 63
PLT. See Program List Table (PLT) 116
PLTPI parameter 116
POF command 263, 268
POFF command 263
POL parameter 140
PON command 263, 268
PORT parameter 174
PPT parameter 140
preparing dual CMRDETL recording 216
private data space 79
PRLcommand 268
problem message threshold table 150
PROBLEM service 151
problem thresholds, setting 150, 151
product libraries 45
Program Auto-Install Facility 161
Program CPU Timing facility, starting 237
Program List Table (PLT)
creating 116
description 114
modifying existing 115
program timing 161
publications, related 13
purge date 96

Q
QOF command 268
QOF line command 234
QOFF command 263
QON command
static and dynamic targets 232
CREGAGT view 268
FST2 transaction 263
START parameter 128

R

RACF command 107
RCT parameter 140
RECOVERY attribute 106
related publications 13
reloading task kill function 236, 263
REMOTE attribute 106
removing static targets 62
REQID parameter 106
RES command 49, 268
RESET command
for CMRPRBT table 155
for CMRSECU table 211
for CMRSOPT 149
for CTG 180, 239
response time, monitoring 139
right parenthesis as session data identifier 165
ROF command 263, 268
ROFF command 263
RON command 263, 268
Rule Processor Detail Control
field summary 155

S

S106 abends 185
SAF security 203
sample members, using
CMRUPRBT 151
CMRUSECU 204
CMRSOPT 145
samples library
list of members 249
SAP monitoring
implementing 184
sample SAPSTEC members 250
SAPCMROT member 184
SAPCMRTW member 184
SAPCMRWRT member 184
SAPSTEC exit 184
SAPSTEC members 250
SASi6ExN members 250
SASIN6Ex members 250
SASINxx members 250
SASINXxx members 250
SDFHEXCI data set 122
SDFHSAMP library 120
SECPRFX parameter 107
security
authorization 107
by operator I 210
CICS considerations 107
CMRSECU macro 204
external 203
for functions and action 205
functions and actions 205
operator ID 210
Plex Manager views 203
SAF 203
trace log data set requirements 191
SERVICE parameter 139
session definitions 114
session monitoring
messages 238
starting and stopping 168, 238
SETDMPD parameter 205
SETJVMP parameter 205
SETSYS parameter 205
SETTCPS parameter 205
SIT. See System Initialization Table 117
SMF 110 records
compatibility 107
data recording 140
format of user data 118
subtype x\xd5 0002\xd5 107
subtypes 140
SMN2 transaction
description 243
task monitoring 214
ZAP for changing ID 244
SRL command 149, 268
SSID parameter 174
START parameter
BBI-SS PAS 43
monitoring 128
starting data collection for other products 215
starting MainView (for CICS) manually 229
starting MainView (for CICS) monitoring
from BBI-SS terminal session in windows mode 232
from CICS terminal 232
from MVS terminal 232
starting task kill function 235
startup JCL, CICS 112
static BBIJNT00 member 66
static JNT entries 61
static targets 62
statistical records, CICS 107
STATIC macro 250
STDENV statement 176
STI parameter 140
STOP parameter 91
stopping full data monitoring 214
stopping MainView (for CICS) monitoring
from MVS console 234
from terminal session in windows mode 234
stopping task kill function 236
STRACSIZ parameter 128, 196
SUBSYS parameter 64
supported CICS releases 17
SUPRA user exits 250
surrogate user 107
SWI command 263, 268
switching CMRDETL data sets 94
syntax statement conventions 14
SYS2K user exits 250
SYSM OFF command 263
SYSM ON command 263
SYSDUMP statement 122
System Initialization Table (SIT)
modifying 117
sample report 250
SECPRFX parameter 107
security 107
system variables for history data sets 88

target definition 213
TARGET parameter 127
TARGET statement 62, 64
task kill function
  global exit points XEIIN and XRMIOU 157
  limitations 159
  overview 155, 156
  reloading 161, 236, 263
  starting 160, 235
  stopping 160, 236
task performance detail history data sets 89
Task Performance Detail History File Definitions 77
Task Related User Exit (TRUE) 77
TASKCON views 167
TASKKILL parameter 128, 160, 235
TASKPGM view 162
TASKSES view 165
TASKTDQ view 169
TASKTDQD view 169
TASKTSQ view 170
TASKTSQD view 170
TCT parameter 140
TD queue monitoring
  starting and stopping 240, 241
TDN command 240, 241, 263, 268
TDOF command 240, 241, 263
TDON command 240, 241, 263
TDY command 240, 241, 263, 268
temporary storage (TS) queues 170
temporary storage function shipping 166
temporary storage queues
  BMCMVCTQ 106, 225
  CMRI prefix 106
Temporary Storage Table (TST)
  compatibility 106
  name conflicts 225
  parameter 140
TERM command 230, 263, 268
TERM function
  and task kill 157
  how it works 47
  initiation 49
TERM request, BBI-SS PAS-to-CICS communication 232
terminated AutoConnect targets 69
terminating MainView (for CICS) manually 230, 268
terminating MainView agents 263
terminating task kill agent 263
TGTACT view 63
TGTDEF INSTALL command 63
TGTDEF view 63
threshold values for count messages 151
thresholds, setting problem 150, 151
time limit expires 80
TIMESTAMP parameter 182
TLDASGEstatement 192
TO-DATE statement 102
trace
  buffer pools 198
  log 191
trace directory, preparing 191
trace log data set
  attributes 199
  creating 194
TRACESZE parameter 122
tracing. See application trace facility 191
transaction class DFHTCL00 205
transaction conflict Zap 244
transient data (TD) function shipping 166
transient data (TD) queues 169
transient data traffic 187
TRCYL parameter 199
TRDIR statement 192
TRM command 268
TRM line command 230
TRPREFIX parameter 199
TRREUSE parameter 199
TRSIZE parameter 199
TRSMSDCL parameter 199
TRSMSPMCL parameter 199
TRSMSSCL parameter 199
TRSUFX parameter 199
TRVOLS parameter 199
TS queue monitoring
  messages 240, 241
TSN command 263
TSOF command 263
TSQUNEx views 205
TST. See Temporary Storage Table 225
TSY command 263, 268
TTABJCL JCL member 185
TTHRJCL member 250
adding a target statement 64
BBJNT00 member 59, 61
data set 213
UBBSAMP
  and TTABJCL 185
  data set 145
UFO parameter 136
UFINIT parameter 136
UFOPGM parameter 136
URMPO parameter 114
USERNM parameter 128
user clocks, DFHMCT 139
user exits
Adabas 250
CMRUACMP 250
DATACOM 250
GENEROL 250
Natural 250
SUPRA 250
user-defined EXCI connection 49

V
virtual storage display 26
VSAM data sets
  allocating for trace 194
  securing shared resource display 205
VSAM performance 90
VSAM REUSE parameter 98

W
WAITINT parameter 128
WARM start 231
WEB parameter 140

X
X\xd5 0B02\xd5 record subtype 107
XCMD parameter 107
XDLIPOST global exit 166
XDLIPRE global exit 166
XEIIIN global exit 157, 166
XEICOUT global exit 166
XFILE parameter 151
XMNOU global exit 263
XNCIRDC1 user exit 250
XRMIOUT global exit 157
XTDEREQC global exit 166
XTRAN parameter 107, 151
XUSER parameter 107

Z

ZAPs, transaction ID 244