MAINVIEW® for DBCTL
Customization Guide

Version 3.3.30

November 2003

Contacting BMC Software

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

United States and Canada

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

Telephone 713 918 8800 or
800 841 2031

Fax 713 918 8000

Outside United States and Canada

Telephone (01) 713 918 8800

Fax (01) 713 918 8000

Customer Support

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

Support Web Site

You can obtain technical support from BMC Software 24 hours a day, 7 days a week at **http://www.bmc.com/support.html**. From this Web site, you can

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

Support 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 800 537 1813. Outside the United States and Canada, please contact your local support center for assistance. To find telephone and e-mail contact information for the BMC Software support center that services your location, refer to the Contact Customer Support section of the Support page on the BMC Software Web site at **http://www.bmc.com/support.html**.

Before Contacting BMC Software

Before you contact BMC Software, have the following information available so that Customer Support can begin working on your problem immediately:

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

Contents

Tables

How to Use This Book

This book contains procedures for customizing MAINVIEW® for DBCTL (MVDBC). MAINVIEW for DBCTL is based on the MAINVIEW environment, which allows a single terminal session (TS) to monitor and manage multiple local or remote targets, whether OS/390 itself (sysplex and nonsysplex) or subsystems like CICS, IMS, and DB2.

For information about what's new in the current release of MAINVIEW for DBCTL, see the product Release Notes, which are available on the BMC Software Support Web pages.

You can view this book online with Adobe Acrobat Reader. Contact your system administrator if you need assistance.

To install and customize MAINVIEW for DBCTL, follow the instructions in the following books.

- *MAINVIEW Installation Requirements Guide* – to determine software, storage, and system requirements
- *OS/390 and z/OS Installer Guide* – to load the product libraries
- *MAINVIEW Common Customization Guide* – to set up the operational environment of all MAINVIEW products at your site
- *Implementing Security for MAINVIEW Products* – to secure MAINVIEW product resources with your external security manager (ESM), such as CA-ACF2, CA-TOP SECRET, or RACF
- *MAINVIEW for DBCTL Customization Guide* (this book) – to tailor MAINVIEW for DBCTL to your site's requirements
- *MAINVIEW Administration Guide* – to maintain the MAINVIEW environment

The *MAINVIEW Common Customization Guide* and the *MAINVIEW Administration Guide* consolidate the instructions for implementing and administering the environment for all MAINVIEW products.

You must customize the MAINVIEW operational environment as described in the *MAINVIEW Common Customization Guide* before customizing MAINVIEW for DBCTL.

If you used AutoCustomization as described in the *OS/390 and z/OS Installer Guide*, you do not need to customize the MAINVIEW environment and MAINVIEW for DBCTL; that process is already done. You can use the *MAINVIEW Common Customization Guide*, the *MAINVIEW Administration Guide*, and this *MAINVIEW for DBCTL Customization Guide* as references during AutoCustomization or manual customization.

If you are installing MAINVIEW for DBCTL or MAINVIEW AutoOPERATOR for IMS, please note that there are some common steps between this guide and the *MAINVIEW AutoOPERATOR Customization Guide* (see Chapter 6, "Implementing MAINVIEW Products in IMS" on page 21).

MAINVIEW for DBCTL Product Library

The MAINVIEW for DBCTL product library includes the following documents:

MAINVIEW for DBCTL Customization Guide

MAINVIEW for DBCTL Analyzers, Monitors, and Traces Reference Manual

MAINVIEW for DBCTL IPSM Reference Manual

MAINVIEW for DBCTL Release Notes

How This Book Is Organized

This book is intended for the system programmer who needs to know how to modify the basic MAINVIEW product installation to include MAINVIEW for DBCTL functions.

This book is organized into the following parts:

- Part 1, "Migration Considerations" on page 1, contains release updates that must be considered if you have a previous release of MAINVIEW for DBCTL online products.
- Part 2, "Customizing MAINVIEW for DBCTL Functions" on page 15, describes how to tailor the MAINVIEW for DBCTL product components.
- Part 3, "Implementing Product Security" on page 69, contains instructions for allowing user access to MAINVIEW for DBCTL services.
- Part 4, "Appendixes" on page 73, provides supplemental information for MAINVIEW for DBCTL usage.

Recommended Reading

Before using this book to customize MAINVIEW for DBCTL, you must

1. Install the product as described in the *OS/390 and z/OS Installer Guide*
2. Tailor the MAINVIEW environment for the MAINVIEW for DBCTL functions as described in the *MAINVIEW Common Customization Guide*

Related MAINVIEW Product Libraries

Other MAINVIEW products that work with MAINVIEW for DBCTL include

MAINVIEW® AutoOPERATOR™
MAINVIEW® for CICS
MAINVIEW® for DB2®
MAINVIEW® FOCAL POINT
MAINVIEW® for IMS
MAINVIEW® for MQSeries
MAINVIEW® for OS/390
MAINVIEW® VistaPoint™

The following manuals document product-specific customization instructions:

MAINVIEW AutoOPERATOR Customization Guide
MAINVIEW for CICS Customization Guide
MAINVIEW for DB2 Customization Guide
MAINVIEW for IMS Online – Customization Guide
MAINVIEW for IMS Offline – Customization and Utilities Guide
MAINVIEW for OS/390 Customization Guide

The following books document the use of general services common to MAINVIEW for DBCTL and to some of the products mentioned above:

MAINVIEW AutoOPERATOR Basic Automation Guide
MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECS
MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECS
MAINVIEW for CICS PERFORMANCE REPORTER User Guide
MAINVIEW for DB2 User Guide (Volumes 1, 2, and 3)
MAINVIEW for IMS Online – Analyzers Reference Manual
MAINVIEW for IMS Online – Monitors and Traces Reference Manual

IBM Publications

OS/390 Initialization and Tuning Guide
IMS Operations Guide

Conventions Used in This Book

The following symbols are used to define command syntax, are ***not*** part of the command, and should never be typed as part of the command:

- Brackets [ ] enclose optional parameters.
- Braces { } enclose a list of parameters; one must be chosen.
- A line | separates alternative options; one can be chosen.
- An <u>underlined</u> parameter is the default.

The following command syntax conventions apply:

- An ITEM IN CAPITAL LETTERS must be typed exactly as shown.
- Items in *italicized*, *lowercase* letters are values that you supply.
- When a command is shown in uppercase and lowercase letters, such as **HSplit**, the uppercase letters show the command abbreviation that you can use (**HS**, for example). The lowercase letters complete the entire command name. Typing the entire command name is an optional, alternative way of entering the command.
- Commands that do not have an abbreviation (**END**, for example) appear in all uppercase letters.

Note: Although MAINVIEW for DBCTL is often referred to as "MVDBC" in this book, the abbreviation is used for brevity only and does not represent a legal product name of BMC Software.

Part 1. Migration Considerations

This part provides information about migrating to the current version of MAINVIEW for DBCTL.

Chapter 1. Migrating to Version 3.3.30 from Version 3.3.20, 3.3.10, or 3.3.00

This chapter describes new options and features that you should review if you are migrating to MAINVIEW for DBCTL (MVDBC) version 3.3.30 from version 3.3.20, 3.3.10, or 3.3.00.

Migrating from Version 3.3.20

This section describes product options and features that were introduced with version 3.3.30 of MVDBC.

New Views for Analyzing Program Isolation Lock Contention

Version 3.3.30 added new views and menus for managing program isolation (PI) lock contention. You can use the new views to quickly recognize and resolve PI resource contention problems. Four of the views (IPIRGLST, IPIRGWT, IPIRSWT, and IPIRSSUM) are used to analyze PI lock contention. The other view (IPISTAT) provides dynamic and internal IMS program isolation pool and QCB statistics.

The PI Region Lock List view (IPIRGLST) shows the regions that are waiting for PI locks, the resource they are waiting for, and the list of "ultimate lock holder" regions that are causing the lock contention. From the IPIRGLST view, you can issue an action that will stop the ultimate lock holder region.

For more information about the new views, see the chapter "Analyzing Program Isolation Lock Contention" in the *MAINVIEW for DBCTL IPSM Reference Manual*.

Support for Analyzing I/O DEDB Performance (IMS 7.1 and later)

With version 3.3.30, MVDBC can provide more DL/I call and I/O activity statistics. The Event Collector can record and time the I/O activity resulting from DL/I calls to DEDB databases, information that is vital for managing DEDB performance.

The DL/I call and I/O activity statistics are available in the IPSM Database Activity (IDA*) views. You can use the views to determine which DEDB databases are performing poorly and to take appropriate actions, such as making application changes, reorganizing databases, and changing database DASD placement.

DEDB read I/O information is available in detail traces.

You can use the new DBFPLVL parameter in BBPARM member IMFECX00 to control the collection of DEDB call and I/O statistics. For more information about the DBFPLVL parameter, see page 51.

New Easy Menus for Database Administrators and Systems Operators

In version 3.3.30, two new Easy Menus were created for database administrators, the DBCTL DBA Easy Menus EZDDBA and EZDDBAR. EZDDBA is for interval statistics, and EZDDBAR is for realtime statistics. The DBA menus provide quick access to database exceptions, locking contention, I/O analysis, buffer pool analysis, pool utilization, transaction trace, and database-related transaction delays.

A new Easy Menu was created for systems operations, the DBCTL Operations Menu EZDOPSR. This realtime menu provides quick access to problems and the commands that are required to correct the problems.

The new menus can be used for a context of one system or a context of multiple systems, and the menus are summarized so that you can issue a single command to change the state of an object in all participating IMS systems.

Enhanced Transaction Trace Views

In version 3.3.30, the Trace List view (ITALIST) displays a separate row for each currently active trace buffer, for each currently active log data set, and for each inactive log data set, and all rows associated with a specific trace are displayed together.

When you hyperlink to the Trace Detail view (ITALISTD) from the ITALIST Start Time field, the ITALISTD view now displays either the contents of a current buffer only or the contents of the selected trace log data set.

The enhanced ITALIST view shows the trace start and stop times, and it indicates whether a row is for trace records in a history log data set (Hist) or in a current buffer (Curr).

A new Trace Menu (ITALMR) provides hyperlinks for examining trace data set details and for viewing a summary of all occurrences of a selected transaction. The menu also provides access to services for managing active and history traces and for executing trace query requests. You can access the ITALMR menu by hyperlinking from the Trace Id field in the ITALIST view.

Message Change

The job name was added to message IM0203I to identify the job that owns the thread.

For a DB/DC environment, the IM0203I message format is:

```
IM0203I EVENT COLLECTOR ACTIVE FOR REGION ID=xxx JOBNAME=yyyyyyyy
```

For a DBCTL environment, the IM0203I message format is:

```
IM0203I EVENT COLLECTOR ACTIVE FOR CICS REGION ID=xxx JOBNAME=yyyyyyyy
```

Migrating from Version 3.3.10

This section describes product options and features that were introduced with version 3.3.20 of MVDBC.

Java and ODBA Region Support

Version 3.3.20 of MVDBC added support for the Java region types and ODBA threads. The IPSM views and the analyzers, monitors, and traces services that show dependent region statistics now display the Java region types and ODBA threads. The parameters used to filter region information include the Java region types and ODBA threads.

Note: Support for Java region types is provided with IMS 7.1 (after application of PTFs UQ61540, UQ61541, and UQ61542) and later.

IMSplex Views

Version 3.3.20 of MVDBC has two new IPSM IMSplex views, IPXSUMR and IMSSPLXR. The IPXSUMR view is a realtime or past interval tabular view that shows the structure and status of IMSplex group members, including both IMS control region members and Structured Call Interface (SCI) component members. The IMSSPLXR view is a realtime or past interval detail view that shows information about the IMSplex connectivity and utilization of a specific IMS system.

The DBCDTL* and DBCPL* views have a new IMSplex group name field that hyperlinks to the IPXSUMR view. Hyperlinks to the IMSplex views were also added to the EZIMS, EZISSI, and EZIFAST menus.

Note: Support for the IMSplex views is provided with IMS 8.1 and later.

Recoverability Status for DEDB Databases

IMS 8.1 added the capability of defining DEDB databases as nonrecoverable. In version 3.3.20 of MVDBC, the IPSM Fast Path DEDB area views (IFP*) and the database views (IDB*) were enhanced to show which DEDB databases are defined as nonrecoverable.

Migrating from Version 3.3.00

This section describes product options and features that were introduced with version 3.3.10 of MVDBC.

Product Authorization

MVDBC version 3.3.10 added support for the use of product authorization tables to activate products on individual CPUs.

Version 3.3.20 continues to support the product keys method of product activation, and you can continue to use the DBC product key to activate MVDBC.

If MVDBC finds the DBC product key during product initialization and the key is valid, MVDBC will not look for a product authorization table. Therefore, if you want to use a product authorization table to activate MVDBC, you must remove the DBC product key from the BBKEYS member of the BBPARM data set.

If no valid product key is found during initialization, MVDBC looks for the MVDBC product authorization table, DBCTBL3P, in the data set pointed to by the BMCPSWD DD statement in the startup JCL for the IMS control region. If the BMCPSWD DD is not present, MVDBC looks in the STEPLIB, JOBLIB, and LNKLST data sets. If the DBCTBL3P table is not located in those data sets, MVDBC looks in the data set pointed to by the BBILOAD DD. If the DBCTBL3P table is found, its authorization information is used to determine whether the CPU is licensed to run MVDBC.

Note: If a product key brings MVDBC up in the IMS control region and a product authorization table brings MVDBC up in the product address space (PAS) *or vice versa*, MVDBC will still work properly as long as the provided authorization is consistent. In other words, you will encounter no initialization problems if you bring up one with a product key and the other with a product authorization table.

The procedure required to use a product authorization table to activate MVDBC is provided on page 29.

For more information about product authorization tables and the Product Authorization utility, see the *OS/390 and z/OS Installer Guide*.

UBBPARM Customization Capabilities

Version 3.3.10 added the capability of using the UBBPARM data set to include unique IMFSYS00 and IMFECP00 parameter members for individual IMS systems, which eliminates the need to create and allocate an *ibbparm* data set for each IMS system. The IMS-specific settings are defined in members that must have the following name formats:

*imsid*SYS	for IMS-specific IMFSYS00 system parameters
*imsid*ECP	for IMS-specific IMFECP00 Event Collector parameters

The name of each IMS-specific parameter member must begin with the identification code for the IMS system.

For more information about the IMS-specific UBBPARM members, see “Setting Up BBPARM Data Sets” on page 23.

Product Initialization Messages

Product initialization messages have changed with this release. If you have automation that depends on the initialization messages or the sequencing of the messages, please see Appendix E on page 89.

Chapter 2. Migrating to Version 3.3.30 from Version 3.2

This chapter describes new options and features that you should review if you are migrating to MAINVIEW for DBCTL (MVDBC) version 3.3.30 from version 3.2.

New Views for Analyzing Program Isolation Lock Contention

Version 3.3.30 added new views and menus for managing program isolation (PI) lock contention. You can use the new views to quickly recognize and resolve PI resource contention problems. Four of the views (IPIRGLST, IPIRGWT, IPIRSWT, and IPIRSSUM) are used to analyze PI lock contention. The other view (IPISTAT) provides dynamic and internal IMS program isolation pool and QCB statistics.

The PI Region Lock List view (IPIRGLST) shows the regions that are waiting for PI locks, the resource they are waiting for, and the list of "ultimate lock holder" regions that are causing the lock contention. From the IPIRGLST view, you can issue an action that will stop the ultimate lock holder region.

For more information about the new views, see the chapter "Analyzing Program Isolation Lock Contention" in the *MAINVIEW for DBCTL IPSM Reference Manual*.

Support for Analyzing I/O DEDB Performance (IMS 7.1 and later)

With version 3.3.30, MVDBC can provide more DL/I call and I/O activity statistics. The Event Collector can record and time the I/O activity resulting from DL/I calls to DEDB databases, information that is vital for managing DEDB performance.

The DL/I call and I/O activity statistics are available in the IPSM Database Activity (IDA*) views. You can use the views to determine which DEDB databases are performing poorly and to take appropriate actions, such as making application changes, reorganizing databases, and changing database DASD placement.

DEDB read I/O information is available in detail traces.

You can use the new DBFPLVL parameter in BBPARM member IMFECX00 to control the collection of DEDB call and I/O statistics. For more information about the DBFPLVL parameter, see page 51.

New Easy Menus for Database Administrators and Systems Operators

In version 3.3.30, two new Easy Menus were created for database administrators, the DBCTL DBA Easy Menus EZDDBA and EZDDBAR. EZDDBA is for interval statistics, and EZDDBAR is for realtime statistics. The DBA menus provide quick access to database exceptions, locking contention, I/O analysis, buffer pool analysis, pool utilization, transaction trace, and database-related transaction delays.

A new Easy Menu was created for systems operations, the DBCTL Operations Menu EZDOPSR. This realtime menu provides quick access to problems and the commands that are required to correct the problems.

The new menus can be used for a context of one system or a context of multiple systems, and the menus are summarized so that you can issue a single command to change the state of an object in all participating IMS systems.

Enhanced Transaction Trace Views

In version 3.3.30, the Trace List view (ITALIST) displays a separate row for each currently active trace buffer, for each currently active log data set, and for each inactive log data set, and all rows associated with a specific trace are displayed together.

When you hyperlink to the Trace Detail view (ITALISTD) from the ITALIST Start Time field, the ITALISTD view now displays either the contents of a current buffer only or the contents of the selected trace log data set.

The enhanced ITALIST view shows the trace start and stop times, and it indicates whether a row is for trace records in a history log data set (Hist) or in a current buffer (Curr).

A new Trace Menu (ITALMR) provides hyperlinks for examining trace data set details and for viewing a summary of all occurrences of a selected transaction. The menu also provides access to services for managing active and history traces and for executing trace query requests. You can access the ITALMR menu by hyperlinking from the Trace Id field in the ITALIST view.

Message Change

The job name was added to message IM0203I to identify the job that owns the thread.

For a DB/DC environment, the IM0203I message format is:

`IM0203I EVENT COLLECTOR ACTIVE FOR REGION ID=`*`xxx`*` JOBNAME=`*`yyyyyyyy`*

For a DBCTL environment, the IM0203I message format is:

`IM0203I EVENT COLLECTOR ACTIVE FOR CICS REGION ID=`*`xxx`*` JOBNAME=`*`yyyyyyyy`*

Java and ODBA Region Support

Version 3.3.20 of MVDBC added support for the Java region types and ODBA threads. The IPSM views and the analyzers, monitors, and traces services that show dependent region statistics now display the Java region types and ODBA threads. The parameters used to filter region information include the Java region types and ODBA threads.

Note: Support for Java region types is provided with IMS 7.1 (after application of PTFs UQ61540, UQ61541, and UQ61542) and later.

IMSplex Views

Version 3.3.20 of MVDBC has two new IPSM IMSplex views, IPXSUMR and IMSSPLXR. The IPXSUMR view is a realtime or past interval tabular view that shows the structure and status of IMSplex group members, including both IMS control region members and Structured Call Interface (SCI) component members. The IMSSPLXR view is a realtime or past interval detail view that shows information about the IMSplex connectivity and utilization of a specific IMS system.

The DBCDTL* and DBCPL* views have a new IMSplex group name field that hyperlinks to the IPXSUMR view. Hyperlinks to the IMSplex views were also added to the EZIMS, EZISSI, and EZIFAST menus.

Note: Support for the IMSplex views is provided with IMS 8.1 and later.

Recoverability Status for DEDB Databases

IMS 8.1 added the capability of defining DEDB databases as nonrecoverable. In version 3.3.20 of MVDBC, the IPSM Fast Path DEDB area views (IFP*) and the database views (IDB*) were enhanced to show which DEDB databases are defined as nonrecoverable.

Product Authorization

MVDBC version 3.3.10 added support for the use of product authorization tables to activate products on individual CPUs.

Version 3.3.20 continues to support the product keys method of product activation, and you can continue to use the DBC product key to activate MVDBC.

If MVDBC finds the DBC product key during product initialization and the key is valid, MVDBC will not look for a product authorization table. Therefore, if you want to use a product authorization table to activate MVDBC, you must remove the DBC product key from the BBKEYS member of the BBPARM data set.

If a valid product key is not found during initialization, MVDBC looks for the MVDBC product authorization table, DBCTBL3P, in the data set pointed to by the BMCPSWD DD statement in the start-up JCL for the IMS control region. If the BMCPSWD DD is not present, MVDBC looks in the STEPLIB, JOBLIB, and LNKLST data sets. If the DBCTBL3P table is not located in those data sets, MVDBC looks in the data set pointed to by the BBILOAD DD. If the DBCTBL3P table is found, its authorization information is used to determine whether the CPU is licensed to run MVDBC.

Note: If a product key brings MVDBC up in the IMS control region and a product authorization table brings MVDBC up in the product address space (PAS) *or vice versa*, MVDBC will still work properly as long as the authorization provided is consistent. In other words, you will encounter no initialization problems if you bring up one with a product key and the other with a product authorization table.

The procedure required to use a product authorization table to activate MVDBC is provided on page 29.

For more information about product authorization tables and the Product Authorization utility, see the *OS/390 and z/OS Installer Guide*.

Detail Trace

Three levels of information are collected by detail trace: CALLS, I/O, and DATA. The DATA level was added in version 3.3.00, and it includes segment search argument (SSA), key feedback area (KFB), and I/O area (IOA) data. Inclusion of the SSA, KFB and IOA data can significantly affect detail trace buffer usage. To help prevent potential buffer size problems, a maximum of 66 bytes is allocated to each of the three record types. The fact that a single call can have multiple SSAs, however, can cause the buffers to fill quickly.

You can choose not to collect the SSA, KFB, and IOA data by setting TRDATA=NO in the BBPARM member IMFBEX00.

Event Collector Option

A new FEATURE=NOEC parameter option was added in version 3.3.00 to allow you to disable the Event Collector if the MVDBC components you use can function without it.

The only MVDBC components that can function without the Event Collector are the Resource Analyzer and the Resource Monitor. If you use only the Resource Analyzer and Resource Monitor components and you want to prevent Event Collector initialization, you can set FEATURE=NOEC in the PARMLIB member IMFSYS00.

Note: Refer to PARMLIB member IMFSYSBB for additional information.

UBBPARM Customization Capabilities

Version 3.3.10 added the capability of using the UBBPARM data set to include unique IMFSYS00 and IMFECP00 parameter members for individual IMS systems, which eliminates the need to create and allocate an *ibbparm* data set for each IMS system. The IMS-specific settings are defined in members that must have the following name formats:

*imsid*SYS	for IMS-specific IMFSYS00 system parameters
*imsid*ECP	for IMS-specific IMFECP00 Event Collector parameters

The name of each IMS-specific parameter member must begin with the identification code for the IMS system.

For more information about the IMS-specific UBBPARM members, see "Setting Up BBPARM Data Sets" on page 23.

Product Initialization Messages

Product initialization messages were changed for versions 3.3.10 and 3.3.00, and new messages were added for version 3.3.00. If you have automation that depends on the initialization messages or the sequencing of the messages, please see Appendix E on page 89.

Easy Menus

The EZIMS, EZIFAST, and EZISSI Easy Menus were redesigned in version 3.3.00 to improve access to product functions and to incorporate the new features of this release.

Chapter 3. Release Compatibility

This chapter lists the products packaged with the current BBI release and describes MAINVIEW for DBCTL (MVDBC) compatibility with previous MVDBC releases.

Compatibility with BBI

MVDBC and the following products are packaged with the current BBI 2.6.0 release:

MAINVIEW AutoOPERATOR
MAINVIEW for CICS
MAINVIEW FOCAL POINT
MAINVIEW for DB2
MAINVIEW for IMS Online and Offline

Compatibility with Previous Release Levels of MVDBC

The UAS (user address space) must be at the same release and service level as the BBI-SS PAS (product address space) and CAS (coordinating address space) to which it connects.

Downward Compatibility

You can access a previous MVDBC version from a UAS running the current version, but the new features and functions of the current version will not be available.

Help panels for services from a previous release might not be accessible from a UAS at the current release level.

Upward Compatibility

You can access the current version of MVDBC from a UAS running an earlier version. However, if a current version of a service was changed to support a new IMS function, that service may not be available to you.

The help panels for services added with a new MVDBC version and the help panels modified for revised services cannot be accessed from a UAS at a earlier version level.

To review a list of features or services that are new or have changed with the current version of MVDBC, see the release notes, which are available on the BMC Software Web site at **http://www.bmc.com/support.html**.

AO Exit

If a previous version of the BMC Software AO exit was installed on the target IMS, make sure that all remnants of the previously installed version are removed by following the instructions in the section “Deleting Modules from a Prior Release” on page 22.

Part 2. Customizing MAINVIEW for DBCTL Functions

This part describes how to tailor MAINVIEW for DBCTL to your site's needs.

Chapter 4. Introduction to DBCTL

DBCTL is a feature of IMS that allows CICS to access DL/I databases through a separate subsystem instead of having all the IMS code and buffers in the CICS region. The DBCTL subsystem is very similar to an IMS DB/DC subsystem, and consists of three or four address spaces: a control address space, DLISAS, DBRC, and, optionally, ILRM for locking. DBCTL is essentially an IMS subsystem without IMS transactions and message queues.

Because DBCTL subsystem functions are the same as IMS DB/DC subsystem functions, many of the performance monitoring services provided by the MAINVIEW for IMS Online are also valid for DBCTL, including services that monitor

- PSB and DMB pools
- application (region) activity and DL/I status
- database buffer pools
- logging
- program isolation or IRLM locking

IMS treats DBCTL threads as regions. DBCTL CICS threads are identified as region type DBT throughout the MAINVIEW for DBCTL library. DBCTL ODBA threads are sometimes included in the DBT region type, but in other cases they are identified as region type ODB.

Chapter 5. Resource Utilization

This chapter summarizes the options that can be used to control usage of CPU by

- the Event Collector
- IPSM samplers

For more information about the Event Collector options, see Chapter 9, “Using the Event Collector” on page 45. For more information about options that can be used to control IPSM sampler usage of CPU, see the *MAINVIEW for DBCTL IPSM Reference Manual.*

CPU Usage – Event Collector

The Event Collector is always active, even when there are no active traces or workload monitors. The default Event Collector option, CPU=DEP, provides a reasonable balance between CPU usage and the detail level of the data the Event Collector collects. If the default option results in too much CPU usage for your site, you can change the CPU option in BBPARM member IMFECP00 to reduce CPU usage.

The CPU options that control Event Collector data collection are listed below in order of highest to lowest to CPU usage.

CPU=ALL **(highest CPU usage)**
: Collects all CPU time data

CPU=DEP
: Collects CPU time data from dependent regions.

CPU=DEPDB2
: Functions the same as the DEPPGM default option except that CPU for DB2 events is reported separately from dependent region CPU time. CPU usage depends upon the number of SQL calls issued by the transaction.

CPU=DEPPGM (the default)
: Collects CPU time data only from dependent region activities, including DB2 CPU time, and times the entire transaction as a single event. This option reduces overhead significantly.

CPU=NONE **(lowest CPU usage)**
: CPU time data is not collected.

For more information about how the CPU options work, see “Event Collector Parameters” on page 45.

Note: The Resource Analyzer and Resource Monitor components can function without the Event Collector. If you use only the Resource Analyzer and Resource Monitor components and you want to prevent Event Collector initialization, you can set FEATURE=NOEC in the PARMLIB member IMFSYS00. (For more information, see the PARMLIB member IMFSYSBB.)

CPU Usage – IPSM Samplers

IPSM collects target system sampling for the components of response time (CORT) and workflow views. Samplers must run continually in the BBI-SS PAS if you want to collect this information. The distributed sampler default provides for a sampling period of 24 hours a day at 2 times per second.

If the default results in too much CPU usage for your site, you can reduce CPU usage by creating and adding a sampler target definition. For more information, see the *MAINVIEW for DBCTL IPSM Reference Manual*.

CSA Usage

The *MAINVIEW Installation Requirements Guide* lists the virtual storage requirements for all MAINVIEW products.

Chapter 6. Implementing MAINVIEW Products in IMS

This chapter describes how to install and implement BMC Software modules, which are required for

MAINVIEW AutoOPERATOR for IMS

MAINVIEW for DBCTL (MVDBC)

Follow the procedures in this chapter if you are customizing one or both of these products and you have not performed AutoCustomization. AutoCustomization tailors your products automatically.

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

Note: If you have multiple IMSs, you may want to allocate a BBPARM data set that is unique to an IMS, as described in “Setting Up BBPARM Data Sets” on page 23 and “Modifying the IMS Control Region JCL” on page 28.

For information about how to use product libraries, including parameter libraries (BBPARM and UBBPARM) and sample libraries (BBSAMP and UBBSAMP), see the *MAINVIEW Common Customization Guide* or the *MAINVIEW Administration Guide*.

Deleting Modules from a Prior Release

You can skip this section if you are installing MVDBC for the first time.

If a previous release of MAINVIEW for IMS, MAINVIEW for DBCTL, or MAINVIEW AutoOPERATOR for IMS was installed on the target IMS, use the information in Table 1 and follow the instructions below the table.

If you copied BBLINK members to a STEPLIB data set using ICOPY, you can use the following BBSAMP jobs to delete the old modules:

Table 1. BBSAMP Jobs to Delete Old Modules

IMS Release	MVIMS 3.1 MVDBC 2.1 AO 3.1, 4.1	MVIMS 3.2 MVDBC 3.2 AO 4.1	MVIMS 3.3.mm MVDBC 3.3.mm AO 5.1, 6.1, 6.2
IMS 5.1	IDEL31$5	IDEL32$5	IDEL33$5
IMS 6.1	IDEL31$6	IDEL32$6	IDEL33$6
IMS 7.1	n/a	n/a	IDEL33$7
IMS 8.1	n/a	n/a	IDEL33$8

To delete the old modules:

1. From Table 1, select the delete job that corresponds to the BMC Software product (or products) and IMS release installed at your site.
2. Edit the delete job you selected to change all &RESLIBs to the name of the data set where the old modules reside.
3. Run the delete job.

If you added BBLINK to the IMS STEPLIB concatenation, replace the BBLINK data set in the current IMS STEPLIB concatenation with the new BBLINK data set.

If you included BBLINK in the link list concatenation, replace the BBLINK data set in the current link list concatenation with the new BBLINK data set. This data set replacement will affect all the IMS address spaces running in that OS/390.

Setting Up BBPARM Data Sets

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

hilevel.ibbparm
hilevel.UBBPARM
hilevel.BBPARM

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

ibbparm A user-defined parameter data set that is unique to this IMS

You can allocate a separate *ibbparm* data set to contain any members that you want to make unique to the IMS, such as IMFSYS00 and IMFECP00. The *ibbparm* data set must be allocated; it is not created through AutoCustomization.

Note: If the only members that require customization for an IMS are IMFSYS00, IMFECP00, or both, an *ibbparm* data set is not required. Instead, you can create a renamed (*imsid*SYS) version of the IMFSYS00 member, a renamed (*imsid*ECP) version of the IMFECP00 member, or both, include them in UBBPARM, and customize them to suit your needs (as explained below in UBBPARM).

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

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

In the UBBPARM data set, you can include a copied, renamed version of IMFSYS00 to customize the system parameters for an IMS system, including the SUBSYS parameter, which establishes communication between the IMS and the BBI-SS PAS. The name of an IMS-specific system parameter member must be in the following format:

*imsid*SYS (where *imsid* is the IMS ID code)

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

*imsid*ECP (where *imsid* is the IMS ID code)

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

BBPARM The target BBPARM data set distributed by BMC Software

Important

Throughout the MVDBC books, parameter library members are normally referred to as BBPARM members, even though customized versions of the members may reside in the UBBPARM data set or in an *ibbparm* data set.

The system parameter member is normally referred to as BBPARM member IMFSYS00, and the Event Collector parameter is normally referred to as BBPARM member IMFECP00, even though the members may reside in a parameter data set with a member name in the format *imsid*SYS or *imsid*ECP.

Customizing BBPARM Members for an IMS System

This section is provided primarily for new installations of MVDBC. If you have a prior version installed, you can use your existing parameter settings, but this would be a good time to review them.

This following procedure customizes the parameter members required to

- establish communication between an IMS region and the BBI-SS PAS
- run the Event Collector to collect workload monitor, trace, and wait data
- run the batch jobs used for report printing

1. Set the system parameters in an *imsid*SYS member in the UBBPARM data set or in an IMFSYS00 member in an *ibbparm* data set. The parameters and their uses are as follows:

   SUBSYS. Use this parameter to identify the subsystem name of the BBI PAS that the IMS region should communicate with. The subsystem should be the same as the one specified in the BBIJNT00 member of the UBBPARM data set or an *ibbparm* data set.

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

   KEYWARN. Use this parameter to set a minimum number of days before expiration warning messages are issued for product keys. The default is 45 days. (The KEYWARN parameter applies to MVIMS Online, MVIMS Offline, and MVDBC only, and it is used only in the IMS control region, not the BBI-SS PAS.)

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

   AOEEXIT and AOIEXIT. Use these parameters to specify the names of user-written AO exit routines. You can also use AOEEXIT and AOIEXIT to specify the order in which AO exit routines get control and whether a return code is padded. See “Enabling AO Exit Routines” on page 30 for more information.

   As distributed, IMFSYS00 activates all MVIMS components in IMS. You can use *imsid*SYS in the UBBPARM data set or IMFSYS00 in an *ibbparm* data set to temporarily deactivate one or more components. BBPARM member IMFSYSBB contains information about how to deactivate components. Copy what you need from IMFSYSBB to your UBBPARM *imsid*SYS member or *ibbparm* IMFSYS00 member.

2. Set up the Event Collector data collection parameters in an IMFECP00 member or an *imsid*ECP member.

   Note: Use UBBPARM member IMFECP00 for Event Collector parameters that are to be shared among multiple IMSs. To specify IMS-specific Event Collector parameters, use an *imsid*ECP member in the UBBPARM data set or an IMFECP00 member in an *ibbparm* data set.

   The following parameters and recommended values set up Event Collector data collection:

   To record BMP and JBP data, specify

   `BMP=YES` (default)

   To time an entire transaction as a single event, specify

   `CPU=DEPPGM` (default)

   To write DBCTL transaction records to the IMS log and send them to the BBI-SS PAS for Workload Analyzer wait and trace processing and for Workload Monitor processing, specify

   `CICS=YES` (default)

3. Set up the Log Edit utility parameters in the IMFLEP00 member.

   Note: Use UBBPARM member IMFLEP00 for Log Edit utility parameters that are to be shared by multiple IMSs. For IMS-specific Log Edit utility parameters, use the IMFLEP00 member in an *ibbparm* data set.

   The following parameter in IMFLEP00 is required only if you have MAINVIEW for IMS Offline installed. The parameter is used by the Log Edit utility (IMFLEDIT).

   To specify the release level of the IMS system where the IMS log is created, specify

   `IMSLEVEL=5100|6100|7100|8100|0000`

   where

   `5100` IMS 5.1
   `6100` IMS 6.1
   `7100` IMS 7.1
   `8100` IMS 8.1
   `0000` Causes the Log Edit utility to scan up to the first 50K records of the log tape to determine the IMS release

Note: For more information about the Event Collector parameters and their use, see Chapter 9 on page 45. For more information about the Log Edit utility, IMFLEDIT, see the Log Edit chapter in the *MAINVIEW for IMS Offline – Customization and Utilities Guide*.

Activating MAINVIEW for DBCTL

To activate MVDBC, you need to enable product initialization and product authorization.

Enabling Product Initialization

To enable MVDBC product initialization, in BBPARM member BBISSP00, specify

```
PRODUCT=MVDBC
```

The `PRODUCT=MVDBC` parameter setting initializes all MVDBC product components at BBI-SS PAS startup.

Enabling Product Authorization

You can use product keys or a product authorization table to enable MVDBC product authorization.

Product Keys

If you decide to use product keys to enable MVDBC activation, you can continue to use the DBC product key.

For detailed information about specifying product keys, see the step called “Specify Product Option Password Keys” in the *MAINVIEW Common Customization Guide*.

Product Authorization Table

If you decide to use a product authorization table to activate the MVDBC components, complete the following steps:

1. Create the product authorization table DBCTBL3P, either through AutoCustomization or with a batch job. Both methods are described in the *OS/390 and z/OS Installer Guide*.
2. Add access to the product authorization table (as described on page 29).

Note that the product initialization messages will now include BBAP prefixed messages issued by the BMC License Manager.

Modifying the IMS Control Region JCL

The following sections describe how to allocate a BBPARM data set, provide IMS access to the Event Collector, and add access to a product authorization table.

Allocating a BBPARM Data Set

To establish the parameters for MVDBC and MAINVIEW AutoOPERATOR for IMS, allocate the BBPARM data sets by adding the following statements to the IMS control region startup procedure:

```
//IMFPARM  DD DSN=hilevel.ibbparm
//         DD DSN=hilevel.UBBPARM
//         DD DSN=hilevel.BBPARM
```

Note: If the only members that require customization for an IMS are IMFSYS00, IMFECP00, or both, an *ibbparm* statement is not required. For more information, see page 23.

Giving IMS Access to the Event Collector

The IMS control region must be able to access the Event Collector to enable execution of MVDBC and MAINVIEW AutoOPERATOR for IMS.

If the BBLINK data set is in the LNKLST concatenation, Event Collector access is already established. If not, you can provide Event Collector access by modifying your IMS control region JCL or by copying the required modules to the site authorized library (such as IMS RESLIB).

To modify the IMS control region JCL, add the BBLINK data set to the IMS STEPLIB concatenation. (BBLINK must be authorized.)

To copy the modules used in IMS, copy the individual BBLINK members to an authorized STEPLIB data set. Select one of the following jobs in BBSAMP:

ICOPY8 for IMS 5.1

ICOPY9 for IMS 6.1

ICOPY1 for IMS 7.1

ICOPY2 for IMS 8.1

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

Adding Access to a Product Authorization Table

If you are using the DBCTBL3P product authorization table to activate MVDBC, the IMS control regions must be able to access the table.

There are four ways you can provide access to the table:

- Concatenate a data set containing the DBCTBL3P product authorization table to the STEPLIB data set in the IMS control region JCL.
- Add the following statement to the IMS control region JCL:

  ```
  //BMCPSWD  DD DSN=hilevel.BMCPSWD,DISP=SHR
  ```

- Copy the DBCTBL3P table into RESLIB or BBLINK.
- Add the DBCTBL3P data set to LNKLST.

If you want the product authorization table to reside in RESLIB, follow the instructions in the "Product Maintenance or Version Upgrades" section in Appendix A of the *OS/390 and z/OS Installer Guide*.

Note: If you previously used BBKEYS to activate the products, you must remove all keys for MVDBC from the BBKEYS member of BBPARM.

Modifying DLISAS Region JCL

This step modifies the DLISAS region JCL to provide access to BBLINK modules for database activity monitoring in the IMSPlex System Manager (IPSM) component.

Make BBLINK modules available to the DLISAS region by adding the BBLINK data set to the DLISAS STEPLIB concatenation (BBLINK must be authorized).

Enabling AO Exit Routines

You can skip this section if you have a prior version of MVDBC installed.

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

BMC Software supplies two AO exit routines to do message processing: a type-1 AO exit routine, DFSAOUE0, and a type-2 AO exit routine, DFSAOE00.

Note: See IBM's *IMS Operations Guide* for an explanation of how the DFSAOUE0 and DFSAOE00 routines differ.

The DFSAOUE0 and DFSAOE00 routines capture MTO messages and IMS commands and pass them to MAINVIEW AutoOPERATOR for IMS. They also perform the following functions:

- provide an interface with your AO exit routines (if any)
- automatically start up MVDBC monitors when IMS starts up
- provide an interface between the Event Collector and the BBI-SS PAS for workload data collection
- initialize the Event Collector

On entry to DFSAOE00 with AOE0FUNC=1, MVIMS loads and executes any user exit specified in AOEINIT=*xxxxxxxx* one time only. The DFSAOE00 exit does not forward control to the user's DFSAOE01 exit on the initialization call.

If you require a DFSAOUE0 or DFSAOE00 exit routine in addition to the exit provided by BMC Software, perform Step 1 and Step 2 below.

1. If you have

   a. **One DFSAOUE0 exit:** Rename it DFSAOUE1, or choose a different name and use the control statements described in Step b.

   b. **Multiple DFSAOUE0 exits in addition to the** BMC Software**–supplied exit, or one or more exits not named DFSAOUE1:** You must add one or more of the following AOIEXIT control statements to an *imsid*SYS member in UBBPARM or to an IMFSYS00 member in an *ibbparm* data set.

      For example:

      ```
      AOIEXIT=MYNAME
      AOIEXIT=DFSAOUE2
      ```

      In the example above, both exits are loaded and executed by the BMC Software DFSAOUE0 exit. Program MYNAME executes first.

2. If you have

   a. **One DFSAOE00 exit:** Rename it DFSAOE01, or choose a different name and use the control statements described in Step b.

   b. **Multiple DFSAOE00 exits in addition to the** BMC Software**–supplied exit, or one exit not named DFSAOE01:** You must add one or more of the following AOEEXIT control statements to an *imsid*SYS member in UBBPARM or to an IMFSYS00 member in an *ibbparm* data set.

      For example:

      ```
      AOEEXIT=MYNAME
      AOEEXIT=DFSAOE02
      ```

      In the example above, both exits are loaded and executed by the BMC Software DFSAOE00 exit. Program MYNAME executes first.

Flow of Control

By default, the BMC Software DFSAOE00 exit invokes the BMC Software AO exit routine (IELOAD) before it invokes your user exits. The return code set by the last user exit processed is passed to IMS. You can change the default processing in an *imsid*SYS member in the UBBPARM data set or an IMFSYS00 member in an *ibbparm* data set by specifying

- the order that exits are to be processed
- the exit return code

For example:

```
AOEEXIT=(DFSAOE01,RC)
AOEEXIT=IELOAD
```

In the example above, the DFSAOE01 user exit assumes control before the IELOAD exit and the DFSAOE01 return code is passed to IMS.

IMS then executes the DFSAOUE0 exit unless indicated otherwise by the return code set by the DFSAOE00 exit. The BMC Software DFSAOUE0 exit calls your exits in the order you specified with the AOIEXIT control statements. By default, the return code set by the last user exit processed is passed to IMS. You can pass the return code from another exit by specifying the RC parameter with the AOIEXIT control statement as shown in the following example:

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

Enhancement to the IMS DFSAOE00 Interface

BMC Software enhances the programming interface when your DFSAOE00 exit is invoked in the following ways:

- Register 11 contains the address of the IMS SCD.
- Register 13 contains the address of 15 prechained save areas.

MAINVIEW AutoOPERATOR for IMS Considerations

When you install your user exit with a BMC Software exit, you should consider the following information:

- If the BMC Software exit is executed before the user exit and the user exit's return code cancels the processing of additional message segments of a multisegment message, the message segments are also canceled for MAINVIEW AutoOPERATOR. The result is that incomplete IMS messages are passed to the MAINVIEW AutoOPERATOR and LAST SEG LOST messages from BBI. The MAINVIEW AutoOPERATOR AO exit holds the first segment for a certain length of time while waiting for additional segments, which can cause the messages to be processed out of timestamp sequence by MAINVIEW AutoOPERATOR.
- When the user exit is executed before the BMC Software exit, changes to the messages made by the user exit are received by the MAINVIEW AutoOPERATOR AO exit.
  - If the user exit sets the length code of a message (or a segment of a multisegment message) to zero, MAINVIEW AutoOPERATOR does not process the message (or segment).
  - If the user exit sets the length code of the first segment to 0 and the return code to 4, all further segments are canceled for MAINVIEW AutoOPERATOR also.

Note: If any modules are specified with the AOIEXIT parameter in IMFSYS00 or *imsid*SYS, but the BMC Software exit is not specified, the exit still executes after all specified exits.

The MAINVIEW AutoOPERATOR AO exit and any other AO exit you use are under ESTAE protection when invoked by the BMC Software routines. If an abend occurs in one of these exits, only that routine is disabled; the other routine and IMS itself are not affected.

Chapter 7. Customizing the Trace Facility

This chapter describes how to manually

- specify trace defaults in BBPARM member IMFBEX00
- set up and maintain a trace directory and trace log data sets

A request for a trace can request that the trace data be recorded to VSAM data sets for later viewing or printing. The logging of trace data requires a pre-allocated trace directory that must be identified to the BBI-SS PAS. Setup of the trace directory can be done automatically by AutoCustomization, as described in the *OS/390 and z/OS Installer Guide.* An alternative is to set up the trace directory manually, as described in this section. Trace log data sets can be pre-allocated, as described here, or they can be dynamically allocated at the time of the trace request, as described in the *MAINVIEW for DBCTL Analyzers, Monitors, and Traces Reference Manual*.

Tuning Tip: If one or more summary traces are defined to start automatically, a continuous workload history is available for later viewing.

Specifying Trace Defaults in IMFBEX00

The IMFBEX00 member of the BBPARM data set defines trace request defaults. These defaults prime the options for the trace request data entry panels.

This section describes the IMFBEX00 keyword parameters that define option defaults for

- all trace requests
- trace logging

Detail Trace Data Collection

Three levels of information are collected by detail trace: CALLS, I/O, and DATA. The DATA level includes segment search argument (SSA), key feedback (KFB), and I/O area (IOA) data. Inclusion of the SSA, KFB and IOA data can significantly affect detail trace buffer allocation. To help prevent potential buffer size problems, a maximum of 66 bytes is allocated to each of the three record types. The fact that a single call can have multiple SSAs, however, can cause the buffers to fill quickly.

You can choose not to collect the SSA, KFB, and IOA data by setting TRDATA=NO in BBPARM member IMFBEX00.

Trace Display Buffer Size

The following parameter defines the default for the trace display buffer size (STORAGE option). This option applies for any trace and is presented when a trace is requested.

STORAGE — Specifies the size of the display buffer for the requested trace. This value overrides any value defined in BBIISP00.

MTRAC stores trace data in the private storage area of the BBI-SS PAS address space in extended private storage. This should be considered when setting up storage requirements for the BBI-SS PAS (see the *MAINVIEW Common Customization Guide* for a description of the storage requirements). When the available allocated area is full, the newest data wraps around and overlays the oldest data unless WRAP=NO is specified with the MTRAC request.

The size of the trace data storage area in the BBI-SS PAS can be specified with the STORAGE parameter in the MTRAC request data entry panel; for example:

```
STORAGE ==> 100K
```

requests a GETMAIN storage of 100K. The default can be specified in either the IMFBEX00 or the BBIISP00 member of the BBI-SS PAS BBPARM data set. IMFBEX00 has priority over BBIISP00.

A summary trace entry requires a minimum of 776 bytes of storage per transaction. The largest amount of storage that can be used is 24,776 bytes. The amount of storage required depends on the number of database trailers used by the Event Collector for the transaction.

Specifying a larger storage value prevents frequent wrapping.

A detail trace generally requires a much larger storage area than a summary trace; 52 bytes are required per detail line. So a detail trace of a transaction with 20 detail lines requires about 20 * 52 + summary trace bytes of storage.

Trace Duration

The following parameter defines the default for the trace duration (STOP option). This option applies for any trace and is presented when a trace is requested.

TRTIME=*n* Specifies the default duration of a trace in minutes (1 to 32,000). The default is no limit.

Note: If TRTIME is specified, the STOP parameter in the MTRAC Start IMS Trace Request data entry panel is primed with this value. If a value is not specified, STOP is not primed. A STOP value that is not in the `hh:mm:ss` format is interpreted as a STOPCNT value in minutes.

Trace Logging Options

The following keyword parameters define the defaults for a trace log data set allocation request:

TRPREFIX
: Defines the data set name prefix for trace log data sets if the value for the Log DSN option on the Start IMS Trace Request panel is specified without quotation marks.

  If a value for TRPREFIX is not defined, the ID of the user requesting the trace is used.

TRREUSE
: Requests data to be overwritten if a log data set is not reset. N (NO) indicates that data is not to be overwritten. The default is Y (YES).

  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, previous data recorded in the log is overwritten.

TRVOLS=(*x*,*x*...)
: Specifies the ID of the default volume(s) for trace log data set allocation. Up to seven volumes can be specified. The default value specified in IMFBEX00 with the TRVOLS parameter is SYSDA.

TRCYL
: Defines the primary allocation default in cylinders (CYLS option) for trace log data sets. The default value is 3.

TRSUFFIX
: Defines the default suffix to add to the name of the trace cluster data set (Data DSN Suffix option) to make the data set name for the data component. The default value is D.

TRSMSSCL
: Defines the default name of the SMS storage class for trace log data set allocation. There is no default value.

TRSMSDCL
: Defines the default name of the SMS data class for trace log data set allocation. There is no default value.

TRSMSMCL
: Defines the default name of the SMS management class for trace log data set allocation. There is no default value.

Updating Trace Defaults

You can refresh the parameters specified in BBPARM member IMFBEX00 by starting or stopping the BBI-SS PAS or by using the following command:

```
.RESET PARM IMFBEX00
```

This command refreshes IMFBEX00 only. For more information about the RESET command, see the *MAINVIEW Administration Guide*.

Specifying Detail Trace Buffers

The following keyword parameters define the default for the TRBUFF and TRSIZE data collection buffers:

TRBUFF=*nnn* Specifies the total number of detail trace buffers to be allocated. The number should be at least equal to the number of concurrent active regions plus two. (This value overrides any value defined in BBIISP00.)

TRSIZE=*nnn* Specifies the size of each buffer in bytes. You can specify the value as *nnn* or as *nn*K. The number is rounded to a multiple of 1K. (This value overrides any value defined in BBIISP00.)

For example, if `TRBUFF=20` and `TRSIZE=32K`, total size is 20 * 32K = 640K.

Note: The detail trace is truncated if more events are being traced than can fit in one buffer. To trace long-running batch programs, you may need to increase TRSIZE.

A detail trace stores trace data for active transactions in ECSA buffers. The number and size of the detail trace buffers can be specified in either the IMFBEX00 or the BBIISP00 member of the BBPARM data set. IMFBEX00 has priority over BBIISP00. The defaults are

```
TRBUFF=10
TRSIZE=4
```

TRBUFF defines the number of detail trace buffers. TRSIZE defines the size of each trace buffer. When an active transaction completes, the contents of its buffer are moved to the trace areas in the BBI-SS PAS and can then be displayed by using the DTRAC service.

A pool of buffers is GETMAINed in ECSA when a detail trace is activated. The pool is shared if multiple detail traces are activated. Then buffers are dynamically allocated to the dependent regions as needed. The buffers are returned to the pool when the trace areas in the BBI-SS PAS are updated. If, during transaction initialization, one of the buffers cannot be obtained, only a summary record is generated for the transaction. The buffer pool is FREEMAINed either when the last detail trace stops (and the program running in each region which has a detail trace buffer allocated terminates) or when the target system terminates. The buffers in ECSA are obtained only if a detail trace is activated.

Note: To calculate an appropriate TRSIZE, estimate about 16K per 100 DL/I or SQL calls. The recommended value for TRBUFF is the maximum number of IMS dependent regions running concurrently plus two. If there are not enough detail trace buffers, only a summary trace record is created.

Setting Up and Maintaining a Trace Directory

You can bypass this step if you used AutoCustomization.

Before a request for trace logging can be started, a trace directory must be pre-allocated and initialized. This section describes how to set up the trace log directory using sample members in the BBSAMP data set and BBPARM member BBIISP00.

Note: If a security management system is installed, you may need to grant BBI-SS PAS authorization to allocate trace log data sets dynamically (see *Implementing Security for MAINVIEW Products*).

Defining and Initializing a Trace Directory Data Set

There is one trace directory per BBI-SS PAS. The trace directory is a VSAM linear data set containing one entry for each trace log data set. Each entry indicates the date and time of data set creation, the current status of the data set, the trace target, and other related information. Entries can be added to or deleted from the directory to allow trace logs to be moved between systems.

To define and initialize the trace directory, use BBSAMP sample member JXT001. Follow these steps:

1. Add your job statement.
2. Update the symbolics as necessary.
3. Submit the job.

Identifying the Trace Directory to BBI

To identify the trace directory to BBI, use BBPARM member BBIISP00 and specify

```
TRDIR=dsn
```

The value `dsn` represents the data set name of a trace log data set directory (there is no default name). The directory must be allocated and initialized before any trace can be started with trace logging. BBSAMP member JXT001 creates the trace directory.

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. So, you may occasionally need to synchronize the trace directory information with the actual status of the data sets.

To verify, purge, or print directory entries, use BBSAMP member JXT003. This member checks for the existence of a trace log data set in the system catalog.

Note: Since every entry in the trace directory is allocated dynamically and read to verify its current status, this process could run for some time.

To synchronize trace directory information with the actual status of the data sets, follow these steps:

1. Add your job statement.
2. Update the symbolics as necessary.
3. Specify one of the following processing options for PARM. If PARM is not specified, no action is taken.

   Blank — Causes an uncataloged entry to be marked as INV (INVALID). The same thing happens when PARM is not specified. (Blank is the default).

   ARCVOL= — Specifies an archive volume serial number. This value is matched against the volume serial number in the system catalog for each entry in the directory. If there is a match, the data set is not verified. You can use this option to bypass recalling all trace log data sets from archives.

   LIST — Lists the directory entries that are changed. If NOVERIFY is specified or implied, all entries are listed (equivalent to LISTALL).

   LISTALL — Lists all entries.

   NOLIST — Does not list changed entries.

   PURGE — Deletes any data sets in the directory that are invalid trace data sets.

   NOPURGE — Does not delete invalid data sets (marked as INV (invalid) in the directory).

   VERIFY — Verifies each of the entries in the trace directory.

   Note: If VERIFY is specified, the defaults are LIST, WRITE, and NOPURGE.

   NOVERIFY — Does not verify entries in the trace directory.

   WRITE — Updates trace directory with status changes.

   NOWRITE — Does not update trace directory with changes detected.

4. Submit the job.

Managing Trace Log Data Sets

This section describes how to create and manage trace log data sets manually using sample members from the BBSAMP data set.

Defining a Trace Log Data Set

You can define different trace logs as often as you need them, or you can let the BBI-SS PAS allocate them for you dynamically (see the JXT011 sample job description in the *MAINVIEW for DBCTL Analyzers, Monitors, and Traces Reference Manual*).

Archiving a Trace Log Data Set

A trace request can be defined to archive a log data set automatically when it is full, as described in the IMFTARC sample job description in the *MAINVIEW for DBCTL Analyzers, Monitors, and Traces Reference Manual*.

You can manually submit the IMFTARC job to archive a trace log data set that is no longer active.

Restoring an Archived Trace Log Data Set

Use BBSAMP member IMFTRLOD to restore an archived trace log data set.

Note: You also can add the linear data set to the online trace directory and view the contents online. Use the NEW command in the History Traces application, as described in the *MAINVIEW for DBCTL Analyzers, Monitors, and Traces Reference Manual*.

Creating a Trace Log Data Set from the IMS Log

You can use BBSAMP member IMFLOGTR to create a user-selected summary trace from the IMS log. See the *MAINVIEW for DBCTL Analyzers, Monitors, and Traces Reference Manual* for a description of how to use IMFLOGTR.

Note: You can add the trace log data set created by IMFLOGTR to the online trace directory and view the contents online. Use the NEW command in the History Traces application, as described in the *MAINVIEW for DBCTL Analyzers, Monitors, and Traces Reference Manual*.

Printing a Trace Log Data Set

You can print a trace log data set either from a batch job or from an online application.

From a Batch Job

You can use BBSAMP member WATBTRAC to print a trace log data set. For a full description of how to print a trace from a batch job, see the chapter about printing history traces in the *MAINVIEW for DBCTL Analyzers, Monitors, and Traces Reference Manual*.

From an Online Application

You can print a trace log data set from the online History Traces application if you are executing your TS from ISPF.

Before printing from the online application, you must first copy the skeleton JCL located in member WATBPRNT of the BBPROF library. Copy this member to an individual user data set (BBPROF) or to a site data set (SBBPROF). The BBPROF or SBBPROF data set must be defined in the CLIST (MAINVIEW CLIST) used to start the terminal session. For more information about BBPROF, see the *MAINVIEW Common Customization Guide*.

To print from the online application:

1. Select option 4 from the Primary Option Menu.
2. In the History Traces panel, enter the P line command next to the data set you want to print. (For a full description of how to use line commands in the History Traces application, see the *MAINVIEW for DBCTL Analyzers, Monitors, and Traces Reference Manual*.)
3. Enter the required information in the next panel that appears. (This panel gives you options that allow you to tailor the print job output to your needs.)
4. Press the End key.

The printed trace data has the same format and content as the online display.

Setting Up Workload History Traces

The BBIISP00 member of the BBPARM data set allows you to select a group of timer-driven monitor and trace requests to start automatically. These requests are defined in another member of the BBPARM data set. If you specify a default block request member BLKIMFT, for example, in BBIISP00 (`TARGET=imsid,BLK=BLKIMFT`), a starter set of monitors and the following summary trace could be requested:

```
REQ=MTRAC TRHIST  TYPE=SUMMARY  TITLE='TRACE HISTORY'  STORAGE=4000K
*          LOGTRAC=Y   TRNUMDS=3   TRSWTIME=24:00
```

The example above is a summary trace of the complete IMS workload. It should be run as a standard request to provide viewing of trace history. You can access it directly from the History Traces application. It adds very minimal overhead since it requires only a summarized trace.

The second line is set up as a comment to show you how you could define trace logging to a set of three data sets, automatically switching to a new data set at midnight. Depending on your IMS workload volume and operations procedures, you may need to modify some of these parameter values or specify others.

All options are defined in the *MAINVIEW for DBCTL Analyzers, Monitors, and Traces Reference Manual* in “Requesting Workload Trace Data Collection (MTRAC).” Setting defaults for all traces, such as the volumes to be used for allocation, is described in “Specifying Trace Defaults in IMFBEX00” on page 34 of this manual.

Although there are many options available, there are basically two ways to set up continuous trace logging. You must evaluate your system characteristics before choosing which is better.

- The first method uses automatic allocation of one or more new trace log data sets each time the trace request is started (at BBI-SS PAS startup), as shown in the preceding example. No DSN is specified so that the generated name is always unique (specifying TRPREFIX in IMFBEX00 defines the high-level node).

  This method can be used if OS/390 and the BBI-SS PAS are rarely brought down. The only consideration is that if the trace log data set allocation fails, perhaps because of lack of space, the trace request also fails.

- The second method is to set up a group of pre-allocated trace log data sets (any number of them) that are continually reused. An archive job can be defined to run automatically (log full, log switched, or trace complete) to save the data and mark that log for reuse. Each time the BBI-SS PAS starts, and this trace request is started, the next available log with the oldest data is chosen automatically for output.

  This method uses fewer online log data sets. However, if you require archiving, this method may require intervention after any unplanned outage of OS/390 or the BBI-SS PAS, since the archive job on the current trace log cannot run. If you do not require archiving, specify `TRREUSE=Y` to allow overwriting of a log without it being reset.

Setting Limits for Transaction Trace Views

The parameters shown in Table 2 can be used in BBPARM member IMFBEX00 to set limits for transaction trace (ITA*) views. The default values are underlined.

Table 2. IMFBEX00 Parameters for Trace Transaction Limits

Parameter	**Valid Values**	**Description**
TRMAXRD	5000 1 to 99999999	Maximum number of trace transaction records read before query terminates.
TRMAXWR	1000 1 to 99999999	Maximum number of trace transaction records written before query terminates.

Note: To activate changes you make to IMFBEX00 parameters, specify `RESET PARM IMFBEX00` or restart the BBI-SS PAS.

Chapter 8. Controlling Workload Thresholds

This chapter describes how to use parameters in BBPARM member IMFBEX00 to control workload thresholds for IMS and IMS Sysplex activity (DBC*) views.

To activate changes you make to IMFBEX00 parameters, specify `RESET PARM IMFBEX00` or restart the BBI-SS PAS.

The parameters shown in Table 3 can be used in BBPARM member IMFBEX00 to control workload thresholds for IMS and IMS Sysplex activity (DBC*) views. The default values are underlined.

Table 3. IMFBEX00 Parameters for IMS and IMS Sysplex Activity Workload Thresholds

Parameter	Valid Values	Description
EXCSINP	<u>0.2</u> 0 to 99.99	Input queue time in seconds. It specifies a threshold for excessive transaction input queue time.
EXCSELP	<u>0.5</u> 0 to 99.99	Elapsed time in seconds. It specifies a threshold for excessive transaction elapsed time.
EXCSCPU	<u>1</u> 0 to 100	Region CPU time percentage. It specifies a threshold for excessive region CPU time.
EXCSDLI	<u>50</u> 0 to 99999	Region DL/I calls count. It specifies a threshold for excessive DL/I calls.
EXCSSQL	<u>50</u> 0 to 99999	SQL call count. It specifies a threshold for excessive SQL calls.

Chapter 9. Using the Event Collector

Event Collector data is written to the IMS log as user record logs `X'FA'` (transaction records) and `X'F9'` (program records) for later processing by the offline components, Performance Reporter and Transaction Accountant. The data can be used by user-developed programs or other offline processing systems. Transaction data collected by the Event Collector is also passed to the online Workload Monitor and Workload Analyzer components for workload wait and trace services.

Note: The Resource Analyzer and Resource Monitor components and the MAINVIEW AutoOPERATOR component do not interact with the Event Collector.

Data collection parameters specified in BBPARM member IMFECP00 affect what the Event Collector does during an IMS session. These parameters control

- configuration, such as SYSID specification
- function, such as the type and amount of data to collect
- recovery, such as action on abend

Parameter options are provided to allow you to limit the data collected by the Event Collector and thereby limit the resources used. These options are described in this chapter.

Starting and Stopping the Event Collector

The Event Collector cannot be dynamically started or stopped. It initializes immediately after IMS initialization is completed and it stays active until IMS terminates.

Event Collector Parameters

The Event Collector data collection parameters are specified in BBPARM member IMFECP00. These parameters are described in the two tables that follow. The parameters specify the data to be collected, the functions to be performed during error recovery, and MVDBC diagnostics. Some of the available collection, recovery, and diagnostic parameters can be CPU-intensive and careful consideration should be given to their selection.
The parameters are read and processed by MVDBC at IMS initialization and remain in effect throughout the IMS session.

The tables in this section define each parameter, the data that is collected, and the impact on CPU usage. The parameters are specified in BBPARM member IMFECP00.

Table 4 lists the Event Collector data collection options alphabetically by parameter name.

Table 4. Event Collector Data Collection Parameters

Parameter	Value	Function, Data Collected, and CPU Usage
BHT	OFF	**Function:** Includes buffer handler time (BHT) data in DL/I CPU time data. **Note:** `BHTO=OFF` is forced if DBIO option is not BFALTERS.
		Data: Included in DL/I CPU time data. **Note:** DEDB and MSDB BHT for IMS Fast Path is always included in DL/I CPU time data. Performance Reporter and Transaction Accountant report `BHT CPU=0`. Performance Reporter, Transaction Accountant, and Workload Analyzer trace services report increased DL/I CPU time (`DL/I CPU = DL/I + BHT`), which produces increased chargeable CPU time in Transaction Accountant. Workload Analyzer trace services report increased DL/I CPU time (`DL/I CPU = DL/I + BHT`).
		CPU Usage: Most efficient option.
	ON	**Function:** Collects BHT data separately.
		Data: BHT data is available for Performance Reporter and Transaction Accountant.
		CPU Usage: Time that Event Collector adds to IMS usage can be increased 20 to 40 percent; that is, if total MVIMS overhead is 10 percent of IMS CPU, the option could cause the total overhead to be 12 to 14 percent. This increase depends on the percentage of DL/I database calls in the total IMS workload.
BILLOVHD – This parameter does not apply to MAINVIEW for DBCTL.		
BMP	YES	**Function:** Collects BMP and JBP transaction and program activity data.
		Data: Data available for workload trace.
		CPU Usage: Event Collector CPU usage depends on the number and activity of all BMPs and JBPs, but usage is higher than if NO or NOCPU is specified.
	NO	**Function:** Does not measure BMP and JBP activity.
		Data: No data available for workload trace.
		CPU Usage: Event Collector CPU usage depends on the number and activity of all BMPs and JBPs, but usage is lower than if YES or NOCPU is specified.
	NOCPU	**Function:** Collects BMP and JBP transaction and program activity data, but not CPU time usage.
		Data: No CPU data available for BMPs and JBPs. All other data, such as DL/I counts and database accesses, are still available for workload trace.
		CPU Usage: CPU time measurement, the largest overhead item, is not taken, which enables the collection of other statistics.

Table 4. Event Collector Data Collection Parameters (continued)

Parameter	Value	Function, Data Collected, and CPU Usage
CICS	YES	**Function:** Combines online and offline functions.
		Data: Data available for offline batch report products, Workload Analyzer wait and trace services, and Workload Monitor services.
		CPU Usage: It depends on the number and activity of CICS and ODBA transactions and programs.
	ONLINE	**Function:** Records data for CICS and ODBA threads and sends it to the BBI-SS PAS for processing by Workload Analyzer wait and trace services and Workload Monitor services, but does not write the data to the IMS log.
		Data: Data is available for workload monitors, wait, and trace.
		CPU Usage: It depends on the number and activity of CICS and ODBA transactions and programs.
	OFFLINE	**Function:** Records data for CICS and ODBA threads and writes it to the IMS log for batch report processing.
		Data: Data available for offline batch products, such as Performance Reporter and Transaction Accountant, but not for the Workload Analyzer wait or trace services, nor for the Workload Monitor services.
		CPU Usage: It depends on the number and activity of CICS and ODBA transactions and programs.
	NO	**Function:** Does not measure CICS and ODBA thread activity data.
		Data: No data available for Performance Reporter or Transaction Accountant, Workload Analyzer wait or trace services, or Workload Monitor services.
		CPU Usage: It depends on the number and activity of CICS and ODBA transactions and programs.
CPICDB2	TERM	**Function:** For CPI-C (explicit APPC) conversations, writes a single transaction record for each conversation. **Note:** The CPICDB2 options are in effect for a conversation until the application does an APSB call. When an APSB call is issued, the CPICDLI options are then in effect for the remainder of the application.
		Data: Workload Analyzer, Workload Monitor, Performance Reporter, and Transaction Accountant report one transaction for each CPI-C conversation.
		CPU Usage: No additional CPU.
	SYNC	**Function:** For CPI-C (explicit APPC) conversations, writes a transaction record for each SRRCMIT.
		Data: Workload Analyzer, Workload Monitor, Performance Reporter, and Transaction Accountant report one transaction for each CPI-C conversation.
		CPU Usage: Minimal increase occurs at sync point.

Table 4. Event Collector Data Collection Parameters (continued)

<table>
<tr><th>Parameter</th><th>Value</th><th>Function, Data Collected, and CPU Usage</th></tr>
<tr><td rowspan="6">CPICDLI</td><td rowspan="3"><u>APSB</u></td><td>Function: For CPI-C (explicit APPC) conversations, writes a transaction record each time a new PSB is allocated by an APSB call.

Note: The CPICDLI options override the CPICDB2 options if and when the application does an APSB call.</td></tr>
<tr><td>Data: Workload Analyzer, Workload Monitor, Performance Reporter, and Transaction Accountant report one transaction for each APSB.</td></tr>
<tr><td>CPU Usage: No additional CPU.</td></tr>
<tr><td rowspan="3">SYNC</td><td>Function: For CPI-C (explicit APPC) conversations, writes a transaction record for each SRRCMIT.</td></tr>
<tr><td>Data: Workload Analyzer, Workload Monitor, Performance Reporter, and Transaction Accountant report one transaction for each sync point.</td></tr>
<tr><td>CPU Usage: Minimal increase occurs at sync point.</td></tr>
</table>

Table 4. Event Collector Data Collection Parameters (continued)

Parameter	Value	Function, Data Collected, and CPU Usage
CPU	DEPPGM	**Function:** Collects CPU time data from dependent regions only. It is recorded as application time and includes DL/I and DB2 time. **Note:** Most chargeable CPU time is still collected.
		Data: CPU fields for CONTROL and DLISAS are zero. Application program CPU contains all dependent region chargeable CPU and message DL/I, buffer, and DB2 CPU contain zeros. This is reflected in all CPU data shown by Performance Reporter, Transaction Accountant, and workload trace.
		CPU Usage: Biggest overhead reduction, since it times the entire transaction as a single event instead of timing each DL/I and SQL call.
	DEPDB2	**Function:** Collects CPU time data from dependent regions only. It is recorded as application and DB2 time. The DL/I CPU time is included in the CPU time.
		Data: CPU fields for CONTROL and DLISAS are zero. Application program CPU contains all dependent region chargeable CPU except DB2 CPU. Message DL/I and buffer CPU contain zeros. This is reflected in all CPU data shown by Performance Reporter, Transaction Accountant, and workload trace.
		CPU Usage: The potential amount of overhead saved from this option is highly dependent on how many SQL calls the program issues.
	DEP	**Function:** Collects CPU time data for transaction processing from dependent regions only. It is recorded as application, DL/I, and DB2 time. **Note:** Most chargeable CPU time is still collected unless `LSO=Y` or BMPs/JBPs are run with nonparallel DL/I.
		Data: CPU fields for CONTROL and DLISAS are zero. Performance Reporter and Transaction Accountant scheduling and open/close CPU fields are zero.
		CPU Usage: Usage is less than with ALL, and most of the CPU time data is collected.
	ALL	**Function:** Collects all CPU time data.
		Data: All CPU time data is available. Reported DL/I CPU time is approximately 5 to 15 percent higher than with `CPU=DEP` option, depending on the amount of DL/I and DB2 activity and the LSO option.
		CPU Usage: An increase of 3 to 12 percent over `CPU=DEP`, depending upon the number of DL/I and DB2 message calls.
	NONE	**Function:** Does not collect CPU time data.
		Data: All CPU time fields in records contain zeroes.
		CPU Usage: Least usage but greatest data loss.

Table 4. Event Collector Data Collection Parameters (continued)

Parameter	Value	Function, Data Collected, and CPU Usage
CPUOVHD	YES	**Function:** Collects control region overhead and DLISAS region overhead in the MVIMS program log record (X'F9'), even if `CPU=ALL` is not specified. **Note:** Control region overhead and DLISAS region overhead are always collected when `CPU=ALL` is specified; therefore, `CPUOVHD=YES` has no affect when `CPU=ALL` is specified. `CPUOVHD=YES` has no affect on dependent region overhead.
		Data: The overhead CPU fields in the records include control region overhead and DLISAS region overhead values.
		CPU Usage: CPUOVHD does not affect CPU usage.
	REFCPU	**Function:** Collects overhead CPU values in the MVIMS program log record (X'F9') based on the CPU parameter specification.

		Data: REFCPU means REFer CPU. What is collected is determined by the use of the CPU and CPUOVHD parameters together:
		
		`CPUOVHD=REFCPU`
		`CPU=NONE`
		
		Does not collect CPU overhead data.
		
		`CPUOVHD=REFCPU`
		`CPU=ALL`
		
		Collects all CPU overhead data from control, DLISAS, and dependent regions.
		
		`CPUOVHD=REFCPU`
		`CPU=DEP`
		
		Collects only CPU overhead data from the dependent region. Does not collect DLISAS or control region CPU overhead data.
		
		Note: The CPU overhead values are set to zero when no CPU timing is done. That is, control region overhead CPU is set to zero when `CPU=NONE` or `CPU=DEP`. Message region overhead CPU is also set to zero when `CPU=NONE`.

		CPU Usage: CPUOVHD does not affect CPU usage.
Note: The CPUOVHD parameter applies only to CPU overhead; it does not affect chargeable CPU time.		
DBFP	NO	**Function:** Collects counts of NONKEY WRITES and NO I/O ALTERS and reports them in the database trailer.
		Data: Counts of NONKEY WRITES and NO I/O ALTERS are collected and are available for performance analysis and billing.
	YES	**Function:** Bypasses collection of NONKEY WRITES and NO I/O ALTERS.
		Data: NONKEY WRITES and NO I/O ALTERS counts are not reported in the database trailer. This value is recommended if the statistics are not required for performance analysis or billing.
		CPU Usage: Reduces CPU usage for Fast Path transactions.

Table 4. Event Collector Data Collection Parameters (continued)

Parameter	Value	Function, Data Collected, and CPU Usage
DBFPLVL	0	**Function:** No DEDB activity is collected.
		Data: No DEDB DL/I call or I/O activity data is available for display in the IPSM database activity views, and no DEDB I/O read data is available for display in detail traces.
		CPU Usage: None.
	1	**Function:** DEDB read I/O data is collected for detail traces.
		Data: DEDB read I/O data is available for display in detail traces.
		CPU Usage: Uses less CPU than options 2 and 3.
	2	**Function:** DEDB read I/O data is collected for detail traces. DEDB DL/I call data and read I/O data are collected for database I/O activity analysis.
		Data: DEDB read I/O data is available for display in detail traces. DEDB DL/I call data and read I/O data are available for display in the IPSM database activity views.
		CPU Usage: Uses less CPU than option 3.
	3	**Function:** DEDB read I/O data is collected for detail traces. DEDB DL/I call data and read I/O and write I/O data are collected for database I/O activity analysis.
		Data: DEDB read I/O data is available for display in detail traces. DEDB DL/I call data and read I/O and write I/O data are available for display in the IPSM database activity views.
		CPU Usage: Uses more CPU than the other options.

Table 4. Event Collector Data Collection Parameters (continued)

Parameter	Value	Function, Data Collected, and CPU Usage
DBIO	IOWAITS	**Function:** Collects reads for each database; collects writes at the transaction level. Forces `BHTO=OFF`. NO-I/Os are not collected.
		Data: All database I/O collected at I/O IWAIT. Workload trace services I/O counts per transaction are very close to values with BFALTERS option. Most WRITEs are collected for the transaction in a special database trailer, ALLDBS, instead of per database.
		CPU Usage: Substantially less than with BFALTERS with minimal loss of data.
	BFALTERS	**Function:** Collects all I/O data for each database.
		Data: Database I/O and NO-I/O collected in the buffer handler interface (during the DL/I call).
		CPU Usage: Usage can be increased 30 to 40 percent over IOWAITS because of the high ratio of buffer handler activity to DL/I calls. Increase depends on the percentage of DL/I database calls in the total IMS workload.
	NONE	**Function:** Forces `BHTO=OFF`. No I/O data is collected.
		Data: DATABASE I/O and NO-I/O fields are zero. Workload trace services data I/O and NO-I/O fields contain zeroes.
		CPU Usage: None.
Note: `DBIO=IOWAITS` and `DBIO=NONE` do not apply to Fast Path.		
DBTNAME	DB	**Function:** Collects both DL/I call counts and database I/O call counts at the database level.
		Data: DL/I call counts and I/O call counts are collected at the DBPCB name (database) level. A database trailer (DBT) is created for each DBPCB name that contains DL/I and I/O calls.
		CPU Usage: Minimal usage.
	DD	**Function:** Collects DL/I call counts at the database level. Collects database I/O counts at the data set level.
		Data: DL/I call counts are collected at the DBPCB name level. I/O call counts are collected at the ddname (data set name) level. A DBT is created for each DBPCB name containing DL/I calls. A DBT is created for each ddname containing I/O calls. This option uses more database trailers than the DB option.
		CPU Usage: Least efficient option.
Note: `DBTNAME=DD` does not apply to Fast Path.		
DBTS	20	**Function:** Sets the maximum number of database trailers allowed per non-BMP/JBP region. Valid values are 2 to 500. The recommended value is 20. The default value is 10.
		Data: If a transaction accesses more than this number of databases, resource data is collected in an overflow trailer named OTHERS.
		CPU Usage: None. Affects ECSA requirement for each DBCTL thread.

Table 4. Event Collector Data Collection Parameters (continued)

Parameter	Value	Function, Data Collected, and CPU Usage
DBTS4BMP	30	**Function:** Sets the maximum number of database trailers allowed per BMP or JBP region. Valid values are 2 to 500 (30 is recommended). If a value is not specified, the value specified for DBTS is the default.
		Data: If a transaction accesses more than this number of databases, resource data is collected in an overflow trailer named OTHERS.
		CPU Usage: None. Affects ECSA requirement for each DBCTL thread.
SYSID	1	**Function:** Identifies the MVDBC system for Performance Reporter and Transaction Accountant. Valid values are 1 to 9 or A to Z. Used to identify and select data from other IMS systems.
		Data: None.
		CPU Usage: None.
TRNSYNC	NO	**Function:** Does not write a transaction record at BMP or JBP checkpoint.
		Data: A record is written only per successful MESSAGE-GET-UNIQUE or at program end.
		CPU Usage: None.
	YES	**Function:** Writes a transaction record at BMP or JBP checkpoint.
		Data: A record is written.
		CPU Usage: Minimal.

Table 5 lists Event Collector recovery parameters.

Table 5. Event Collector Recovery Parameters

Parameter	Value	Function, Data Collected, and CPU Usage
ABCOUNT	02	**Function:** Specifies the number of abend retries allowed. Valid values are 01 to 99.
BACKOUT	YES	**Function:** For severe errors, back out Event Collector; do not abend IMS.
	NO	**Function:** For severe errors abend IMS; do not back out Event Collector.
DEPREC	YES	**Function:** Performs extended recovery.
		Data: Performs recovery for additional abend conditions.
		CPU Usage: Usage increased 10 to 15 percent, depending on the other selected Event Collector parameter options. This option should be set to NO after MVDBC is thoroughly tested with YES selected.
	NO	**Function:** Performs basic recovery.
		Data: Some potential abend conditions cannot be recovered.
		CPU Effect: Most efficient. Usage depends on the other selected Event Collector parameter options. This option should be used after MVDBC is thoroughly tested with YES selected.
DUMPS	YES	**Function:** Takes SVC dumps.
	NO	**Function:** Does not take SVC dumps; produces a LOGREC only.
RGNIOPT	ABEND	**Function:** Abends the IMS dependent region if MVDBC initialization fails because of CSA shortage.
		Data: Full recording ensured.
		CPU Usage: None.
	CONTINUE	**Function:** Continues the IMS dependent region if MVDBC initialization fails.
		Data: No recording is done for that region.
		CPU Usage: None.

Evaluating Data Collection Options

Each data collection option has an effect on the function provided by MVDBC and on Event Collector CPU usage. Event Collector CPU usage is extremely dependent on the workload and configuration characteristics of the IMS system being monitored. This dependency, coupled with the different ways each site uses MVDBC data, make it impossible to summarize all these variables into standard CPU usage estimates or option setting recommendations.

This section provides a short description of the parameters, with an overview of their effect on the data collected and on CPU usage. These descriptions can help you evaluate the options and choose those most suited to your environment.

Note: All CPU usage values are expressed in relation to the overhead that the Event Collector adds to total IMS CPU. For example, if the Event Collector usage is defined as 10 percent and total IMS CPU has a theoretical value of 200, the total IMS + MVDBC usage is 220, as shown in Example #1, below.

Example #1:

```
IMS = 200
Event Collector = 10% of IMS = 200 * .10 = 20
Total = IMS + Event Collector = 200 + 20 = 220
```

An indicated percentage increase in Event Collector CPU usage (for example, 30 to 40 percent for `DBIO=BFALTERS`) is relative to the Event Collector usage value (20 in the previous example). Thus, for the previous example, the `DBIO=BFALTERS` would increase the total CPU usage by a value of 6 to 8 (30 to 40 percent of 20), as shown in Example #2, below.

Example #2:

```
IMS = 200
Event Collector = 10% of IMS = 200 * .10 = 20
Total = IMS + Event Collector + BFALTERS = 200 + 20 + 8 = 228
```

DBCTL Threads

`CICS=YES|ONLINE|OFFLINE|NO`

The CICS parameter controls whether records are collected for DBCTL thread data (which includes both CICS and ODBA threads). The CPU usage is the same for all options. CPU usage depends on the number and activity of CICS transaction programs.

BMP and JBP Data

`BMP=YES|NO|NOCPU`

The BMP parameter controls whether activity data for BMP and JBP transactions and programs is collected. If BMP or JBP processing is causing bottlenecks in the IMS online system, you may want to avoid the extra overhead that MVDBC monitoring adds. However, this option is viable only if the MVDBC BMP or JBP data is not required for IMS performance analysis. In general, most sites will want to collect BMP data.

The effect of this parameter on MVDBC CPU usage depends on the number and activity of all BMPs or JBPs.

Buffer Handler Timing

BHTO=OFF|ON

The BHTO parameter controls whether IMS buffer handler activity is included with DL/I CPU or timed separately. The default is to include it with DL/I (BHTO=OFF).

The high ratio of buffer handler calls to application program DL/I calls in IMS makes separate collection of buffer handler CPU very CPU-intensive for MVDBC. The ratio can be as high as 20 to 1, so collecting separate CPU time data for each buffer handler request can become too expensive when compared with the worth of the data. Depending on the number of database calls and the amount of buffer handler activity, BHTO=ON can increase MVDBC CPU usage by 20 to 40 percent.

The BHTO=ON option is provided for product compatibility, but it is not a recommended option.

CPU Data Collection Options

CPU=DEPPGM|DEPDB2|DEP|ALL|NONE

The CPU parameter controls the level of CPU data collected by the Event Collector in all IMS regions, in just the dependent regions, or in none.

CPU=DEPPGM causes the Event Collector to time only the dependent region activities. It times the entire transaction as a single event and does not time individual DL/I or DB2 calls.
The single resulting CPU time (representing all the chargeable time for the transaction) is attributed to application program CPU time. All other chargeable timings are zero. Overhead CPU time, however, is still kept separately.

CPU=DEPPGM offers the biggest overhead reduction, since it times the entire transaction as a single event instead of timing each DL/I and SQL call. However, the amount of overhead saved depends to a large extent on the current transaction processing profiles. For example, a BMP program issuing 10,000 DL/I calls saves more than an MPP program issuing only 10 DL/I calls. However, even when savings from each transaction are small, they add up quickly.

CPU=DEPDB2 functions similarly to DEPPGM except that the Event Collector separates the dependent region DB2/SQL time from the application program CPU time.

CPU=DEPDB2 causes the Event Collector to time the DB2 events (SQL calls). As a result, the potential amount of overhead saved from this option is highly dependent on how many SQL calls the transaction/program issues. For example, if an MPP program issues only two DL/I calls and 100 SQL calls, the amount saved is minimal.

CPU=DEP causes the Event Collector to attribute charageable CPU application program or DL/I processing CPU time to a specific transaction and user. When CPU=DEP is used, Event Collector CPU usage increases 25 to 35 percent, depending on the amount of DL/I activity, over CPU=NONE.

CPU=ALL adds collection of DL/I processing CPU in the control region and measurement of various overhead categories such as program scheduling activity. It can increase MVDBC CPU usage by 3 to 12 percent over the CPU=DEP option.

CPU=ALL is the best choice if the various overhead CPU categories are needed for performance analysis or if any of the following conditions are true of the monitored IMS:

- The IMS parameter LSO equals Y.
- BMPs and JBPs are run in nonparallel DL/I mode.
- The percentage of message queue DL/I calls compared to database calls is high. (On average, message queue calls are 5 to 15 percent of the total DL/I calls.)

All of these factors increase the amount of IMS CPU incurred in either the control or DLISAS regions.

Database I/O Options

DBIO=IOWAITS|BFALTERS|NONE

DBIO controls the level of database I/O data to be collected by the Event Collector. The DBIO=BFALTERS option collects all database activity indicators at the database level for each transaction.

With the BFALTERS option, reporting can be made by transaction and user, and by database for extended performance analysis. NO-I/O counts (the number of reads without I/Os), which show buffer handler activity, also can be collected when BFALTERS is selected. BFALTERS uses an IMS buffer handler interface, which is expensive because of the high ratio of requests to the buffer handler compared with DL/I calls and actual I/O.

DBIO=IOWAITS activates a more efficient method of data collection. DL/I calls are collected by database. I/Os are measured at actual occurrence (using the DC Monitor IWAIT interface) instead of in the buffer handler. With the IWAIT interface, reads and writes that occur during call processing are collected by database, but writes that occur at sync point (the majority) can be associated only with the transaction and user, not with the specific database. Most writes are collected at the transaction level and reported under a special database entry ALLDBS. NO-I/O counts are not collected.

IOWAITS provides the same level of data as BFALTERS for accounting and for the transaction, program, and totals levels of I/O analysis. For performance analysis at the database level, DL/I calls, reads, and some writes are still available. The other writes are reported per program.

IOWAITS is the default and recommended option because Event Collector CPU usage is significantly less than with BFALTERS, which can increase MVDBC CPU usage by 30 to 40 percent over the IOWAITS option, depending on the amount of database activity. Using the IOWAITS option increases the MVDBC CPU usage by 5 to 10 percent over DBIO=NONE, depending on the number of database I/Os.

DBIO=NONE specifies that reads, writes, and NO-I/O counts are not collected. DL/I calls are still available by database.

Note: The DBIO parameter does not affect Fast Path databases.

Extended Recovery

`DEPREC=YES|NO`

DEPREC controls whether recovery from additional abend conditions in dependent regions is enabled and performed as necessary.

MVDBC CPU usage may be increased 10 to 30 percent over the `DEPREC=NO` option, depending on the options chosen for other parameters (because the more work the Event Collector does, the more overhead is added by this option).

The default should remain set until MVDBC is thoroughly tested and stable in each environment. If CPU utilization is still of concern once the other options are chosen, this parameter could then be set to NO for additional savings.

Parameter Option Sets

This section describes three sets of parameter definitions.

Standard Option Set

The standard option set is distributed with the product. These options are defined in BBPARM member IMFECP00.

```
BMP=YES
CPU=DEPPGM
DBFPLVL=3
DBIO=IOWAITS
BHTO=OFF
DEPREC=YES
CICS=YES
```

These standard settings define a level of data collection and CPU usage that is acceptable for most configurations and users. Most I/O and CPU data is available and MVIMS CPU usage is noticeably less than with the full option set.

Note: Additional savings are possible without loss of data if you specify `DEPREC=NO`. Depending on some of the IMS characteristics described previously, you may want to use `CPU=ALL`, even though it increases MVDBC CPU usage.

The standard options, either as distributed or with the variations mentioned, generally result in MVDBC CPU usage of 6 to 16 percent of total IMS CPU.

Full Option Set

The full option set defines the maximum level of data collection by the Event Collector. These settings result in the greatest amount of MVDBC CPU usage.

```
BMP=YES
CPU=ALL
DBFPLVL=3
DBIO=BFALTERS
BHTO=OFF
DEPREC=YES
CICS=YES
```

More detailed database I/O analysis is possible with the full option set than with the standard set because the full option set collects writes per database and NO I/O counts.

For the CPU parameter, you can substitute the DEPPGM option for the ALL option to conserve CPU usage. Selecting the DEPPGM option is likely to be a one-time decision, although you might decide to use `CPU=ALL` occasionally for performance analysis.

For the DBIO parameter, the level of data collected with the BFALTERS option is rarely used on a daily basis. You can reduce collection overhead by using the IOWAITS option instead. You can then specify the BFALTERS option on only those occasions where more data is needed for database analysis.

`BHTO=ON` is not recommended because of its cost in relation to the value of the data collected (buffer handler CPU time).

You can decrease CPU usage by specifying `DEPREC=NO` without losing any data.

The full option set generally results in MVDBC CPU usage of 15 to 25 percent of total IMS CPU, though some individual BMPs or JBPs with a large amount of DL/I activity can be higher.

Minimum Option Set

The following option settings define the minimum level of data collection by the Event Collector. These settings result in the least amount of MVDBC CPU usage.

```
BMP=NO (optional)
CPU=NONE
DBFP=YES
DBFPLVL=0
DBIO=NONE
BHTO=OFF
DEPREC=NO
CICS=NO
```

The minimum option settings allow enough data collection for true transaction-level accounting and performance analysis, while keeping Event Collector CPU usage at a minimum. Records containing all the identifiers (transaction, program, region, and so on) available with MVDBC are still created for each transaction and program. Data collected includes

- all elapsed timings and storage usage figures
- DL/I call counts per call type
- DL/I call counts per database

The minimum option set generally results in MVDBC CPU usage of 3 to 10 percent of total IMS CPU.

Affects of Option Sets on Workload Trace Data and DBCTL Threads

The following section lists the online services that report one or more of the Event Collector data elements.

The basis for comparison is the level of data collection when the full option set, shown below, is used.

```
BMP=YES
CPU=ALL
DBFPLVL=3
DBIO=BFALTERS
BHTO=OFF
DEPREC=YES
CICS=YES
```

The following list indicates the effect of specifying values different from those shown above:

- BMP and JBP trace

  If BMP=NO, BMP and JBP transactions and programs cannot be traced.

- LTRAC

  If CPU=NONE or CPU=DEPPGM, the DB2TIME parameter is not applicable for the LTRAC service.

  If CPU=NONE, CPU=DEPPGM, or CPU=DEPDB2, the DLITIME parameter is not applicable for the LTRAC service.

- STRAC

  If CPU=NONE, all CPU data is zero.

- DTRAC

  If DBFPLVL=0, no DEDB I/O read information is available for display in detail traces.

- #I/O, AVG #I/O

  If DBIO=NONE, the values in these fields are zero.

- DTRAC DL/I CPU times

  If CPU=DEPPGM, CPU=DEPDB2, or CPU=NONE, the values in these fields are zero.

- DTRAC DB2 CPU times

  If CPU=DEPPGM or CPU=NONE, the values in these fields are zero.

- DBCTL thread tracing

  If CICS=NO or CICS=OFFLINE, no trace data is collected for DBCTL threads.

- DBCTL thread monitoring

  If CICS=NO or CICS=OFFLINE, no data is collected for the Workload Monitor services.

Changing Event Collector Parameters

If you change an Event Collector parameter in BBPARM member IMFECP00, the change does not take effect until IMS restarts.

Chapter 10. Defining Target DBCTL Systems

MAINVIEW for DBCTL operates in the MAINVIEW environment, which allows a terminal session (TS) to communicate with multiple targets associated with a BBI-SS product address space (PAS). As described in the *MAINVIEW Common Customization Guide*, BBPARM member BBIJNT00 is used to define all eligible target systems and associate them with the subsystem IDs of a BBI-SS PAS.

To define a target DBCTL system, use the jobname of the DBCTL region with the TARGET parameter in BBIJNT00. If you specify a DBCTL target, you also must specify `IMSTYPE=DBCTL` in that TARGET statement as follows:

```
TARGET=jobname,TYPE=IMS,IMSTYPE=DBCTL,RELEASE=0n10
```

Chapter 11. Customizing Event Collector User Exit Routines

This chapter describes the sample Event Collector user exit routines provided by BMC Software. The Event Collector user exit routines are skeleton programs in BBSAMP that you can customize to build more extensive routines to meet you specific needs.

Transaction Record User Exit Routine (IMRUTRN)

This routine receives control from the Event Collector just before a transaction record (`X'FA'`) is written. Control is always received in the IMS control region. The captured record can then be evaluated and changed if necessary before control is returned to the Event Collector. When control is returned to the Event Collector, Register 15 (R15) must be zero or the record is not logged. The routine must be logically reentrant.

Documentation about how to activate this routine is provided in BBSAMP member IMRUTRN3.

See “IMRUTRN and IMRUPGM Cross-Memory Mode Considerations” on page 65 for special considerations about using this exit.

Program Record User Exit Routine (IMRUPGM)

This routine receives control from the Event Collector just before a program record (`X'F9'`) is written. The captured record can then be evaluated and changed if necessary before control is returned to the Event Collector. When control is returned to the Event Collector, R15 must be zero or the record is not logged. The routine must be logically reentrant.

Documentation about how to activate this routine is provided in BBSAMP member IMRUPGM3.

See “IMRUTRN and IMRUPGM Cross-Memory Mode Considerations” on page 65 for special considerations about using this exit.

IMRUTRN and IMRUPGM Cross-Memory Mode Considerations

IMRUTRN and IMRUPGM user exits can be invoked in cross-memory mode. However, when you invoke these exits from cross-memory mode, SVCs cannot be issued from the exits. Use OS/390 services that can be issued in cross-memory mode with an EUT FRR in effect. For example, instead of GETMAIN, use the STORAGE macro or a branch entry to GETMAIN.

These exits gain control with primary addressability set to the control region. The Event Collector sets an EUT FRR to provide recovery while these exits are in control, regardless of the DEPREC parameter value in BBPARM member IMFECP00. If the routine abends, a warning message, a LOGREC, and (optionally) a dump are produced. The warning message is an action message that does not scroll off the operator’s console.

Abends in these exits are charged against the Event Collector. If the number of abends exceeds the ABCOUNT value specified in IMFECP00, MVDBC takes one of the following actions.

- If BACKOUT=NO, MVDBC abends the IMS control region.
- If BACKOUT=YES, MVDBC disables the Event Collector.

The BACKOUT parameter, specified in IMFECP00, has a default of YES (see Table 5 on page 54).

DL/I-CALL-END User Exit Routine (IMRUDLI)

This routine can be used to extract job accounting information from the dependent regions or to extract user activity information from the transaction input message. This information can be especially valuable for BMP and JBP accounting and for application generators such as ADF and TELON.

The routine receives control from the Event Collector in the dependent region just after the first DC and the first DB DL/I call completed by IMS for a transaction or program. Therefore, for most transactions the exit is given control twice. This process accommodates programs that may not issue calls of both types (for example, a non-message-driven BMP that issues only DB calls or an MPP that issues only DC calls to perform message switching).

Caution

The transaction record is not in its final format at the time this user exit receives control. Modifications to any fields other than those reserved for the user may be overwritten by later processing.

This routine is loaded into CSA at IMS/MVDBC initialization time and must be logically reentrant.

Documentation about how to activate this routine is in BBSAMP member IMRUDLI3.

Chapter 12. Service Utility Commands

This chapter documents service utility commands, which are used by system programmers to control and maintain service availability.

Locking a Service (LOCK)

The LOCK command can be used to remove a service from general availability. It prevents the use of a specific service until a corresponding ULOCK request is issued. To issue the LOCK command, either

- type LOCK in the SERV field of an application, or
- select a service from a list application with an L line command.

Unlocking a Service (ULOCK)

A service may become locked if

- the service abended
- the service was the target of a LOCK command
- the service routine could not be loaded

After the condition is corrected, the ULOCK command can be issued to unlock the service. To issue the ULOCK command, either

- type ULOCK in the SERV field of an application, or
- select a service from a list application with a U line command

Part 3. Implementing Product Security

This part describes how users can be authorized to access to MAINVIEW for DBCTL services.

Chapter 13. Security for Analyzer and Monitor Services

Security for analyzer and monitor services is defined both in service tables that you can modify (see the ACCESS parameter in “Service Table Definition” on page 84) and by the user. Users are authorized to access analyzer and monitor services through the PMACC parameter in the user’s BBPARM authorization (USERID) member (see *Implementing Security for MAINVIEW Products*). The security level for each analyzer or monitor service is shown in the analyzer and monitor list applications.

Service Selection Lists by User Group

Service lists selected from the following Service Selection Menu options can be restricted to list only the services for which the user has authority.

- analyzer display services
- data collection monitors
- active timer requests
- general commands

The SERVLIST parameter in BBPARM member BBIISP00 determines whether this feature is activated. The default is `SERVLIST=ALL`, which means you see all services on the selection list displays. `SERVLIST=RESTRICT` specifies that you see only those services for which you are authorized. The default service security code is A.

BBPARM member IMFSTD00 is an example of how to set up the service security codes by IMS IMFSTD00 functional area, such as IRLM functions or IMS workload. Each service is assigned a security code according to its area. USERID members can then be created either for groups (such as MTOs or system programmers) or for individuals to access only specific services by listing one or more security codes. If the corresponding security code is not defined in the user’s authorization member, the user does not see those services on the service selection displays.

Command Authorization

Users must be authorized to issue commands or use applications against a BBI-SS PAS target system. Command authorization is defined in user authorization members of the BBPARM data set or by using BBPARM member $GENERIC. The authorization parameter is

PMACC — Service class authorization, global authorization, or request authorization (modify or purge).

The ACCESS parameter described on page 86 must match the user’s authorized access code specified for PMACC.

A description of how to use the PMACC parameter is in *Implementing Security for MAINVIEW Products*. The comment field of the USERID member of the BBPARM data set also describes this parameter and its options.

Trace Authorization

The USERID member of the BBPARM member data set can be used to provide user authorization for traces. Users can have their own members with their user IDs as the member names or the default $USERID member can be used.

The trace authorization parameters in BBPARM member USERID are TRACE, TRALLOC, and PMACC. TRACE determines whether a user can start a summary or detail trace. TRALLOC authorizes a user for dynamic allocation of trace log data sets by the BBI-SS PAS. PMACC provides trace service access and change authority.

Part 4. Appendixes

This part provides supplemental information for MAINVIEW for DBCTL use.

Appendix A. IMS Dump Analysis

This appendix describes how to analyze an IMS dump with MAINVIEW for DBCTL or MAINVIEW AutoOPERATOR for IMS installed.

MAINVIEW AutoOPERATOR for IMS Routines in IMS

During initialization, the MAINVIEW AutoOPERATOR for IMS AO code creates two subtasks under the IMS control task. Each of these is protected by ESTAE routines and uses different control blocks than those used by IMS. These subtasks generally can be ignored during IMS dump analysis since they do not affect the IMS flow. These subtasks are terminated correctly at IMS termination.

Event Collector

The following information should be noted about IMS dumps.

- MAINVIEW for DBCTL register save areas are in MAINVIEW for DBCTL data areas, not in the IMS prechained save areas. The IMS chains remain unchanged.
- Sometimes the R14 return register in an IMS save area does not point back into the calling IMS module. The IMS R14 value can be found 4 bytes in front of the address pointed to by R14.
- A MAINVIEW for DBCTL module at entry saves the registers of an IMS module in the next IMS prechained save area pointed to by R13.
- MAINVIEW for DBCTL module registers are always saved in MAINVIEW for DBCTL save areas. These save areas are dynamically assigned as required. Normally, one of several preallocated save areas per region is used.

  Each active IMS region has a MAINVIEW for DBCTL data area acquired for it at region initialization. This data area is in ECSA and is named IMERD*nnn*, where *nnn* is the PST number. The preallocated save areas are in this block.

  If more save areas are required, a dynamic storage pool is used. This pool also is in ECSA and is named IMFSP000.

  Any area in actual use as a MAINVIEW for DBCTL save area, whether in IMERD*nnn* or IMFSP000, is identified with SAR or ISA.
- In most cases, only a save area backward pointer exists, pointing from the MAINVIEW for DBCTL area to the previous IMS save area. While a MAINVIEW for DBCTL module is in control, the current R12 is its base register and the current R13 points to its SAR.
- In some cases, a MAINVIEW for DBCTL module transfers control to an IMS module, but needs to regain control after it completes processing. In this case also, the IMS save area chains remain unchanged.

  The only sign of the presence of a MAINVIEW for DBCTL module is an R14 value in an IMS save area which does not point back to the calling IMS module. If you need to verify the IMS path, the IMS R14 value is saved in the MAINVIEW for DBCTL SAR, 4 bytes in front of the address pointed to by the R14 in the IMS save area.

Note: When MAINVIEW for DBCTL interfaces between two IMS modules, MAINVIEW for DBCTL is transparent to the IMS modules. All registers are preserved.

- IMECSRVx and IMFCSRVx appear in dumps as active ITASKS. This situation is normal as long as the current save area is for DFSIWAIT.

Appendix B. How Product Libraries Should Be Used

Several distributed libraries are included with your MAINVIEW products, including a parameter library (BBPARM), a sample 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 automatically, with AutoCustomization.

Warning

The distributed libraries should never 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 6 lists the distributed name, the corresponding customized library created by AutoCustomization, and leaves space for you to note any other corresponding library that may have been created for your site.

Table 6. Product Libraries

Distributed Library Name	**Library Created by AutoCustomization**	**Other Site-Customized Copy**
BBPARM	UBBPARM	
BBSAMP	UBBSAMP	
BBPROF	SBBPROF	

For detailed information about all the product libraries, see the *MAINVIEW Common Customization Guide* or the *MAINVIEW Administration Guide*.

Appendix C. BBSAMP Data Set Members

To help you understand and use your BMC Software product and make it easier to use, the BBSAMP data set contains members that you can edit for your site's use. These members contain macros, sample JCL, sample user exit routines, and sample statements for a variety of functions.

Table 7 describes the BBSAMP members used for MAINVIEW for IMS and MAINVIEW for DBCTL.

Table 7. BBSAMP Data Set JCL Members

BBSAMP Member Name	**Description**
ARCHCTL1	Sample SYSIN for the log archive utility (see ARCHJCL) that selects only MAINVIEW for IMS log records.
ARCHCTL2	Sample SYSIN for the log archive utility (see ARCHJCL) that selects all MAINVIEW for IMS log records and some IMS log records.
ARCHJCL	Sample JCL for the IMS log archive utility, which produces an MAINVIEW for IMS user file.
BLKDBTW	Sample starter set of monitors for MAINVIEW for DBCTL.
BLKIMFW	Sample starter set of monitors for MAINVIEW for IMS.
CIMLAR01	Sample COBOL layout for IRUF terminal (LTERM) accounting record (LAR).
CIMPAR01	Sample COBOL layout for IRUF program accounting record (PAR).
CIMTAR01	Sample COBOL layout for IRUF transaction accounting record (TAR) for MVIMS version 3.2 and earlier.
CIMTAR02	Sample COBOL layout for IRUF transaction accounting record (TAR) for MVIMS version 3.3.00 and later.
CIRUFR01	Macro to map all IRUF records in Assembler for MVIMS version 3.2 and earlier.
CIRUFR02	Macro to map all IRUF records in Assembler for MVIMS version 3.3.00 and later
FPORUN	Sample JCL to read the IMS log and create an IRUF with response option and produce several reports.
GTFIMF	Diagnostics. Sample JCL (see GTFUSE).
GTFIMFP	Diagnostics. Sample JCL (see GTFUSE).
GTFIMF00	Diagnostics. Sample parameters (see GTFUSE).
GTFUSE	Diagnostics. How to use GTF trace facility for MAINVIEW for IMS Event Collector.
ICOPY*n*	Sample JCL to copy BMC Software modules from the BBLINK library to a site-authorized library, where *n* applies to an IMS release.

Table 7. BBSAMP Data Set JCL Members (continued)

BBSAMP Member Name	Description
IMEDBT	Macro to map the database trailer (DBT) for the MAINVIEW for IMS transaction log record in Assembler.
IMEPGM	Macro to map the MAINVIEW for IMS program log record in Assembler.
IMETMEQU	Macro used within IMETRN.
IMETRN	Macro to map the MAINVIEW for IMS transaction log record in Assembler.
IMFACTIV	IMS PR. Sample JCL to create a general activity analysis for all LTERMs and databases.
IMFARB	Macro referenced when assembling some user-written services.
IMFASYDS	Macro referenced when assembling some user-written services.
IMFCLNDR	Performance Reporter. Sample JCL to produce sample calendar reports. One sample of each type of calendar report is produced.
IMFCOSTR	Sample JCL to sort an IRUF into customer ID sequence and summarize it (program TASCOSTR).
IMFFNSUM	Transaction Accountant. Sample JCL to create a financial summary analysis from a costed IRUF.
IMFFPRPT	Sample JCL to create Fast Path transaction processing statistics.
IMFLEDIT	Sample JCL to create an IRUF by editing the IMS system log.
IMFMVSXA	Macro referenced when assembling some user-written services. This macro can be referenced in the BBXS macro library BBMAC, or the BBSAMP member IMFMVSXA can be renamed to BBXMVSXA.
IMFPLOT	Sample JCL to produce X-Y plots of selected variables.
IMFPROG	Performance Reporter. Sample JCL to produce program processing statistics.
IMFREGUT	Performance Reporter. Sample JCL to produce message region utilization analysis.
IMFRESP	Performance Reporter. Sample JCL to produce a response-time distribution report.
IMFRPTS	Sample JCL to read the IMS log, create an IRUF with response option, and produce several reports.
IMFSELEC	Sample JCL to select a subset of IRUF records with which to generate reports.
IMFSETAM	Macro referenced when assembling some user-written services. This macro can be referenced in the BBXS macro library BBMAC, or the BBSAMP member IMFSETAM can be renamed to BBXSETAM.
IMFTARC	Sample job to archive a trace log data set.

Table 7. BBSAMP Data Set JCL Members (continued)

BBSAMP Member Name	Description
IMFTRAN	Performance Reporter. Sample JCL to produce transaction processing statistics.
IMFTRLOD	Sample job to restore an archived trace log data set.
IMFTRND2	Performance Reporter. Sample JCL to produce DB2 transaction processing statistics.
IMFTRNFP	Performance Reporter. Sample JCL to produce Fast Path transaction processing statistics.
IMFVT	Macro referenced when assembling some user-written services.
IMRUDLI3	Sample Event Collector user exit routine for DL/I user exit routine CALL/END.
IMRUPGM3	Sample Event Collector user exit routine to access program records for evaluation.
IMRUTRN3	Sample Event Collector user exit routine to access transaction records for evaluation.
JASEXIT	Program to extract IMS charges from a summarized IRUF and create a charge record that can be used as input to the CONTROL/SMF charge-out system.
JRNLMSG	Macro referenced when assembling some user-written services.
JXT001	Sample job to set up and maintain a trace log directory.
JXT003	Sample job to verify existence of trace log data set in the system catalog.
JXT011	Sample job to define a trace log data set using IDCAMS.
LNKCEXIT	Linkage editor statements to replace default MAINVIEW for IMS Log Edit user exit routine that defines the customer ID with a user-written routine.
LNKLEXIT	Linkage editor statements to replace default MAINVIEW for IMS Log Edit user exit routine that accesses the log file with a user-written routine.
LINKSAP	Sample job to link SAPEXIT into SAP program library.
LOGCPU13	Sample JCL to produce a summary of the CPU times collected by the MAINVIEW for IMS Event Collector from the log records.
LOGREC	Sample JCL to print the software LOGREC records in SYS1.LOGREC.LOGREC
PRSCEXIT	Sample Assembler user exit routine to define the customer ID field in the IRUF.
PRSCEXITC	Sample COBOL user exit routine to define the customer ID field in the IRUF.

Table 7. BBSAMP Data Set JCL Members (continued)

BBSAMP Member Name	Description
PRSLEXIT	Sample Assembler user exit routine to access a log record read from the IMS system log.
PRSLEXTC	Sample COBOL user exit routine to access a log record read from the IMS system log.
PRSLEXTA	Sample user exit routine to access a log record read from the IMS system log and write the record to an external file.
PRSPRINT	Sample JCL to select and print IRUF records.
RARGEN	Resource Analyzer. Macro to set global values from defined parameters. These values are then used in macro RARGFN to generate code for Resource Analyzer region displays.
RARGFN	Resource Analyzer. Macro to generate code for functions selected in macro RARGEN.
RAUSR00	Resource Analyzer. Sample Resource Analyzer user analyzer prototype. This prototype establishes the interfaces to MAINVIEW for IMS and to the IMS main control block (SCD), from which most IMS and OS/390 control blocks can be accessed.
RMUSR01	Resource Monitor. Sample Resource Monitor user analyzer prototype. This DL/I Resource Monitor prototype establishes the interfaces to MAINVIEW for IMS and to the IMS main control block (SCD), from which most IMS and OS/390 control blocks can be accessed.
RTOPTCH	Macro to generate a patch area for MAINVIEW for IMS modules. This macro should not be issued more than once in a module.
SASIRUF	SAS definition of IRUF terminal, program, and transaction accounting records.
TACCOSTR	Transaction Accountant. Sample linkage editor statements to link a user exit (TASEXIT) into the Transaction Accountant version of program TASCOSTR.
TASxxxxx	Sample layouts for the IRUF records used in Transaction Accountant.
TASEXIT	Sample user exit routine for TASCOSTR IRUF summarization.
WATBTRAC	Sample batch JCL to print history trace data.

Appendix D. Customizing Analyzer and Monitor Services

MAINVIEW product environment customization is described in the *MAINVIEW Common Customization Guide*. This appendix describes how to tailor MAINVIEW for DBCTL online services to your site's needs.

MAINVIEW for DBCTL services have a modular, table-driven design so that you can tailor them to meet specific needs—for example, tailoring the services to provide end-user access security (as shown by the examples on page 88).

Modifying Service Characteristics

MAINVIEW for DBCTL services are defined in service tables that specify the characteristics of the service. You can modify service characteristics dynamically by altering the service table entries, or by creating new service table entries. Service tables are located in BBLINK library load modules.

Use BBPARM member IMFSTD00 (Service Table Definition member) to change the IMFSTD00 characteristics of any service or to define new services. Any changes made to IMFSTD00 dynamically modify or create MAINVIEW for DBCTL services when MAINVIEW for DBCTL is started.

Note: The two most common characteristics changed by users of MAINVIEW for DBCTL are the security and title specifications. See "Security for Analyzer and Monitor Services" on page 71 and refer to *Implementing Security for MAINVIEW Products* before you change these specifications.

Adding a Service

To add a service:

1. Code the new service routine using one of the user service prototypes in the BBSAMP data set as a model.
2. Assemble and link edit the new service routine into BBLINK.
3. Add the definition of the new service to the service table using BBPARM member IMFSTD00.
4. Restart BBI-SS PAS so that the IMFSTD00 member is processed.
5. Test the new service.

Service Table Definition

BBPARM member IMFSTD00 is used to define new services and to change the specifications of existing services. A maximum of 50 new services is allowed.

Note: Message IM2103E is issued if the limit of 50 new services is exceeded.

The following rules apply to creating IMFSTD00.

- The IMFSTD00 must be defined in the BBI-SS PAS BBPARM data set. For more information about the BBI-SS PAS, see the *MAINVIEW Common Customization Guide*.
- All 80 columns of each statement can be used for specifying the various parameters and their values. Sequence numbers can be placed in columns 73 to 80, but there must be at least one blank between the last specification and the sequence number.
- All the parameters needed to define a new service or to modify an existing service can be contained in one statement or split over multiple statements.
- A specific parameter and its values must be contained in the same specification statement.
- An asterisk can be placed in column 1 of any statement to designate it as a comment. Comment statements can be interspersed with specification statements.
- Commas can be used as delimiters in statements; leading blanks are ignored.
- Comments are allowed within specification statements if one blank separates the specification from the comment.
- If REQUEST is used, it must be the first parameter in a statement, and it must be followed immediately by the SERVICE parameter. If REQUEST is not specified, the SERVICE parameter must be first in the statement (REQUEST defaults to DEFINE).
- If any syntax errors are found in a request to define or modify a service, the accepted parameters up to the error are used to execute a partial change to the service table.

Table 8 lists the parameters and parameter options for service tables. The default for each parameter is underlined.

Table 8. Service Table Parameters and Options

Parameter	Parameter Description	Options	
REQUEST	Identifies the start of a new service table entry addition or change. It must be the first parameter in a statement. If REQUEST is not specified, it defaults to DEFINE and SERVICE must be the first parameter (see the examples on page 88).	DEFINE MODIFY	Specifies that a new service is to be defined in the MVIMS service table. This new service is listed on the General Commands service display. Specifies that an existing service definition is to be changed.
SERVICE	Identifies the name of the service to be added or modified. **Note:** SERVICE must be the first parameter in a statement if it is used to define a new service instead of using REQUEST (see the examples on page 88).	xxxxx	Specifies the name of the service to be added or modified. Service names can be from 1 to 5 characters. If `REQUEST=DEFINE`, the service name must be unique and not in use at the time of the DEFINE. If `REQUEST=MODIFY`, the service name must be already defined and exist in one of these lists: • Analyzer Display Services • Data Collection Monitors • General Commands **Note:** The first two to three characters of a user service must not be the same as any of these commands. For example, a user service named ATIV could not be used since MAINVIEW for IMS would transfer to AT, the Active Timer Requests display.
IMSREL| RELEASE	Identifies the IMS release for which this service is valid. **Note:** This parameter is required and must follow the SERVICE specification.	nnnn	Specifies the IMS release as • 0510 - IMS 5.1 • 0610 - IMS 6.1 • 0710 - IMS 7.1 • 0810 - IMS 8.1
TITLE	Identifies the title to be given to a service.	x...x	Specifies the title. The title can be from 1 to 24 characters. The title must be contained in single quotation marks if the title contains embedded blanks or commas. If `REQUEST=DEFINE`, the title is assigned to the new service. If `REQUEST=MODIFY`, the existing title is changed.

Table 8. Service Table Parameters and Options (continued)

Parameter	Parameter Description	Options	
ACCESS	Identifies the access code required to use this service.	A	Specifies the access code that is to be matched with the user's authorized access code. Can be any alphabetic character, A through Z. The user's authorized access code is specified by the PMACC parameter in BBPARM member USERID.
TYPE	Identifies the service type.	ANALYZER	Specifies that this is an analyzer service. It can be invoked directly from a terminal and produces a full-screen display. The service may be invoked asynchronously using the SET command to log the display to the image log.
		MONITOR	Specifies that this is a monitor service. It measures the value of one or more system variables at periodic intervals and compares the value to a threshold. The measurements taken by the service may be plotted using PLOT.
SUPP	Identifies the operating system environment in which this service can run.	BOTH	Specifies that this service may be selected in both an XA and a non-XA environment.
		XA	Specifies that this service supports only an XA or ESA environment.
		Note:	Two services can be defined with the same `SERVICE=name` but different `MODULE=names`, one with `SUPP=370` and the other with `SUPP=XA`.
CB	Identifies the data requirements of the service.	LOCAL	Specifies that the service requires no access to data in the IMS control region private area.
		IMS	Specifies that the service requires access to data in the IMS or DBCTL control region private area.
LOG	Identifies the default logging option for analyzer services. It is invalid for monitors. The LOG parameter on a SET request overrides this specification in the service table.	NO	Specifies that the display produced by the service should not be logged.
		SYNC	Specifies that the display produced by the service should be logged when it is invoked directly from a terminal.
		ASYNC	Specifies that the display produced by the service should be logged when it is invoked indirectly as a result of a SET service request.
		BOTH	Specifies the combination of both SYNC and ASYNC.

Table 8. Service Table Parameters and Options (continued)

Parameter	Parameter Description	Options	
MONTYPE	Identifies the type of measurement taken by a monitor. This information is used by the PLOT service and when a monitor warning message is generated.	COUNT	Specifies that the value collected represents the status of the data item when sampled. The value must remain constant or increase with each measurement. It can never decrease.
		AVERAGE	Specifies that the value collected is an average value over an interval.
		STATUS	Specifies that the value collected represents the status of the data item when sampled. The value may decrease, remain constant or increase with each measurement.
		PERCENT	Specifies that the value collected is a percentage of the total of the data item sampled.
		WARNING	Specifies that the monitor does not maintain history information. It is used only to alert the operator of a problem. This type of monitor can never be plotted.
PARM	Indicates whether the service uses a parameter when invoked.	NO	Specifies that a parameter is never allowed when invoking this service.
		OPT	Specifies that a parameter is optional, if one is provided.
		REQ	Specifies that a parameter is required. The service cannot be invoked without it.
PARMTYPE	Identifies the type of parameter needed by this service. If PARM=NO was specified, PARMTYPE should not be used.	NAME	Specifies that the service uses a parameter in name format, from 1 to 8 characters.
		QNAME	Specifies that the service uses a parameter in name format, from 1 to 8 characters. The name can be qualified by using plus signs as positional qualifiers.
		INTEGER	Specifies that the service uses a parameter in integer format, from 1 to 8 decimal digits.
		HEX	Specifies that the service uses a parameter in hexadecimal format, from 1 to 8 hexadecimal digits.
MODULE	Identifies the load module name of the routine to be invoked by MAINVIEW for IMS when this service is requested.	xxxxxxxx	Specifies the load module name. Valid entries are from 1 to 8 characters. The module name defaults to the 1- to 5-character service name specified with the SERVICE parameter.

Parameter Examples

The following example shows how to use service table parameters to change the security access code of an existing service:

```
REQUEST=MODIFY,SERVICE=DBST,IMSREL=0x10,ACCESS=B
```

In the following example, SERVICE is used to define a new service (REQUEST defaults to DEFINE):

```
SERVICE=NEWSV,IMSREL=0x10,TYPE=ANALYZER,
TITLE='new service title',ACCESS=A
```

In this example, NEWSV is to be added as a new service.

Service Logging

A service can be recorded to an image log data set synchronously and asynchronously:

- Synchronous logging occurs when a service is requested from a terminal and the service table specifies logging. The type of service logging can also be changed dynamically on the panel when the display is requested (these displays are written to the TS Image log).
- Asynchronous logging occurs as a result of a timer-driven request to log an analyzer service or a general command display, such as logging PLOT or DMON, of monitor measurement values. These displays are written to the BBI-SS PAS Image log.

To change the default logging option for any service, see the LOG parameter definition on page 84. For example, the following statement in BBPARM member IMFSTD00 changes the logging option for the DREGN service:

```
REQUEST=MOD,SERVICE=DREGN,LOG=NO
```

Appendix E. Message Changes

Messages were changed for versions 3.3.30, 3.3.10, and 3.3.00 of MAINVIEW for IMS Online, MAINVIEW for IMS Offline, and MAINVIEW for DBCTL, and new messages were added for version 3.3.00. The message changes and additions are listed in this appendix.

Message Changed with Version 3.3.30

Table 10 shows the message that was changed with version 3.3.30.

Table 9. Message Changed with Version 3.3.30

Message ID	Key	Message Text	Sub-system
IM0203I	old	`IMF EVENT COLLECTOR ACTIVE FOR REGION ID=xxx` `IMF EVENT COLLECTOR ACTIVE FOR CCTL REGION ID=xxx`	IMS
IM0203I	new	`EVENT COLLECTOR ACTIVE FOR REGION ID=xxx JOBNAME=yyyyyyyy` `EVENT COLLECTOR ACTIVE FOR CCTL REGION ID=xxx JOBNAME=yyyyyyyy`	IMS

Messages Changed with Version 3.3.10

Table 10 lists product initialization messages that were changed with version 3.3.10.

Table 10. Product Initialization Messages Changed with Version 3.3.10

Message ID	Message Text	Sub-system
IM1113I	`MAINVIEW FOR IMS ONLINE 3.3.10 INITIALIZED  PUT` (*level*)	PAS
IM1113I	`MAINVIEW FOR IMS DBCTL  3.3.10 INITIALIZED  PUT` (*level*)	PAS
IM0109I	`MAINVIEW FOR IMS 3.3.10 CICS=YES     OPTION ACTIVE`	IMS
IM0109I	`MAINVIEW FOR DBCTL 3.3.10 CICS=YES     OPTION ACTIVE`	IMS
IM0113I	`MAINVIEW FOR IMS ONLINE  3.3.10 ACTIVE  PUT` (*level*)	IMS
IM0113I	`MAINVIEW FOR IMS OFFLINE 3.3.10 ACTIVE  PUT` (*level*)	IMS
IM0113I	`MAINVIEW FOR DBCTL 3.3.10 ACTIVE  PUT` (*level*)	IMS

Messages Added with Version 3.3.00

Table 11 lists product initialization messages that were added with version 3.3.00.

Table 11. Product Initialization Messages Added with Version 3.3.00

Message ID	Message Text	Sub-system
IM0202I	`MAINVIEW FOR IMS   DD FUNCTION ENABLED     IMSID`	IMS
IM0202I	`MAINVIEW FOR DBCTL DD FUNCTION ENABLED     IMSID`	IMS
IM1113I	`MAINVIEW FOR IMS ONLINE 3.3.0 INITIALIZED  SSID` (This message is produced if you are using the new ION key.)	PAS
IM1113I	`PRODUCT IMF VERSION 3.3.0     INITIALIZED  SSID` (This message is produced if you are using the old IRA, IRW, IWA, IWM keys.)	PAS
IM0113I	`MAINVIEW FOR IMS OFFLINE 3.3.0 ACTIVE`	IMS

Note: An IOF initialization key is required for the MAINVIEW for IMS Offline components.

Messages Removed with Version 3.3.00

Table 12 lists product initialization messages that were removed with version 3.3.00, and lists the new messages that replaced them.

Table 12. Product Initialization Messages Removed with Version 3.3.00

Message ID	Message Text	Replaced with	Subsystem
IM0201I	`IMF/EC DD FUNCTION ENABLED IMSID`	IM0109I	IMS
IM1202I	`MAINVIEW FOR DBCTL VERSION 3.2.0 ACTIVE`	IM0110I	IMS
IM0131I	`IMF FAST PATH OPTION ACTIVE`	—	IMS
IM1114I	`IMF COMPONENT RA KEY NOT SELECTED`	IM1118W	PAS
IM1114I	`IMF COMPONENT RM KEY NOT SELECTED`	IM1118W	PAS
IM1114I	`IMF COMPONENT WA KEY NOT SELECTED`	IM1118W	PAS
IM1114I	`IMF COMPONENT WM KEY NOT SELECTED`	IM1118W	PAS
IM1114I	`IMF EXTENSION LM KEY NOT SELECTED`	IM1118W	PAS
IM1114I	`IMF EXTENSION AD KEY NOT SELECTED`	IM1118W	PAS
IM1114I	`IMF EXTENSION MD KEY NOT SELECTED`	IM1118W	PAS

Note: The IM0131I `IMF FAST PATH OPTION ACTIVE` message is no longer produced. Versions 3.3.00 and 3.3.10 of MVIMS Online, MVIMS Offline, and MVDBC continue to provide the same Fast Path support provided with the Release 3.2 versions of the products.

Messages Changed with Version 3.3.00

Table 13 lists the MVIMS Online product initialization messages that were changed with version 3.3.00.

Note: The messages identified as “old” in the Key column are produced if you use the old IRA, IRW, IWA, and IWM keys. The messages identified as “new” in the Key column are produced if you use the new ION key.

Table 13. Product Initialization Messages Changed with MVIMS Online Version 3.3.00

Message ID	Key	Message Text	Sub-system
IM0100I	old	IMF/EC COMPONENT VERSION 3.2.0 ACTIVE	IMS
IM0110I	new	MAINVIEW FOR IMS 3.3.0 EVENT COLLECTOR ACTIVE IMSID	IMS
IM0109I	old	CICS=YES IN EFFECT IMSID	IMS
	new	MAINVIEW FOR IMS 3.3.0 CICS=YES OPTION ACTIVE IMSID	IMS
IM0120I	old	- IMF EVENT COLLECTOR NOT STARTED - NO USING COMPONENTS WERE INITIALIZED	IMS
	new	MAINVIEW FOR IMS OR DBCTL FOUND NO VALID PRODUCT KEYS	IMS
IM0109I	old	PRODUCT IMF VERSION 3.2.0 INITIALIZED SSID	PAS
	old	PRODUCT IMF VERSION 3.3.0 INITIALIZED SSID	PAS
	new	MAINVIEW FOR IMS ONLINE 3.3.0 INITIALIZED SSID	PAS
IM1118W	old	WARNING: ALL IMF COMPONENTS FAILED TO ACTIVATE SSID	PAS
	new	WARNING: MAINVIEW FOR IMS FAILED TO ACTIVATE SSID	PAS

Table 14 lists the MVDBC product initialization messages that were changed with version 3.3.00.

Note: The messages identified as "old" in the Key column are produced if you use the old IRA, IRW, IWA, and IWM keys. The messages identified as "new" in the Key column are produced if you use the DBC key.

Table 14. Product Initialization Messages Changed with MVDBC Version 3.3.00

Message ID	Key	Message Text	Sub-system
IM0100I	old	`IMF EVENT COLLECTOR ACTIVE - VERSION = MAINVIEW FOR DBCTL 3.2.0`	IMS
	new	`MAINVIEW FOR DBCTL 3.3.0 EVENT COLLECTOR ACTIVE      IMSID`	IMS
IM0109I	old	`CICS=YES     IN EFFECT IMSID`	IMS
	new	`MAINVIEW FOR DBCTL 3.3.0 CICS=YES     OPTION ACTIVE  IMSID`	IMS
IM0120I	old	`- IMF EVENT COLLECTOR NOT STARTED - NO USING COMPONENTS WERE INITIALIZED`	IMS
	new	`MAINVIEW FOR IMS OR DBCTL FOUND NO VALID PRODUCT KEYS`	IMS
IM0109I	old	`MAINVIEW FOR DBCTL 3.2.0  INITIALIZED    SSID`	PAS
	new	`MAINVIEW FOR DBCTL 3.3.0      INITIALIZED  SSID`	PAS
IM1118W	old	`WARNING: ALL IMF COMPONENTS FAILED TO ACTIVATE  SSID`	PAS
	new	`WARNING: MAINVIEW FOR IMS FAILED TO ACTIVATE  SSID`	PAS

Index

U

V

W

END USER LICENSE AGREEMENT
NOTICE

BY OPENING THE PACKAGE, INSTALLING, PRESSING "AGREE" OR "YES" OR USING THE PRODUCT, THE ENTITY OR INDIVIDUAL ENTERING INTO THIS AGREEMENT AGREES TO BE BOUND BY THE FOLLOWING TERMS. IF YOU DO NOT AGREE WITH ANY OF THESE TERMS, DO NOT INSTALL OR USE THE PRODUCT, PROMPTLY RETURN THE PRODUCT TO BMC OR YOUR BMC RESELLER, AND IF YOU ACQUIRED THE LICENSE WITHIN 30 DAYS OF THE DATE OF YOUR ORDER CONTACT BMC OR YOUR BMC RESELLER FOR A REFUND OF LICENSE FEES PAID. IF YOU REJECT THIS AGREEMENT, YOU WILL NOT ACQUIRE ANY LICENSE TO USE THE PRODUCT.

This Agreement ("**Agreement**") is between the entity or individual entering into this Agreement ("You") and BMC Software Distribution, Inc., a Delaware corporation located at 2101 CityWest Blvd., Houston, Texas, 77042, USA or its affiliated local licensing entity ("BMC"). "You" includes you and your Affiliates. "Affiliate" is defined as an entity which controls, is controlled by or shares common control with a party. THIS AGREEMENT WILL APPLY TO THE PRODUCT, UNLESS (1) YOU AGREED TO A WEB BASED LICENSE AGREEMENT WITH BMC WHEN ORDERING THE PRODUCT, IN WHICH CASE THAT WEB BASED LICENSE AGREEMENT GOVERNS THE USE OF THE PRODUCT, OR (2) IF YOU DID NOT AGREE TO A WEB BASED LICENSE AGREEMENT WITH BMC WHEN ORDERING THE PRODUCT AND YOU HAVE A WRITTEN LICENSE AGREEMENT WITH BMC, THEN THAT WRITTEN AGREEMENT GOVERNS THE USE OF THE PRODUCT. THE ELECTRONIC AGREEMENT PROVIDED WITH THE PRODUCT AS PART OF THE INSTALLATION OF THE PRODUCT WILL NOT APPLY. In addition to the restrictions imposed under this Agreement, any other usage restrictions contained in the Product installation instructions or release notes shall apply to Your use of the Product.

PRODUCT AND CAPACITY. "**Software**" means the object code version of the computer programs provided, via delivery or electronic transmission, to You. Software includes computer files, enhancements, maintenance modifications, upgrades, updates, bug fixes, and error corrections.

"**Documentation**" means all written or graphical material provided by BMC in any medium, including any technical specifications, relating to the functionality or operation of the Software.

"**Product**" means the Software and Documentation.

"**License Capacity**" means the licensed capacity for the Software with the pricing and other license defining terms, including capacity restrictions, such as tier limit, total allowed users, gigabyte limit, quantity of Software, and/or other capacity limitations regarding the Software. For licenses based on the power of a computer, You agree to use BMC's current computer classification scheme, which is available at http://www.bmc.com or can be provided to You upon request.

ACCEPTANCE. The Product is deemed accepted by You, on the date that You received the Product from BMC.

LICENSE. Subject to the terms of this Agreement, as well as Your payment of applicable fees, BMC grants You a non-exclusive, non-transferable, perpetual (unless a term license is provided on an order) license for each copy of the Software, up to the License Capacity, to do the following:

(a) install the Software on Your owned or leased hardware located at a facility owned or controlled by You in the country where You acquired the license;

(b) operate the Software solely for processing Your own data in Your business operations; and

(c) make one copy of the Software for backup and archival purposes only (collectively a "**License**").

If the Software is designed by BMC to permit you to modify such Software, then you agree to only use such modifications or new software programs for Your internal purposes or otherwise consistent with the License. BMC grants You a license to use the Documentation solely for Your internal use in Your operations.

LICENSE UPGRADES. You may expand the scope of the License Capacity only pursuant to a separate agreement with BMC for such expanded usage and Your payment of applicable fees. There is no additional warranty period or free support period for license upgrades.

RESTRICTIONS: You agree to **NOT**:
(a) disassemble, reverse engineer, decompile or otherwise attempt to derive any Software from executable code;

(b) distribute or provide the Software to any third party (including without limitation, use in a service bureau, outsourcing environment, or processing the data of third parties, or for rental, lease, or sublicense); or

(c) provide a third party with the results of any functional evaluation or benchmarking or performance tests, without BMC's prior written approval, unless prohibited by local law.

TRIAL LICENSE. If, as part of the ordering process, the Product is provided on a trial basis, then these terms apply: (i) this license consists solely of a non-exclusive, non-transferable evaluation license to operate the Software for the period of time specified from BMC or, if not specified, a 30 day time period ("**Trial Period**") only for evaluating whether You desire to acquire a capacity-based license to the Product for a fee; and (ii) Your use of the Product is on an AS IS basis without any warranty, and **BMC, ITS AFFILIATES AND RESELLERS, AND LICENSORS DISCLAIM ANY AND ALL WARRANTIES (INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT) AND HAVE NO LIABILITY WHATSOEVER RESULTING FROM THE USE OF THIS PRODUCT UNDER THIS TRIAL LICENSE ("Trial License")**. BMC may terminate for its convenience a Trial License upon notice to You. When the Trial Period ends, Your right to use this Product automatically expires. If You want to continue Your use of the Product beyond the Trial Period, contact BMC to acquire a capacity-based license to the Product for a fee.

TERMINATION. This Agreement shall immediately terminate if You breach any of its terms. Upon termination, for any reason, You must uninstall the Software, and either certify the destruction of the Product or return it to BMC.

OWNERSHIP OF THE PRODUCT. BMC or its Affiliates or licensors retain all right, title and interest to and in the BMC Product and all intellectual property, informational, industrial property and proprietary rights therein. BMC neither grants nor otherwise transfers any rights of ownership in the BMC Product to You. BMC Products are protected by applicable copyright, trade secret, and industrial and intellectual property laws. BMC reserves any rights not expressly granted to You herein.

CONFIDENTIAL AND PROPRIETARY INFORMATION. The BMC Products are and contain valuable confidential information of BMC ("**Confidential Information**"). Confidential Information means non-public technical and non-technical information relating to the BMC Products and Support, including, without limitation, trade secret and proprietary information, and the structure and organization of the Software. You may not disclose the Confidential Information to third parties. You agree to use all reasonable efforts to prevent the unauthorized use, copying, publication or dissemination of the Product.

WARRANTY. Except for a Trial License, BMC warrants that the Software will perform in substantial accordance with the Documentation for a period of one year from the date of the order. This warranty shall not apply to any problems caused by software or hardware not supplied by BMC or to any misuse of the Software.

EXCLUSIVE REMEDY. BMC's entire liability, and Your exclusive remedy, for any defect in the Software during the warranty period or breach of the warranty above shall be limited to the following: BMC shall use reasonable efforts to remedy defects covered by the warranty or replace the defective Software within a reasonable period of time, or if BMC cannot remedy or replace such defective copy of the Software, then BMC shall refund the amount paid by You for the License for that Software. BMC's obligations in this section are conditioned upon Your providing BMC prompt access to the affected Software and full cooperation in resolving the claim.

DISCLAIMER. EXCEPT FOR THE EXPRESS WARRANTIES ABOVE, THE PRODUCT IS PROVIDED "AS IS." BMC, ITS AFFILIATES AND LICENSORS SPECIFICALLY DISCLAIM ALL OTHER WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. BMC DOES NOT WARRANT THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR FREE, OR THAT ALL DEFECTS CAN BE CORRECTED.

DISCLAIMER OF DAMAGES. IN NO EVENT IS BMC, ITS AFFILIATES OR LICENSORS LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES RELATING TO OR ARISING OUT OF THIS AGREEMENT, SUPPORT, AND/OR THE PRODUCT (INCLUDING, WITHOUT LIMITATION, LOST PROFITS, LOST COMPUTER USAGE TIME, AND DAMAGE OR LOSS OF USE OF DATA), EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, AND IRRESPECTIVE OF ANY NEGLIGENCE OF BMC OR WHETHER SUCH DAMAGES RESULT FROM A CLAIM ARISING UNDER TORT OR CONTRACT LAW.

LIMITS ON LIABILITY. BMC'S AGGREGATE LIABILITY FOR DAMAGES IS LIMITED TO THE AMOUNT PAID BY YOU FOR THE LICENSE TO THE PRODUCT.

SUPPORT. If Your order includes support for the Software, then BMC agrees to provide support (24 hours a day/7 days a week) ("**Support**"). You will be automatically re-enrolled in Support on an annual basis unless BMC receives notice of termination from You as provided below. There is a free support period during the one year warranty period.

(a) **Support Terms**. BMC agrees to make commercially reasonable efforts to provide the following Support: (i) For malfunctions of supported versions of the Software, BMC provides bug fixes, patches or workarounds in order to cause that copy of the Software to operate in substantial conformity with its then-current operating specifications; and (ii) BMC provides new releases or versions, so long as such new releases or versions are furnished by BMC to all other enrolled Support customers without additional charge. BMC may refuse to provide Support for any versions or releases of the Software other than the most recent version or release of such Software made available by BMC. Either party may terminate Your enrollment in Support upon providing notice to the other at least 30 days prior to the next applicable Support anniversary date. If You re-enroll in Support, BMC may charge You a reinstatement fee of 1.5 times what You would have paid if You were enrolled in Support during that time period.

(b) **Fees**. The annual fee for Support is 20% of the Software's list price less the applicable discount or a flat capacity based annual fee. BMC may change its prices for the Software and/or Support upon at least 30 days notice prior to Your support anniversary date.

VERIFICATION. If requested by BMC, You agree to deliver to BMC periodic written reports, whether generated manually or electronically, detailing Your use of the Software in accordance with this Agreement, including, without limitation, the License Capacity. BMC may, at its expense, audit Your use of the Software to confirm Your compliance with the Agreement. If an audit reveals that You have underpaid fees, You agree to pay such underpaid fees. If the underpaid fees exceed 5% of the fees paid, then You agree to also pay BMC's reasonable costs of conducting the audit.

EXPORT CONTROLS. You agree not to import, export, re-export, or transfer, directly or indirectly, any part of the Product or any underlying information or technology except in full compliance with all United States, foreign and other applicable laws and regulations.

GOVERNING LAW. This Agreement is governed by the substantive laws in force, without regard to conflict of laws principles: (a) in the State of New York, if you acquired the License in the United States, Puerto Rico, or any country in Central or South America; (b) in the Province of Ontario, if you acquired the License in Canada (subsections (a) and (b) collectively referred to as the "**Americas Region**"); (c) in Singapore, if you acquired the License in Japan, South Korea, Peoples Republic of China, Special Administrative Region of Hong Kong, Republic of China, Philippines, Indonesia, Malaysia, Singapore, India, Australia, New Zealand, or Thailand (collectively, "**Asia Pacific Region**"); or (d) in the Netherlands, if you acquired the License in any other country not described above. The United Nations Convention on Contracts for the International Sale of Goods is specifically disclaimed in its entirety.

ARBITRATION. ANY DISPUTE BETWEEN YOU AND BMC ARISING OUT OF THIS AGREEMENT OR THE BREACH OR ALLEGED BREACH, SHALL BE DETERMINED BY BINDING ARBITRATION CONDUCTED IN ENGLISH. IF THE DISPUTE IS INITIATED IN THE AMERICAS REGION, THE ARBITRATION SHALL BE HELD IN NEW YORK, U.S.A., UNDER THE CURRENT COMMERCIAL OR INTERNATIONAL, AS APPLICABLE, RULES OF THE AMERICAN ARBITRATION ASSOCIATION. IF THE DISPUTE IS INITIATED IN A COUNTRY IN THE ASIA PACIFIC REGION, THE ARBITRATION SHALL BE HELD IN SINGAPORE, SINGAPORE UNDER THE CURRENT UNCITRAL ARBITRATION RULES. IF THE DISPUTE IS INITIATED IN A COUNTRY OUTSIDE OF THE AMERICAS REGION OR ASIA PACIFIC REGION, THE ARBITRATION SHALL BE HELD IN AMSTERDAM, NETHERLANDS UNDER THE CURRENT UNCITRAL ARBITRATION RULES. THE COSTS OF THE ARBITRATION SHALL BE BORNE EQUALLY PENDING THE ARBITRATOR'S AWARD. THE AWARD RENDERED SHALL BE FINAL AND BINDING UPON THE PARTIES AND SHALL NOT BE SUBJECT TO APPEAL TO ANY COURT, AND MAY BE ENFORCED IN ANY COURT OF COMPETENT JURISDICTION. NOTHING IN THIS AGREEMENT SHALL BE DEEMED AS PREVENTING EITHER PARTY FROM SEEKING INJUNCTIVE RELIEF FROM ANY COURT HAVING JURISDICTION OVER THE PARTIES AND THE SUBJECT MATTER OF THE DISPUTE AS NECESSARY TO PROTECT EITHER PARTY'S CONFIDENTIAL INFORMATION, OWNERSHIP, OR ANY OTHER

PROPRIETARY RIGHTS. ALL ARBITRATION PROCEEDINGS SHALL BE CONDUCTED IN CONFIDENCE, AND THE PARTY PREVAILING IN ARBITRATION SHALL BE ENTITLED TO RECOVER ITS REASONABLE ATTORNEYS' FEES AND NECESSARY COSTS INCURRED RELATED THERETO FROM THE OTHER PARTY.

U.S. GOVERNMENT RESTRICTED RIGHTS. The Software under this Agreement is "commercial computer software" as that term is described in 48 C.F.R. 252.227-7014(a)(1). If acquired by or on behalf of a civilian agency, the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of this Agreement as specified in 48 C.F.R. 12.212 (Computer Software) and 12.211 (Technical Data) of the Federal Acquisition Regulations ("**FAR**") and its successors. If acquired by or on behalf of any agency within the Department of Defense ("**DOD**"), the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of this Agreement as specified in 48 C.F.R. 227.7202 of the DOD FAR Supplement and its successors.

MISCELLANEOUS TERMS. You agree to pay BMC all amounts owed no later than 30 days from the date of the applicable invoice, unless otherwise provided on the order for the License to the Products. You will pay, or reimburse BMC, for taxes of any kind, including sales, use, duty, tariffs, customs, withholding, property, value-added (VAT), and other similar federal, state or local taxes (other than taxes based on BMC's net income) imposed in connection with the Product and/or the Support. This Agreement constitutes the entire agreement between You and BMC and supersedes any prior or contemporaneous negotiations or agreements, whether oral, written or displayed electronically, concerning the Product and related subject matter. No modification or waiver of any provision hereof will be effective unless made in a writing signed by both BMC and You. You may not assign or transfer this Agreement or a License to a third party without BMC's prior written consent. Should any provision of this Agreement be invalid or unenforceable, the remainder of the provisions will remain in effect. The parties have agreed that this Agreement and the documents related thereto be drawn up in the English language. Les parties exigent que la présente convention ainsi que les documents qui s'y rattachent soient rédigés en anglais.

SW EULA Int 030102

Notes

41325