MainView for IMS and MainView for DBCTL
Customization Guide

Supporting

Version 5.1 of MainView for IMS Online
Version 5.1 of MainView for IMS Offline
Version 5.1 of MainView for DBCTL

January 2014
Contacting BMC Software

Several methods are available for contacting BMC Software.

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

United States and Canada

Address  BMC SOFTWARE INC  Telephone  1 713 918 8800  Fax  1 713 918 8000
2101 CITYWEST BLVD  or
HOUSTON TX 77042-2827 USA  1 800 841 2031

Outside United States and Canada

Telephone  +01 713 918 8800  Fax  +01 713 918 8000

© Copyright  2001-2014 BMC Software, Inc.

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

CICS, DB2, IBM, IMS, MQSeries, MVS, System z, and VTAM are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

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

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

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

Restricted rights legend

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 UNPUBLISHED—RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES.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

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

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

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

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

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

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

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

About this book

Related publications ................................................................. 11

Conventions ........................................................................... 12

Chapter 1 Compatibility and CPU usage 13

Compatibility with previous releases .................................... 13
  Downward compatibility .................................................... 13
  Upward compatibility .......................................................... 13
AO exit compatibility ............................................................ 14
IMS log compatibility ............................................................ 14
CPU time precision compatibility ........................................... 14
IRUF records compatibility ................................................... 14
CPU usage by the Event Collector ......................................... 15
CPU usage by the samplers .................................................. 15

Chapter 2 Implement a product 17

Delete modules from a prior release .................................... 18
Set up customer IDs ............................................................... 19
Establish daily cut-off time .................................................. 21
Set up BBPARM data sets ...................................................... 21
Customizing BBPARM members for an IMS system .............. 23
Activating a product ............................................................. 27
Define target systems ............................................................ 28
Activate line commands in views ......................................... 28
Modifying the IMS control region JCL .................................. 29
  Allocate a BBPARM data set .............................................. 29
  Give IMS access to the Event Collector .............................. 29
Enabling AO exit routines ..................................................... 30
  Flow of control ................................................................. 32
  Enhancement to the IMS DFSAOE00 interface ..................... 32
  MainView AutoOPERATOR for IMS considerations ............. 33
Use Energizer for IMS Connect ............................................. 33
  Creating a product-specific logspace ................................... 34
  Implementation methods .................................................. 35
Find messages in MainView Logger ....................................... 36
Dynamic install and uninstall ................................................ 36
### Chapter 3 Customize the trace facility

- Specify trace defaults in IMFBEX00 ........................................... 41
- Detail trace data collection .......................................................... 42
- Trace display buffer size ............................................................. 43
- Trace duration .............................................................................. 44
- Trace logging options ................................................................... 44
- Update trace defaults ................................................................... 45
- Specify detail trace buffers ............................................................. 45
- Set up and maintain a trace directory ........................................ 47
  - Defining and initializing a trace directory data set ......................... 47
  - Identify the trace directory to BBI .............................................. 47
  - Verifying trace directory entries ................................................. 48
- Manage trace log data sets ............................................................. 49
  - Define a trace log data set ......................................................... 49
  - Archive a trace log data set ...................................................... 49
  - Restore an archived trace log data set ........................................ 50
  - Create a trace log data set from the IMS log ............................... 50
- Print a trace log data set ............................................................... 50
- Set up workload history traces .................................................... 51
- Set limits for transaction trace views ........................................... 52

### Chapter 4 Control thresholds, data collection, and record processing

- Specify thresholds for IMS and IMS sysplex activity views ........... 55
- Control shared message queue data collection ............................. 56
  - Control data query time out ..................................................... 56
  - Control refresh frequency ....................................................... 57
  - CQS information views with message age fields ....................... 57
  - Non-workflow views with message count fields ....................... 58
  - Workflow views with message count fields ............................... 59
- Control resource records processed for cross-reference views ....... 59

### Chapter 5 Event Collector usage

- Event Collector overview ............................................................ 61
- Start and stop the Event Collector ............................................... 63
- Data collection parameters .......................................................... 63
- CPU timing parameters ............................................................... 64
  - Application program CPU ....................................................... 64
  - Message DL/I CPU ............................................................... 65
Appendix B  BBSAMP data set members  123
Descriptions of several BBSAMP members ..............................................................123

Appendix C  Specify product keys  127
Determining the internal CPU model and version number .......................................127
Creating and modifying BBPARM member BBKEYS .............................................129

Appendix D  Message changes  131
Version 5.1.00 messages ...............................................................................................131
Version 4.6.00 message changes .................................................................................132
Version 4.5.00 message changes .................................................................................133
About this book

This book documents how to implement, configure, and customize the MainView for IMS Online, MainView for IMS Offline, and MainView for DBCTL products for your use.

For information about the features, functions, and use of the products, see the following publications:

- MainView for IMS Online User Guide
- MainView for IMS Offline User Guide
- MainView for DBCTL User Guide

For information about new features in the current release of a product, see the product's release notes, which are available on the BMC Support Central website at http://www.bmc.com/support.

The software also offers online Help. To access Help, press F1 or click Help.

Related publications

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

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

- View BMC Quick Course demos (short overviews of selected product concepts, tasks, or features), which are included in the BMC Documentation Center.

- Read individual product documents (books and notices) within the “A – Z Supported Product List.”

You can order hardcopy documentation from your BMC sales representative or from the support site. You can also subscribe to proactive alerts to receive e-mail messages when notices are issued.
Tip
You can access the BMC Support Central site at http://www.bmc.com/support.

Conventions

This document uses the following special conventions:

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

■ Variable text in path names, system messages, or syntax is displayed in italic text: `tests/instance/file`.

■ This document uses a symbol to show menu sequences. For example, Actions => Create Test instructs you to choose the Create Test command from the Actions menu.
Compatibility and CPU usage

You should know about certain compatibility and CPU usage information before using MainView for IMS Online, MainView for IMS Offline, or MainView for DBCTL.

Compatibility with previous releases

BMC recommends that the UAS (user address space) 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 product 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 the product 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 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, see the release notes, which are available on the BMC Customer Support website at http://www.bmc.com/support.
AO exit compatibility

If a previous version of the BMC AO exit was installed on the target IMS, make sure that all remnants of the previously installed version are removed.

Instructions for removing previously installed versions are in “Delete modules from a prior release” on page 18.

IMS log compatibility

Log Edit will process IMS logs created by the MainView for IMS product version 4.4 and later.

CPU time precision compatibility

The precision of the CPU time on reports has increased from thousandths of a second to microseconds. As a result, the format of the TAR records has changed.

If you have IRUF processing programs, you might need to change them to work with the change in data format of these fields.

The provided IMF46TAR conversion utility reads an IRUF as input and either

- Converts the current version of the IRUF to a 4.6 formatted version
- Takes a 4.6 formatted version of the IRUF and creates an IRUF that does not have the enhanced precision timing in it

See sample JCL in the BBSAMP member IMF46TAR.

IRUF records compatibility

The Performance Reporter and Transaction Accountant components (and associated utilities) will process IRUF records created with MainView for IMS version 4.6 and later.
CPU usage by the Event Collector

The Event Collector is always active, even when there are no active traces or workload monitors.

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU=ALL</td>
<td><em>(Highest CPU usage)</em> Collects all CPU time data</td>
</tr>
<tr>
<td>CPU=DEP</td>
<td><em>(The default)</em> Collects CPU time data from dependent regions</td>
</tr>
<tr>
<td>CPU=DEPPGM</td>
<td>Collects CPU time data from dependent region activities, including DB2 CPU time, and times the entire transaction as a single event. This option reduces overhead significantly.</td>
</tr>
<tr>
<td>CPU=DEPDB2</td>
<td>Functions the same as the DEPPGM option except that CPU for DB2 events are reported separately from dependent region CPU time. CPU usage depends upon the number of SQL calls issued by the transaction.</td>
</tr>
<tr>
<td>CPU=NONE</td>
<td><em>(Lowest CPU usage)</em> CPU time data is not collected</td>
</tr>
</tbody>
</table>

For more information about the Event Collector options, see “Event Collector usage” on page 61.

**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 by the samplers

The product 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 User Guide* or the *MainView for IMS Online User Guide*.
Implement a product

The Installation System installs a default configuration for the products.

After using the Installation System, there are additional tasks to complete and changes you might want to make to the default configuration. Use the information in the following table as a roadmap of what to review:

<table>
<thead>
<tr>
<th>Section to review</th>
<th>Product used by</th>
<th>Exceptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Compatibility and CPU usage” on page 13</td>
<td>MainView for IMS Online</td>
<td>“Use Energizer for IMS Connect ” on page 33 only applies to MainView for IMS Online</td>
</tr>
<tr>
<td></td>
<td>MainView for IMS Offline</td>
<td>“Activate line commands in views” on page 28 only applies to MainView for IMS Online and MainView for DBCTL</td>
</tr>
<tr>
<td></td>
<td>MainView for DBCTL</td>
<td>“Set up customer IDs” on page 19 and “Establish daily cut-off time” on page 21 only apply to MainView for IMS Online</td>
</tr>
<tr>
<td>“Implement a product” on page 17</td>
<td>MainView for IMS Online</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MainView for IMS Offline</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MainView for DBCTL</td>
<td></td>
</tr>
<tr>
<td>“Customize the trace facility” on page 41</td>
<td>MainView for IMS Online</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MainView for DBCTL</td>
<td></td>
</tr>
<tr>
<td>“Control thresholds, data collection, and record processing” on page 55</td>
<td>MainView for IMS Online</td>
<td>“Control shared message queue data collection” on page 56 only applies to MainView for IMS Online</td>
</tr>
<tr>
<td></td>
<td>MainView for DBCTL</td>
<td></td>
</tr>
<tr>
<td>“Event Collector usage” on page 61</td>
<td>MainView for IMS Online</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MainView for IMS Offline</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MainView for DBCTL</td>
<td></td>
</tr>
<tr>
<td>“Define target systems” on page 28</td>
<td>MainView for IMS Online</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MainView for IMS Offline</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MainView for DBCTL</td>
<td></td>
</tr>
<tr>
<td>Section to review</td>
<td>Product used by</td>
<td>Exceptions</td>
</tr>
<tr>
<td>------------------------------------------------------</td>
<td>------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>“Customize user exit routines” on page 109</td>
<td>MainView for IMS Online</td>
<td>“Log edit user exit routines” on page 111 and “Summarized IRUF exit” on page 114 only apply to MainView for IMS Offline</td>
</tr>
<tr>
<td></td>
<td>MainView for IMS Offline</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MainView for DBCTL</td>
<td></td>
</tr>
<tr>
<td>“Optional modifications” on page 117</td>
<td>MainView for IMS Online</td>
<td>“Modifications to reconcile response time monitoring data” on page 118 only applies to MainView for IMS Online</td>
</tr>
<tr>
<td></td>
<td>MainView for IMS Offline</td>
<td>“Modification to suppress IMFLEDIT warning messages” on page 120 only applies to MainView for IMS Offline</td>
</tr>
<tr>
<td></td>
<td>MainView for DBCTL</td>
<td></td>
</tr>
<tr>
<td>“IMS dump analysis” on page 121</td>
<td>MainView for IMS Online</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MainView for DBCTL</td>
<td></td>
</tr>
<tr>
<td>“BBSAMP data set members” on page 123</td>
<td>MainView for IMS Online</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MainView for IMS Offline</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MainView for DBCTL</td>
<td></td>
</tr>
<tr>
<td>“Specify product keys” on page 127</td>
<td>MainView for IMS Online</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MainView for DBCTL</td>
<td></td>
</tr>
<tr>
<td>“Message changes” on page 131</td>
<td>MainView for IMS Online</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MainView for IMS Offline</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MainView for DBCTL</td>
<td></td>
</tr>
</tbody>
</table>

Delete modules from a prior release

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

If you installed a prior product release using ICOPY to copy BBLINK members to a STEPLIB data set, use the BBSAMP jobs in Table 1 on page 19 to delete the old modules.

If you did not copy BBLINK members to a data set to install a prior release, skip to “Set up BBPARM data sets” on page 21.
To delete the old modules

1. From Table 1 on page 19, select the delete job that corresponds to the BMC product (or products) and IMS release installed at your site.

2. Edit the delete job you selected to change all &RESLIBs to the name of the data set where the old modules reside.

3. Run the delete job.

4. 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.

5. 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 operating system.

Set up customer IDs

Because the IMS environment does not have batch accounting codes or any similar facility, a unique identifier must be created to associate jobs with specific customers.

This identifier is called the MainView for IMS product customer ID. Both the IMS resource utilization file (IRUF) and the MainView for IMS transaction accounting...
system are based on the customer ID field, which is used to associate all resource usage measurements and costs to a specific user.

The customer ID field is also used by the TASCOSTR program when it performs IRUF summarization.

- Transaction records are summarized by customer ID and transaction code.
- Terminal (LTERM) records are summarized by customer ID and LTERM name.
- Program records are summarized by program name.

**Note**
Setting up customer IDs is only needed for MainView for IMS Offline. Because transactions from different users can be processed in one program scheduling, the MainView for IMS customer ID is not used in program records.

The customer ID is an 18-byte alphanumeric field defined by the user. The field is included in the IRUF through a user exit routine (PRSCEXIT) during Log Edit processing.

The PRSCEXIT user exit routine (described in “Customize user exit routines” on page 109) is distributed with MainView for IMS Offline and has a customer ID default value of LTERM. Although this value can be modified by the user, it is recommended that the default LTERM or equivalent be retained as part of the customer ID. Retaining the default or equivalent ensures maximum efficiency during the IRUF summarization process.

Many of the identification fields available in IRUF records can be used to associate individual records with the right customer. The IRUF record identification fields that can be used to create the MainView for IMS customer ID are listed below. Fields with an asterisk (*) are those that are used in MainView for IMS reports.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGN</td>
<td>PERFORMANCE GROUP</td>
</tr>
<tr>
<td>ASID</td>
<td>PLINE*</td>
</tr>
<tr>
<td>CLASS*</td>
<td>PROGRAM*</td>
</tr>
<tr>
<td>CUSTOMER ID*</td>
<td>PROGRAM TYPE*</td>
</tr>
<tr>
<td>FP ROUTE CODE*</td>
<td>PTERM*</td>
</tr>
<tr>
<td>MVIMS SYSID*</td>
<td>REGION*</td>
</tr>
<tr>
<td>IMSID</td>
<td>SMFID</td>
</tr>
<tr>
<td>LTERM*</td>
<td>TRancode*</td>
</tr>
<tr>
<td>MSC DESTINATION ID</td>
<td>USERID*</td>
</tr>
<tr>
<td>MSC ORIGIN ID</td>
<td>VTAM NODE</td>
</tr>
</tbody>
</table>

20  MainView for IMS and MainView for DBCTL Customization Guide
The customer ID should be structured in the same way as existing site budget and accounting codes. Ideally, the customer ID should be identical, or at least similar, to the accounting codes used in the site batch processing environment.

**Note**

For more information about MainView for IMS summarization or the use of LTERM in the customer ID, see the *MainView for IMS Offline User Guide*.

---

**Establish daily cut-off time**

For sites that execute IMS around the clock, a regular daily cut-off time must be established to ensure a consistent daily unit that can be used to create IRUFs.

The IMFLEDIT program is run against the IMS log to create IRUFs. For more information about IRUFs, see BBSAMP member IMFIRUF and the *MainView for IMS Offline User Guide*.

An IMS cut-off time can occur before or after midnight, whichever is more convenient, but the cut-off should occur at the same time every day. At sites where IMS is operating 24 hours a day, 2400 hours (midnight) is the recommended cut-off time.

**Note**

It is especially important to create a cut-off time at midnight if you want to produce calendar reports or plots based on time of day.

Cut off times are only applicable for MainView for IMS Offline.

---

**Set up BBPARM data sets**

Use this information to set up your BBPARM data sets in the following situations:

- If you did not use MainView Customization to customize your product.

- If you have multiple IMSs, you might want to allocate a BBPARM data set that is unique to an IMS.

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

These customization instructions refer to the following data sets:

- **HLQ.ibbparm**
- **HLQ.UBBPARM**
- **HLQ.BBPARM**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HLQ</strong></td>
<td>High-level data set name qualifier used at your site</td>
</tr>
</tbody>
</table>
| **ibbparm** | 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 MainView Customization.  
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 (*imsidSYS*) version of the IMFSYS00 member, a renamed (*imsidECP*) version of the IMFECP00 member, or both, include them in UBBPARM, and customize them to suit your needs (as explained below in UBBPARM).  
If you require a customized member AAOPRMxx, you can specify the suffix using AAOPRM=xx in the IMFSYS00 (or *imsidSYS*) member in use for this IMS. |
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UBBPARM</td>
<td>Parameter data set that is tailored from the distributed BBPARM data set and is shared by all IMS systems. If you used MainView Customization, you can use the UBBPARM data set created by MainView Customization. If you did not use MainView Customization, 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: <code>imsidSYS</code> (where <code>imsid</code> 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: <code>imsidECP</code> (where <code>imsid</code> is the IMS ID code). If you do not need to customize other UBBPARM members for an IMS system, you do not need to create and allocate an <code>ibbparm</code> data set for that system. You can include AAOPRMxx member by specifying the suffix with the <code>AAOPRM=xx</code> parameter in the IMFSYS00 (or <code>imsidSYS</code>) member in use for this IMS.</td>
</tr>
<tr>
<td>BBPARM</td>
<td>Target BBPARM data set distributed by BMC</td>
</tr>
</tbody>
</table>

**Tip**
Throughout the documentation, parameter library members are normally referred to as BBPARM members, even though customized versions of the members may reside in the UBBPARM data set or in an `ibbparm` data set. The system parameter member is 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 `imsidSYS` or `imsidECP`.

---

**Customizing BBPARM members for an IMS system**

This section is provided primarily for new installations of the product. If you have a prior version installed, you can use your existing parameter settings, but this would be a good time to review them.
Follow the procedures in this topic if you did not use MainView Customization to customize the product.

This 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
- Offload workflow sampler data collection to a zIIP processor

**To customize the BBPARM members**

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.

   - **MSGLVLI**
     Use this parameter to set the Event Collector informational message destination to MTO, WTO, BOTH, or NONE. The parameter affects informational messages only (those with the format IM xxxxI).

   - **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 MainView for IMS Online and the MainView for IMS Offline product 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 passed. For more information, see “Enabling AO exit routines” on page 30.

As distributed, IMFSYS00 activates all MainView for DBCTL components in IMS. You can use insid$SYS in the UBBPARM data set or IMFSYS00 in an iibparm data set to temporarily deactivate one or more components. BBPARM member IMFSYSBB contains information about how to deactivate components. Copy what you need from IMFSYSBB to your UBBPARM insid$SYS member or iibparm IMFSYS00 member.

2 Set up the Event Collector data collection parameters in an IMFEC$CP00 member or an insid$ECP member.

**Note**
Use UBBPARM member IMFEC$CP00 for Event Collector parameters that are to be shared among multiple IMSs. To specify IMS-specific Event Collector parameters, use an insid$ECP member in the UBBPARM data set or an IMFEC$CP00 member in an iibparm data set.

The following parameters set up Event Collector data collection:

- BHTO controls whether IMS buffer handler activity is included with DL/I CPU or timed separately
- BILLOVHD determines how dependent region CPU time is treated
- BMP controls whether activity data for BMP and JBP transactions and programs is collected
- CICS controls whether records are collected for DBCTL CICS and ODBA thread data
- CPU controls the level of CPU data collected
- DBIO controls the level of database I/O data to be collected
- DEPREC controls whether recovery from additional abend conditions in dependent regions is enabled and performed as necessary

The following default values are recommended:

- BMP=YES
- CPU=DEP
- CICS=YES

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 the MainView for IMS Offline product 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:

```
IMLEVEL = 1110 | 1210 | 1310 | 0000
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1110</td>
<td>Specifies IMS version 11.1</td>
</tr>
<tr>
<td>1210</td>
<td>Specifies IMS version 12.1</td>
</tr>
<tr>
<td>1310</td>
<td>Specifies IMS version 13.1</td>
</tr>
<tr>
<td>0000</td>
<td>The Log Edit utility will scan up to the first 50 KB records of the log tape to determine the IMS release</td>
</tr>
</tbody>
</table>

**Note**

For more information about the Event Collector parameters, see “Event Collector usage” on page 61. For more information about the Log Edit utility, IMFLEDIT, see the MainView for IMS Offline User Guide.

4 Set the ZIIPWFL parameter in IMFBEX00 to send workflow sampler data collection processing to a zIIP processor.

Set ZIIPWFL as follows:

- YES to use zIIP processing
- NO (the default) to not use zIIP processing

**Note**

To enable zIIP processing in general for the product address space (PAS), use the USEZIIP parameter in the BBISSP00 member of BBPARM. For more information about the USEZIIP parameter, see the MainView Customization Reference.
Activating a product

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

To enable product initialization

1. In BBPARM member BBISSP00, specify:
   - For MainView for IMS Online, PRODUCT=MVIMS
   - For MainView for IMS Offline, no initialization is required
   - For MainView for DBCTL, PRODUCT=MVDBC

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

To enable product authorization

1. Create the product authorization table during the installation and customization process.

   The table name is:
   - For MainView for IMS Online, IONTBL3P
   - For MainView for IMS Offline, IOFTBL3P
   - For MainView for DBCTL, DBCTBL3P

   For more information, see the *Installation System User Guide*.

2. Add access to the product authorization table in one of the following ways:
   - Concatenate a data set containing the product authorization table to the STEPLIB data set in the IMS control region JCL.
   - Add the following statement to the IMS control region JCL:
     ```java
     //BMCPSSWD DD DSN=hlq.BMCPSWD,DISP=SHR
     ```
   - Copy the product authorization table into RESLIB or BBLINK.
     If you want the product authorization table to reside in RESLIB, see the information about product maintenance in the *Installation System User Guide*.
   - Add the product authorization table data set to LNKLST.
The product initialization messages will now include BBAP prefixed messages issued by the BMC Product Authorization utility.

**Note**
Older implementations of the products use password keys (BB keys). For information about using password keys, see “Specify product keys” on page 127.

## Define target systems

The MainView for IMS and MainView for DBCTL products operate 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 Customization Reference, BBPARM member BBJNT00 is used to define all eligible target systems and associate them with the subsystem IDs of a BBI-SS PAS.

**Note**
Defining target systems is only applicable for MainView for IMSOnline and MainView for DBCTL.

To define a target system:

- Use the jobname of the region with the TARGET parameter in BBJNT00.
- 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=1n10
  ```

## Activate line commands in views

To use the action line commands available on the views, you must:

- Implement MainView AutoOPERATOR
- Enable the RULIPX00 rule set

The RULIPX00 rule set is distributed in the BBPARM data set.

You can find information about using and enabling rule sets in the MainView AutoOPERATOR Basic Automation Guide Volume 1: Using Rules.
Modifying the IMS control region JCL

This procedure describes how to use the IMS control region JCL to start IMS and install MainView for IMS, MainView for DBCTL, and MainView AutoOPERATOR for IMS.

To modify the IMS control region JCL

1. Modify the JCL by allocating a BBPARM data set (see “Allocate a BBPARM data set” on page 29).

2. Provide IMS access to the Event Collector (see “Give IMS access to the Event Collector” on page 29).

Allocate a BBPARM data set

To establish the parameters for MainView for IMS, MainView for DBCTL, or MainView AutoOPERATOR for IMS, allocate the BBPARM data sets by adding the following statements to the IMS control region startup procedure:

```
//IMFPARM DD DSN=hlq.ibbparm
// DD DSN=hlq.UBBPARM
// DD DSN=hlq.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 “Set up BBPARM data sets” on page 21.

Give IMS access to the Event Collector

The IMS control region must be able to access the Event Collector to enable execution of MainView for IMS, MainView for DBCTL, 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:

- ICOPY5 for IMS 11.1
- ICOPY6 for IMS 12.1
- ICOPY7 for IMS 13.1

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

**Note**

If you plan to use dynamic uninstall and install to apply Event Collector or MainView AutoOPERATOR for IMS maintenance to an active IMS system, read “Dynamic install and uninstall” on page 36. That section describes an alternate method of modifying the IMS JCL so that IMS can access the Event Collector.

---

## Enabling AO exit routines

You can skip this section if you have a prior version of the product installed and will use the same method for defining exits in the new version.

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

**Note**

See the IBM *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 the MainView for IMS monitors when IMS starts

■ 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, the product loads and executes any user exit specified in AOEINIT=xxxxxxxx one time only. The DFSAOE00 exit does not forward control to your DFSAOE01 exit on the initialization call.

**To enable AO exit routines**

Perform this procedure if you require a DFSAOUE0 or DFSAOE00 exit routine in addition to the exit provided by BMC.

1. Perform one of the following actions based on the number of DFSAOUE0 exits you have:
   - If you have *one* DFSAOUE0 exit, rename it DFSAOUE1 or choose a different name and use the control statements described for use when you have multiple DFSAOUE0 exits.
   - If you have *multiple* DFSAOUE0 exits in addition to the exit supplied by BMC, or one or more exits not named DFSAOUE1, add one or more AOIEXIT control statements to an imsidSYS member in UBBPARM or to an IMFSYS00 member in an *ibbparm* data set.
     For example:
     
     AOIEXIT=MYNAME
     AOIEXIT=DFSAOUE2
     
     In the example, both exits are loaded and executed by the BMC DFSAOUE0 exit. Program MYNAME executes first.

2. Perform one of the following actions based on the number of DFSAOE00 exits you have:
   - If you have *one* DFSAOE00 exit, rename it to DFSAOE01 or choose a different name and use the control statements described for use when you have multiple DFSAOE00 exits.
   - If you have *multiple* DFSAOE00 exits, in addition to the exit supplied by BMC, or one exit not named DFSAOE01, add one or more AOEEXIT control
Enabling AO exit routines

statements to an imsidSYS member in UBBPARM or to an IMFSYS00 member in an ibbparm data set.
For example:
AOEEXIT=MYNAME
AOEEXIT=DFSAOE02
In the example, both exits are loaded and executed by the BMC DFSAOE00 exit. Program MYNAME executes first.

Flow of control

By default, the BMC DFSAOE00 exit invokes the BMC AO exit routine (IELOAD) before it invokes your user exits. The return code set by the last user exit processed is passed to IMS.

You can change the default processing in an imsidSYS member in the UBBPARM data set or an IMFSYS00 member in an IBBPARM data set by specifying:

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

For example:

AOEEXIT=(DFSAOE01,RC)
AOEEXIT=IELOAD

In the example, 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 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 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

Consider the following information when installing your user exit with a BMC exit:

- If the BMC exit is executed before the user exit and the user exit’s return code cancels the processing of additional message segments of a 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 exit, changes to the messages made by the user exit are received by the MainView AutoOPERATOR AO exit.
  - If the user exit sets the length code of a message (or a segment of a multisegment message) to zero, MainView AutoOPERATOR does not process the message (or segment).
  - If the user exit sets the length code of the first segment to 0 and the return code to 4, all further segments are canceled for MainView AutoOPERATOR also.

Note

If any modules are specified with the AOIEXIT parameter in IMFSYS00 or imsidSYS, but the BMC exit is not specified, the exit still executes after all specified exits.

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

Use Energizer for IMS Connect

If you want to display transaction trace information from Energizer for IMS Connect in the MainView for IMS Online views, you must create a product-specific logspace. If you do not want this information, you can skip this section.
Additionally, you can customize the MainView Logger log views to display information related to IMS only.

**Note**
The use of Energizer *for IMS Connect* is only applicable to MainView *for IMS* Online.

### Creating a product-specific logspace

The information in this section provides specific information for MainView *for IMS* Online.

How to create a product-specific logspace is explained in the *MainView Customization Reference*.

This product-specific logspace is in addition to the required, primary logspace that is created when MainView Logger is installed, and is in addition to any other product-specific logspace you might have.

**Parameter requirements**

The following parameters are required when you use a product-specific logspace:

- When the MainView Logger primary logspace is created, ensure that the following LOGGER00 parameter is set to YES, which is the default setting:
  
  ```
  DEFAULTLOGGER=YES
  ```

- When the BBI-SS PAS is created, ensure that the following BBISSPxx parameters are set as indicated:

  ```
  —MVLOG=YES
  Setting MVLOG to YES indicates that BBI-SS PAS journal messages should be written to a MainView Logger logspace.
  —MVLOGSS=xxxx
  The xxxx variable is the subsystem ID of the logspace to which records will be written. That is, the subsystem ID of the product-specific logspace you have created.
  ```
Sample LOGGERxx PARMLIB member

The following example shows a valid definition for your product-specific logspace:

```
*-------------------------------------------
MVSSYSTEM=SJSC **LPAR**
* Description- mvlogger for energizer ipr
*
LOGSSID=L40T **New SSID**
LOGSPACE=IPR **Must be “IPR”**
DEFAULTLOGGER=NO **Must be “NO”**
WTOLOG=YES
COMPRESSION=YES
MULTIREC=NO
DUMPCOUNTMAX=1
REGISTERDSN=MVIQA.MV440A.P40T.REG01
MAXPERSEC=0
DIVDSN=MVIQA.MV440A.P40T.DIV01
DIVBLOCKS=180
LOGPREFIX=MVIQA.MV440A.P40T.LOG01
STORAGECLASS=
MANAGEMENTCLASS=
LOGCYLINDERS=50
```

Implementation methods

You can implement your product-specific logspace by adding it to the MainView Logger started task or by creating a separate started task.

Adding to the MainView Logger started task

Include the product-specific logspace in the MainView Logger started task by adding the xx in the LOGGERxx parameter member name to the SYSP parameter in the JCL.

The following JCL is an example:

```
//I10TLOGR PROC SYSP=(00,01),
//                    NEWREG=
//                    RESET=
//                    RESETA=
```

This example uses both the LOGGER00 member (default logger) and the new LOGGER01 member in the UBBPARM library specified in the MainView Logger JCL.

Creating a separate started task

Create a separate started task for your product-specific logspace, using the xx in the LOGGERxx parameter member for your logspace in the JCL.
The following JCL is an example:

```
//I10TLOGR PROC SYSP=(01),
//             NEWREG=,
//             RESET=,
//             RESETA=
```

This example uses only the LOGGER01 member in the UBBPARM library specified in the MainView Logger JCL. Even when you start a separate MainView Logger for your logspace, MainView for IMS Online will go through primary MainView Logger logspace to acquire the information from the logs in this separate address space.

## Find messages in MainView Logger

You can use the MainView Logger LOGMSG and LOGJRNL views to display the IMS information. These views, however, display large numbers of messages at one time.

You can set up profile criteria that limits the LOGMSG and LOGJRNL views to just those messages in which you are interested by specifying message criteria.

To view all the IMS messages in which you are interested, you have to customize both the LOGMSG and LOGJRNL views.

- The LOGJRNL view displays the same BBI-SS PAS information as the BBI2 log journal.
- The LOGMSG view displays JES log messages.

How to display logged messages and specify customization criteria is explained in the *MainView Administration Guide*.

## Dynamic install and uninstall

If you use the dynamic install function to install the products, you do not need to modify the standard IMS JCL.

The product data sets are not required in STEPLIB, and the IMFPARM DD and BMCPWD DD statements are not necessary. However, the recommended procedure is to use the standard JCL method to install the products, so that MainView for IMS, MainView for DBCTL, and MainView AutoOPERATOR for IMS are always available when IMS is initialized. Then you can use the dynamic uninstall and install functions to implement maintenance for the product while IMS is still running.
If the product backs itself out because of a software failure, you can use the dynamic uninstall and install functions to reinstall the product (after applying corrective maintenance, if appropriate) without having to stop and restart IMS.

If you plan to continue using the IMS control region JCL method and you want to use dynamic uninstall and install to apply maintenance, follow these steps.

**To use dynamic uninstall and install to apply maintenance**

1. Run the ICOPYAOI job, which is provided in the BBSAMP data set. The job copies members DFSAOE00 and DFSAOUE0 from your target library into a separate data set.

2. Change the IMS control region JCL:
   a. Replace the BBLINK data set in the STEPLIB concatenation with the new data set created by ICOPYAOI.
   b. Add an IMFLOAD DD pointing to the BBLINK data set.

DFSAOE00 and DFSAOUE0 will be loaded from STEPLIB. The JCL install process will load the product modules from IMFLOAD. IMFLOAD will be de-allocated after installation completes so that you can apply maintenance while IMS is up. After maintenance is applied, you can then uninstall and reinstall the product to incorporate the new maintenance without stopping and restarting IMS.

**Dynamic install JCL**

Use the sample dynamic install JCL to install MainView for IMS, MainView for DBCTL, or MainView AutoOPERATOR for IMS in an active IMS system.

You can use MainView Customization to add the appropriate data sets to the JCL, or you can add the data sets manually. The JCL is provided in BBSAMP member IBINSTL.

If the dynamic install is successful, the following messages are issued:

- From the IMS control region

```plaintext
IB1133I INSTALL OF MainView FOR IMS AND AUTOOPERATOR FOR IMS STARTED
  --IMFLOAD DATASETS--
  dsname.BBLINK
  --IMFPARM DATASETS--
  dsname.UBBPARM
  --BMCPSWD DATASETS--
  dsname.BMCPSWD
IB1129I OLT AOE EXIT DISPATCHING ENABLED
IM0202I MAINVIEW FOR IMS DD FUNCTION ENABLED FOR FF AND FP I9X
IM0100I MVIMS COMPONENT VERSION x.x.xx ACTIVE I8X
IM0109I CICS=YES IN EFFECT I9X
```
From the dynamic install batch job:

```
IB1140I ECAO INSTALL COMPLETED SUCCESSFULLY
```

If the install fails, check the messages issued by the batch job to determine what went wrong. Related messages could also be written in the IMS control region. If you are not able to correct the problem indicated by the messages, contact BMC Customer Support.

If the control region issues message IB1136E, try to correct the indicated problem and then try the dynamic install again. If the control region issues message IB1138E, you must run the uninstall job first to clean up the partial install. Then you can try to correct the problem that caused the installation failure and try the install again.

```c
//INSTALL JOB (XXXX),'USER NAME','CLASS=X,MSGCLASS=X,NOTIFY=&SYSUID
/*---------------------------------------------------*/
/* MAINVIEW FOR IMS AND AUTOOPERATOR FOR IMS DYNAMIC INSTALL JOB */
/* THIS JOB CAN BE USED TO INSTALL MAINVIEW FOR IMS AND AUTOOPERATOR */
/* FOR IMS IN AN ACTIVE IMS SYSTEM. */
/* THE STEPLIB, IMFLOAD, AND IMSRESL DATASETS MUST BE APF AUTHORIZED. */
/* PARAMETERS: */
/*   COMP      'ECAO' - INSTALL THE MAINVIEW FOR IMS EVENT COLLECTOR */
/*             AND MAINVIEW AUTOOPERATOR FOR IMS. */
/*             'EC'  - INSTALL THE MAINVIEW FOR IMS EVENT COLLECTOR. */
/*             'AO'  - INSTALL MAINVIEW AUTOOPERATOR FOR IMS. */
/*   IMSID            - IMS TARGET OF THE INSTALL */
/*   INSTLDSN         - THE DATA SET CONTAINING THE INSTALL PROGRAM */
/*                      (IBDYN00). THIS DATA SET WILL BE BOTH THE */
/*                      STEPLIB AND IMFLOAD. */
/*   IMFPARMX         - THE MAINVIEW FOR IMS AND AUTOOPERATOR FOR */
/*                      IMS PARAMETER DATA SETS. YOU MAY HAVE UP TO */
/*                      5 DATA SETS. TO USE ADDITIONAL DATA SETS, */
/*                      UNCOMMENT AND SPECIFY A VALID DATA SET NAME */
/*                      FOR THE IMFPARM2= OR IMFPARM3= PARAMETERS */
/*                      AND UNCOMMENT THE APPROPRIATE LINES AFTER */
/*                      THE IMFPARM DD STATEMENT. IF YOU NEED MORE */
/*                      THAN 3 DATA SETS, MODIFY THE PROCEDURE. */
/*   BMCPSWD          - THE BMC PASSWORD DATA SET */
/*   IMSRESL          - THE IMS RESLIB DATA SET */
/*---------------------------------------------------*/

/DYNINSTL PROC COMP=,
   IMSID=,
   INSTLDSN=,
   IMFPARM1=,
   IMFPARM2=,
   IMFPARM3=,
   BMCPSWD=,
   IMSRESL=,
/*---------------------------------------------------*/

/IBDYN00 EXEC PGM=IBDYN00,
   PARM='INSTALL,&COMP,&IMSID',
   TIME=1439,REGION=1024K
/*---------------------------------------------------*/
/* STEPLIB: DATASET CONTAINING INSTALL PROGRAM (IBDYN00)
*/
```

MainView for IMS and MainView for DBCTL Customization Guide
Dynamic uninstall JCL

The sample dynamic uninstall JCL is used to uninstall MainView for IMS, MainView for DBCTL, or MainView AutoOPERATOR for IMS in an active IMS system.

You can use MainView Customization to add the appropriate data sets to the JCL, or you can add the data sets manually. The JCL is provided in BBSAMP member IBUNINST.

If the dynamic uninstall is successful, the following messages are issued:

- From the IMS control region:
  
  ```
  IE1134I UNINSTALL OF MainView AUTOOPERATOR FOR IMS STARTED
  IE1135I UNINSTALL OF MainView AUTOOPERATOR FOR IMS ENDED
  IB1134I UNINSTALL OF MainView FOR IMS STARTED
  IB1135I UNINSTALL OF MainView FOR IMS ENDED
  ```

- From the dynamic uninstall batch job:

  ```
  IB1143I ECAO UNINSTALL COMPLETED SUCCESSFULLY
  ```

If the uninstall fails, check the messages issued by the batch job to determine what went wrong. Related messages could also be written in the IMS control region. If you
are not able to correct the problem indicated by the messages, contact BMC Customer Support.

//UNINSTAL JOB (XXXX), 'USER NAME', CLASS=X, MSGCLASS=X, NOTIFY=&SYSUID
/// * MainView FOR IMS DYNAMIC UNINSTALL JOB
/// *
/// * THIS JOB CAN BE USED TO UNINSTALL MainView FOR IMS AND/OR
/// * MAINVIEW AUTOOPERATOR FOR IMS FROM AN ACTIVE IMS SYSTEM.
/// *
/// * THE STEPLIB DATASETS MUST BE APF AUTHORIZED.
/// *
/// * PARAMETERS:
/// * COMP     'EC'   -UNINSTALL THE MVIMS EVENT COLLECTOR'
/// *         'AO'   -UNINSTALL MainView AUTOOPERATOR FOR IMS
/// *         'ECAO' -UNINSTALL EC AND AO
/// *         NOTE: WHEN THIS JOB IS SUBMITTED WITH 'ECAO' OR
/// *         'AO' AUTOOPERATOR SHARED VARIABLE QAIMSUN
/// *         WILL BE SET TO 'YES'. WHILE THE VARIABLE
/// *         IS SET TO 'YES', THE QIMS VARIABLES WILL
/// *         NOT BE MAINTAINED.
/// * IMSID         -IMS TARGET OF THE UNINSTALL
/// * UNSTLDSN     -THE DATA SET CONTAINING THE UNINSTALL PROGRAM
/// * (IBDYN00). THIS DATA SET WILL BE STEPLIB
/// *
/// DYNUNSTL PROC COMP=,
///             IMSID=,
///             UNSTLDSN=
/// *
/// IBDYN00 EXEC PGM=IBDYN00,
///             PARM='UNINSTALL,&COMP,&IMSID',
///             TIME=1439, REGION=1024K
/// *
/// STEPLIB: DATASET CONTAINING THE UNINSTALL PROGRAM (IBDYN00)
/// *
/// STEPLIB DD DISP=SHR, DSN=&UNSTLDSN
/// *
/// SYSPRINT DD SYSOUT=*
/// IBDYN00 PEND
/// *
/// EXEDYNAM EXEC DYNUNSTL
/// //
Customize the trace facility

A trace request can specify that 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 MainView Customization.

An alternative is to set up the trace directory manually, as described in this section. Trace log data sets can be pre-allocated or they can be dynamically allocated at the time of the trace request, as described in the MainView for IMS Online User Guide and the MainView for DBCTL User Guide.

**Note**
Customizing the trace facility is applicable only for MainView for IMS Online and MainView for DBCTL.

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

Specify 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
To see the current values of the parameters in the IMFBEX00 member, use the IBEXSUMR view.

## Detail trace data collection

The following 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.

You can use the following parameters in BBPARM member IMFBEX00 to control the maximum length of the I/O area copied into the detail trace for specific call types. To determine the current settings in IMFBEX00, use the IBEXSUMR view.

The default for each parameter is 64 bytes of data, and the maximum length you can specify is 65536. If you specify 0 for a parameter, no I/O area data is captured for that data type.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCIOAL=(nnnnn,mmmmm,ppppp,qqqqq)</td>
<td>Maximum for DC calls</td>
</tr>
<tr>
<td>EMIOAL=(nnnn,mmmm)</td>
<td>Maximum for EMH calls</td>
</tr>
<tr>
<td>FFIOAL=(nnnn,mmmm)</td>
<td>Maximum for full function database calls</td>
</tr>
<tr>
<td>FPIOAL=(nnnn,mmmm)</td>
<td>Maximum for Fast Path database calls</td>
</tr>
</tbody>
</table>

The DCIOAL operands define the maximum length of I/O area to be copied into the detail trace for the following DC call types:

<table>
<thead>
<tr>
<th>DCIOAL operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnnn</td>
<td>Maximum length for IOPCB GET message calls</td>
</tr>
<tr>
<td>mmmmmm</td>
<td>Maximum length for IOPCB GET scratchpad area (SPA) calls</td>
</tr>
<tr>
<td>ppppp</td>
<td>Maximum length for IOPCB other calls</td>
</tr>
<tr>
<td>qqqqq</td>
<td>Maximum length for all alternate IOPCB calls</td>
</tr>
</tbody>
</table>
The EMIOAL operands define the maximum length of I/O area to be copied into the detail trace for the following EMH call types:

<table>
<thead>
<tr>
<th>EMIOAL operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnnn</td>
<td>Maximum length for IOPCB GET calls</td>
</tr>
<tr>
<td>mmmmmm</td>
<td>Maximum length for IOPCB GET other calls</td>
</tr>
</tbody>
</table>

The FFIOAL operands define the maximum length of I/O area to be copied into the detail trace for the following full function database call types:

<table>
<thead>
<tr>
<th>FFIOAL operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnnn</td>
<td>Maximum length for database GET calls</td>
</tr>
<tr>
<td>mmmmmm</td>
<td>Maximum length for other database calls</td>
</tr>
</tbody>
</table>

The FPIOAL operands define the maximum length of I/O area to be copied into the detail trace for the following Fast Path database call types:

<table>
<thead>
<tr>
<th>FPIOAL operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnnn</td>
<td>Maximum length for Fast Path GET calls</td>
</tr>
<tr>
<td>mmmmmm</td>
<td>Maximum length for other Fast Path calls</td>
</tr>
</tbody>
</table>

**Trace display buffer size**

The STORAGE 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. This value overrides any value defined in BBIISPxx.

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

```
STORAGE ==> 100K
```
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 \(20 \times 52 + \) summary trace bytes of storage.

**Trace duration**

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

The value is specified 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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRPREFIX</td>
<td>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.</td>
</tr>
<tr>
<td>TRREUSE</td>
<td>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.</td>
</tr>
</tbody>
</table>

Specify trace defaults in IMFBEX00
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRVOLS=(x,x...)</td>
<td>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.</td>
</tr>
<tr>
<td>TRCYL</td>
<td>Defines the primary allocation default in cylinders (CYLS option) for trace log data sets. The default value is 3.</td>
</tr>
<tr>
<td>TRSUFFIX</td>
<td>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.</td>
</tr>
<tr>
<td>TRSMSSCL</td>
<td>Defines the default name of the SMS storage class for trace log data set allocation. There is no default value.</td>
</tr>
<tr>
<td>TRSMSDCL</td>
<td>Defines the default name of the SMS data class for trace log data set allocation. There is no default value.</td>
</tr>
<tr>
<td>TRSMSMCL</td>
<td>Defines the default name of the SMS management class for trace log data set allocation. There is no default value.</td>
</tr>
</tbody>
</table>

**Update 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*.

**Specify detail trace buffers**

The following keyword parameters define the default for the TRBUFF and TRSIZE data collection buffers:
**Parameter** | **Description**
--- | ---
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 2. (This value overrides any value defined in BBIISPxx.)
TRSFSIZE=nnn | Specifies the size of each buffer in bytes. You can specify the value as nnn or as nnK. The number is rounded to a multiple of 1K. (This value overrides any value defined in BBIISPxx.)

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

**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 data space buffers. The number and size of the detail trace buffers can be specified in either the IMFBEX00 or the BBIISPxx member of the BBPARM data set. IMFBEX00 has priority over BBIISPxx. The defaults are:

TRBUFF=10
TRSFSIZE=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 allocated from a data space 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 storage is released from the data space when the last detail trace stops (and the program running in each region that has a detail trace buffer allocated terminates). The data space is deleted when the target system terminates.

**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.
Set up and maintain a trace directory

You can bypass this step if you used MainView Customization.

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 BBIISPxx.

**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 MainView Security Reference Manual).

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**

1. Edit the BBSAMP sample member JXT001.
2. Add your job statement.
3. Update the symbolics as necessary.
4. Submit the job.

Identify the trace directory to BBI

To identify the trace directory to BBI, use BBPARM member BBISSP00 and specify:

```
TRDIR=d sn
```

The variable `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 might 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
Because 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

1. Edit BBSAMP member JXT003.
2. Add your job statement.
3. Update the symbolics as necessary.
4. Specify one of the following processing options for PARM. If PARM is not specified, no action is taken.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| Blank   | Marks an uncataloged entry 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). |
<p>| LISTALL | Lists all entries |
| NOLIST  | Does not list changed entries |
| PURGE   | Deletes any data sets in the directory that are invalid trace data sets |</p>
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOPURGE</td>
<td>Does not delete invalid data sets (marked as INV (invalid) in the directory)</td>
</tr>
<tr>
<td>VERIFY</td>
<td>Verifies each of the entries in the trace directory</td>
</tr>
<tr>
<td>NOVERIFY</td>
<td>Does not verify entries in the trace directory</td>
</tr>
<tr>
<td>WRITE</td>
<td>Updates trace directory with status changes</td>
</tr>
<tr>
<td>NOWRITE</td>
<td>Does not update trace directory with changes detected</td>
</tr>
</tbody>
</table>

5 Submit the job.

Manage 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.

Define 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 IMS Online User Guide* or the *MainView for DBCTL User Guide*.

Archive 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.

For more information about the IMFTARC sample job, see the *MainView for IMS Online User Guide* or the *MainView for DBCTL User Guide*.

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

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

*Note*
You can also 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 IMS Online User Guide* or the *MainView for DBCTL User Guide*.

Create 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.

For a description of how to use IMFLOGTR, see the *MainView for IMS Online User Guide* or the *MainView for DBCTL User Guide*.

*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.

Print a trace log data set

You can print a trace log data set 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 information about printing history traces in the *MainView for IMS Online User Guide* or the *MainView for DBCTL User Guide*.

**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 Customization Reference.

To print from the online application, take the following steps:

1. Enter **HT** on the COMMAND line and press **Enter**.

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 IMS Online User Guide or the MainView for DBCTL User Guide.

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.

---

## Set up workload history traces

The BBIISP.xx 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 BBIISP.xx (*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 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 third 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 IMS Online User Guide* or the *MainView for DBCTL User Guide*. Setting defaults for all traces, such as the volumes to be used for allocation, is “Specify trace defaults in IMFBEX00” on page 41.

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 the operating system 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 the operating system 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.

---

### Set limits for transaction trace views

The parameters shown in the following table are used in BBPARM member IMFBEX00 to set limits for transaction trace (ITA*) views.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Valid values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRMAXRD</td>
<td>1 to 99999999; default is 5000</td>
<td>Maximum number of trace transaction records read before query terminates</td>
</tr>
<tr>
<td>TRMAXWR</td>
<td>1 to 50000; default is 1000</td>
<td>Maximum number of trace transaction records written before query terminates</td>
</tr>
</tbody>
</table>

To determine the current limit, use the IBEXSUMR view.

To activate changes you make, perform one of the following tasks:
To immediately reset the limits without performing a reset or restart, on the COMMAND line of any view:

1. Execute the SETOPTS command.
2. Type a value in the **Trace max reads** field and press **END**.
3. Execute the SETOPTS command.
4. Type a value in the **Trace max writes** field and press **END**.

Specify RESET PARM IMFBEX00.

Restart the BBI-SS PAS.
Set limits for transaction trace views
Control thresholds, data collection, and record processing

You use parameters in the BBPARM member IMFBEX00 to:

- Control workload thresholds for IMS and IMS Sysplex activity (IMS* and DBC*) views

- *(MainView for IMS Online only)* Control the collection and refresh rates of shared message queue (SMQ) data in views that show message statistics

- Control the number of resource records that are processed for display in the cross-reference views (IX*)

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

To see the current values of the parameters in the IMFBEX00 member, use the IBEXSUMR view.

Note

Shared message queue information only applies to MainView *for IMS* Online and MainView for DBCTL.

Specify thresholds for IMS and IMS sysplex activity views

The parameters shown in the following table are used in BBPARM member IMFBEX00 to control workload thresholds for activity views:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Valid values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXCSINP</td>
<td>0 to 99.99; the default is 0.2</td>
<td>Input queue time in seconds. EXCSINP specifies a threshold for excessive transaction input queue time.</td>
</tr>
<tr>
<td>EXCSELP</td>
<td>0 to 99.99; the default is 0.5</td>
<td>Elapsed time in seconds. EXCSELP specifies a threshold for excessive transaction elapsed time.</td>
</tr>
<tr>
<td>EXCSOCC</td>
<td>0 to 100; the default is 75</td>
<td>Region occupancy percentage. EXCSOCC specifies a threshold for excessive region occupancy time.</td>
</tr>
<tr>
<td>EXCSCPU</td>
<td>0 to 100; the default is 1</td>
<td>Region CPU time percentage. EXCSCPU specifies a threshold for excessive region CPU time.</td>
</tr>
<tr>
<td>EXCSDLI</td>
<td>0 to 99999; the default is 50</td>
<td>Region DL/I calls count. EXCSDLI specifies a threshold for excessive DL/I calls.</td>
</tr>
<tr>
<td>EXCSSQL</td>
<td>0 to 99999; the default is 50</td>
<td>SQL call count. EXCSSQL specifies a threshold for excessive SQL calls.</td>
</tr>
</tbody>
</table>

**Control shared message queue data collection**

Several of the workflow, CQS information, and terminal and user views can display shared queue statistics for IMS systems that use shared message queues.

For information about managing shared message queues and the Shared Queues Data Server, see the *MainView for IMS Online User Guide*.

*Note*

Shared message queue information only applies to MainView for IMS Online.

**Control data query time out**

The ISQTIME parameter in BBPARM member IMFBEX00 defines the amount of time that will pass before a CQS data query times out.
When a query times out, the Shared Queues Data Server is turned off and CQS data is no longer collected.

The default time-out value is 45 seconds, and the range of acceptable values is from 45 to 300. If you experience frequent time-out conditions, you might want to increase the ISQTIME value.

Control refresh frequency

The ISQQUERY parameter in BBPARM member IMFBEX00 is used to control if and how frequently CQS is queried to refresh the shared queues data tables.

You can use the parameter to control the amount of resources used to collect shared message queues data.

Views that use the shared queues data tables will display static values in fields that display message count and message age data until expiration of the refresh period defined with the ISQQUERY parameter.

The following statement is a representation of the ISQQUERY format:

\[ ISQQUERY=(x,y,z) \]

- The \( x \) value controls the refresh rate of message age data in the CQS information views (the ISQ* views). For more information, see “CQS information views with message age fields” on page 57.

- The \( y \) value controls the refresh rate of message count data for shared message queues in non-workflow views. For more information, see “Non-workflow views with message count fields” on page 58.

- The \( z \) value controls the refresh rate of message count fields for shared message queues in workflow views. For more information, see “Workflow views with message count fields” on page 59.

CQS information views with message age fields

The \( x \) value in ISQQUERY\((x,y,z)\) applies to CQS information views (ISQ*) that show message age data.

The \( x \) value is used to:

- Specify whether or not message age data will be collected and displayed for shared message queues
Define the refresh rate (in seconds) for the data in message age fields if such data is collected

If a refresh rate is specified for the message data, no matter how frequently a CQS view is refreshed, message ages that are reported in the view will not be refreshed until the specified refresh period expires.

The x value can be 0 or from 60 through 7200. The default is 60.

A value of zero in the x position prevents the collection of message age data for display in the CQS views. When the age data is not being collected, the letters SMQ are displayed in message age fields, rather than message age data.

**Note**
Because frequent refreshing of message age data can consume a significant amount of processing time and can delay IMS and other CQS activity, if you allow collection of the data, you should try to set the x value reasonably high. If you are concerned only with messages that are older than an hour, for example, you should consider setting the value as high as 3600.

---

**Non-workflow views with message count fields**

The y value in ISQUERY=(x,y,z) applies to non-workflow views that show message count data, such as the CQS information (ISQ*), transaction (ITR*), and user/terminal information (IU*) views. The data in the views is collected on demand in real time.

The y value is used to:

- Specify whether or not message count data will be collected and displayed for shared message queues
- Define the refresh rate (in seconds) for the data in message count fields if such data is collected

If a refresh rate is specified for the message data, no matter how frequently a view is refreshed, message counts that are reported in the view will not be refreshed until the specified refresh period expires.

The y value can be 0, 5, 10 or 15. The default is 5.

A value of zero in the y position prevents the collection of message count data for display in non-workflow views. When the message count data is not being collected, the letters SMQ are displayed in message count fields, rather than message count data.
Workflow views with message count fields

The $z$ value in ISQQUERY=$(x,y,z)$ applies only to workload views. The data for workload views is collected continuously by the workflow sampler, which provides realtime and historical data.

The $z$ value is used to:

- Specify whether or not message count data will be collected and displayed for shared message queues
- Define the refresh rate (in seconds) for the data in message count fields if such data is collected

If a refresh rate is specified for the message data, no matter how frequently a view is refreshed, message counts that are reported in the view will not be refreshed until the specified refresh period expires.

The $z$ value can be 0 or 5. The default is 5.

A value of zero in the $z$ position prevents the collection of message count data for display in workflow views. When the message count data is not being collected, the letters SMQ are displayed in message count fields, rather than message count data.

Control resource records processed for cross-reference views

You can use the XREFLIM parameter in BBPARM member IMFBEX00 to control the number of resource records that are processed for display in the cross-reference views. Setting an appropriate record limit with XREFLIM can prevent excessive CPU and storage usage.

The recommended XREFLIM value is 100000. The minimum value is 5000. There is no upper limit to the parameter value, but an overly large value might use excessive resources in the PAS. (If the parameter is omitted, no limit is imposed.)

When the XREFLIM processing limit is reached, processing stops before all possible records are displayed, and the following message is displayed:

Cross-reference processing limit (nnn) reached

To determine the current limit, use the IBEXSUMR view. To immediately reset the limit without performing a reset or restart, use the SETOPTS primary command on any view and specify a limit in the Cross-reference limit field.
Event Collector usage

The Event Collector is a MainView for IMS product component that collects and records IMS event data. The collected data is used as input to the MainView for IMS Online product and the MainView for IMS Offline product components.

Event Collector overview

The Event Collector collects IMS event data and writes it to the IMS log as user record logs X'FA' (transaction records) and X'F9' (program records).

Data collected by the Event Collector is:

- Used by the online Workload Monitor and Workload Analyzer components for workload wait and trace services
- Processed by Log Edit to create the IMS resource utilization file (IRUF), which is the primary file used by the MainView for IMS Offline product
- Used by user-developed programs or other offline processing systems

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

Control the Event Collector

The data collection parameters specified in BBPARM member IMFECPO00 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 provide a method for you to limit the data collected by the Event Collector and thereby limit the resources used. The IMFECPO0 member documents the default options.

**Event Collector CPU usage**

The data collection parameters have an effect 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 MainView for IMS data, make it impossible to summarize all these variables into standard CPU usage estimates or option setting recommendations.

The description of the parameters discuss 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 + MainView for IMS usage is 220, as shown in “Data collection example 1” on page 62.

---

**Data collection example 1**

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 this example).

**Example**

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS</td>
<td>200</td>
</tr>
<tr>
<td>Event Collector = 10% of IMS</td>
<td>20</td>
</tr>
<tr>
<td>Total = IMS + Event Collector</td>
<td>220</td>
</tr>
</tbody>
</table>

Thus, for this 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 “Data collection example 2” on page 63.
Data collection example 2

**Example**

IMS = 200

Event Collector = 10% of IMS = \(200 \times 0.10 = 20\)

Total = IMS + Event Collector + BFALTERS = 200 + 20 + 8 = 228

Start and stop the Event Collector

The Event Collector initializes immediately after IMS initialization is completed and it stays active until IMS terminates.

The Event Collector can also be dynamically installed and uninstalled, a function that is typically used to implement the MainView for IMS Online product maintenance while IMS is still up and running (see “Dynamic install and uninstall” on page 36).

Data collection parameters

The following Event Collector parameters determine what IMS event data is collected in the IMS log file:

- BHTO controls whether IMS buffer handler activity is included with DL/I CPU or timed separately
- BILLOVHD determines how dependent region CPU time is treated
- BMP controls whether activity data for BMP and JBP transactions and programs is collected
- CICS controls whether records are collected for DBCTL CICS and ODBA thread data
- CPU controls the level of CPU data collected
- DBIO controls the level of database I/O data to be collected
- DEPREC controls whether recovery from additional abend conditions in dependent regions is enabled and performed as necessary
CPU timing parameters

The Event Collector accumulates CPU times in various categories and maintains several CPU fields in the MainView for IMS log records and IRUF records. The values in these fields, or various combinations of the values, are reported.

The following topics discuss the Event Collector parameters that control what CPU time information is collected:

“Application program CPU” on page 64
“Message DL/I CPU” on page 65
“Control DL/I CPU” on page 65
“DB2 CPU” on page 66
“Message buffer CPU” on page 66
“Control buffer CPU” on page 66
“Message OPEN/CLOSE CPU” on page 67
“Control OPEN/CLOSE CPU” on page 67
“Program scheduling CPU” on page 67
“Message region overhead CPU” on page 68
“Control region overhead” on page 68

Application program CPU

Application program CPU, also called message region CPU, is collected unless CPU=NONE.

This value is the time spent by the application program in the dependent region.

Note

This field includes user-attributable CPU time incurred in DB2 through the IMS Attach Facility if the parameter FEATURE=NODB2 is specified in PARMLIB member IMFSYS00 (see PARMLIB member IMFSYSBB for more information).
Application program CPU is included in chargeable CPU, which is CPU time that is directly attributable to the user who submitted the transaction.

The BILLOVHD parameter can affect this value by optionally adding some dependent region overhead.

**Message DL/I CPU**

Message DL/I CPU is the time spent in the dependent region processing DL/I requests.

This time value usually includes most of the time involved in processing database calls.

This value is collected unless CPU=NONE. The time is included in chargeable CPU, which is CPU time that is directly attributable to the user who submitted the transaction.

LSO=Y and BMPs and JBP in nonparallel DL/I mode reduce this value (moving time to control DL/I CPU).

**Control DL/I CPU**

Control DL/I CPU is the time spent in the control region (or in DLISAS if LSO=S) processing DL/I requests.

The major portion of this time is for message queue calls.

If LSO=Y, most database DL/I work is done under the LSO subtasks in the control region. If LSO=S, the serialized database processing occurs in the DLISAS region but is accumulated with the control region CPU time.

The control DL/I CPU value is collected only if CPU=ALL. The value is included in chargeable CPU, which is CPU time that is directly attributable to the user who submitted the transaction.

If CPU=DEP, total DL/I CPU is generally 5 to 15 percent less than with the CPU=ALL option, because control DL/I CPU is not measured. If LSO=Y or BMP and JBP nonparallel processing is used, much more data is lost.
DB2 CPU

DB2 CPU is the amount of dependent region CPU time (in thousandths of a second) that is used by the transaction to make DB2 requests.

The request generally runs in cross-memory mode under the IMS dependent region program controller task.

The DB2 CPU value is collected only if CPU=ALL, CPU=DEP, or CPU=DEPDB2. The value is included in chargeable CPU, which is CPU time that is directly attributable to the user who submitted the transaction.

Message buffer CPU

Message buffer CPU is the time spent in the IMS database buffer handler routines during database DL/I call processing.

Message buffer CPU is collected under the dependent region task.

The message buffer CPU value is collected if BHTO=ON and if CPU=ALL or CPU=DEP. BHTO is forced to OFF unless DBIO=BFALTERS.

If BHTO=OFF (the default), this time is zero and the CPU is included in message DL/I CPU. Message buffer CPU time can be timed separately to exclude it from chargeable CPU, because the time can be considered an overhead function within IMS.

LSO=Y and BMPs and JBPs in nonparallel DL/I mode reduce this value (moving time to control buffer CPU).

Control buffer CPU

Control buffer CPU is the time spent in the IMS database buffer handler routines during database DL/I call processing.

Control buffer CPU is collected in the control or DLISAS regions.

The control buffer CPU value is collected if BHTO=ON and CPU=ALL. BHTO is forced to OFF unless DBIO=BFALTERS.

If BHTO=OFF (the default), this time is zero and the CPU is included in control DL/I CPU. The time can be timed separately to exclude it from chargeable CPU, because the time can be considered an overhead function within IMS.
LSO=Y and BMPs and JBPs in nonparallel DL/I mode increase this value.

**Message OPEN/CLOSE CPU**

IMS DL/I CPU time spent in database OPEN/CLOSE activity under the dependent region TCB is for Fast Path databases only.

Full function database OPEN/CLOSE activity is performed in the control region.

MainView *for IMS* treats DL/I CPU time as overhead CPU.

The DL/I CPU time value is collected unless CPU=NONE.

**Control OPEN/CLOSE CPU**

All full function database OPEN/CLOSE activity is performed in the control region under the control task TCB.

This activity includes all processing in the IMS OPEN/CLOSE module DFSDLOC0. If LSO=S, the processing occurs in the DLISAS address space but is accumulated in this field.

MainView *for IMS* treats control OPEN/CLOSE CPU time as overhead CPU.

This value is only collected if CPU=ALL.

**Program scheduling CPU**

Program scheduling activity occurs in the control region and, if LSO=S, partially in the DLISAS region.

The program scheduling activity in both regions are accumulated in this field.

MainView *for IMS* treats program scheduling CPU time as overhead CPU.

This value is only collected if CPU=ALL.
Message region overhead CPU

The message region overhead CPU value is always collected.

This value is the amount of overhead CPU time (both TCB and SRB) that was spent in the dependent region and that was not directly attributable to a transaction. The value usually includes program initialization and termination.

- If the startup parameter BILL OVHD=NO (the default), the value includes the time between the end of scheduling and the first program DL/I call.

- If BILLOVHD=YES, this time is included in application program CPU (dependent region). Program load time is attributed to the first transaction processed and is chargeable.

- If CPU=NONE, the value includes all CPU time incurred in the dependent regions. For all the other CPU options, the value includes all dependent region CPU not identified as application program, DL/I, buffer, or OPEN/CLOSE CPU (for example, region startup).

**Note**
Use of the parameter CPUOVHD=REFCPU sets this value to zero.

Control region overhead

The control region overhead value is always collected:

- If CPU=ALL, this value includes all control/DLISAS CPU not identified as DL/I, buffer, program scheduling or OPEN/CLOSE CPU (control region initialization, for example).

- For all the other CPU options, the value includes all CPU time incurred in the control/DLISAS regions.

**Note**
Use of the parameter CPUOVHD=REFCPU sets this value to zero.

The control region overhead CPU time for a program includes the nonattributable control region overhead CPU time (both TCB and SRB) measured between the last program termination (in any IMS region) and the termination of this one. The program accounting record (PAR) has the following control region overhead fields:

- The first **PAR** field for control region overhead contains the nonattributable overhead field and several other overhead fields, accumulated for a program. This
value includes scheduling CPU time from this program record and the three control region CPU times from the corresponding transaction records.

- The second PAR field includes only the nonattributable overhead figure. If you are writing a program to accumulate total CPU time for any time period, use this field.

**Tip**

When you review CPU time, make comparisons carefully. If the comparisons are against address space statistics, be sure to include all the CPU times for that region and no others.

The jobname of the dependent region (region ID) where processing took place is available in both the TAR and PAR records.

### BMP and JBP data

BMP and JBP data is always collected unless BMP=NO or BMP=NOCPU is specified. If BMP=NO, no BMP and JBP transaction and program records are produced, which affects all MainView for IMS reports. If BMP=NOCPU, all CPU timing fields in the BMP and JBP records are zero. DL/I calls and database I/O statistics are collected.

For more information about the BMP Event Collector parameter, see the *MainView for IMS and MainView for DBCTL Customization Guide*.

### Parameter option sets

The product supplies several predefined 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=DEP
- DTSQLCPU=YES
- DBFPNLVL=3
- DBIO=IOWAITS
- BHTC=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 MainView for IMS 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 MainView for IMS CPU usage.

The standard options, either as distributed or with the variations mentioned, generally result in MainView for IMS 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 MainView for IMS CPU usage.

- **BMP=YES**
- **CPU=ALL**
- **DTSQLCPU=YES**
- **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 DEP option for the ALL option to conserve CPU usage. Selecting the DEP option is likely to be a one-time decision, although you might decide to use CPU=ALL occasionally for performance analysis.

The DTSQLCPU parameter has no affect unless detail trace is active. To reduce transaction CPU time and elapsed time, set DTSQLCPU to NO; however, you will lose individual SQL CPU timing information, but will retain total CPU timing information.

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 MainView for IMS CPU usage of 15 to 25 percent of total IMS CPU, though some individual BMPs or JBP's 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 MainView for IMS CPU usage.

<table>
<thead>
<tr>
<th>Option</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMP</td>
<td>NO (optional)</td>
</tr>
<tr>
<td>CPU</td>
<td>NONE</td>
</tr>
<tr>
<td>DTSQLCPU</td>
<td>NO</td>
</tr>
<tr>
<td>DBFP</td>
<td>YES</td>
</tr>
<tr>
<td>DBFPLVL</td>
<td>0</td>
</tr>
<tr>
<td>DBIO</td>
<td>NONE</td>
</tr>
<tr>
<td>BHTO</td>
<td>OFF</td>
</tr>
<tr>
<td>DEPREC</td>
<td>NO</td>
</tr>
<tr>
<td>CICS</td>
<td>NO</td>
</tr>
</tbody>
</table>

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 MainView for IMS are still created for each transaction and program. Data collected includes:

- All elapsed timings and storage usage
- DL/I call counts per call type
- DL/I call counts per database
- DL/I call counts per LTERM

The minimum option set generally results in MainView for IMS 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.

<table>
<thead>
<tr>
<th>Option</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMP/JBP trace</td>
<td>If BMP=NO, BMP and JBP transactions and programs cannot be traced.</td>
</tr>
<tr>
<td>LTRAC</td>
<td>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.</td>
</tr>
<tr>
<td>STRAC</td>
<td>If CPU=NONE, all CPU data is zero.</td>
</tr>
<tr>
<td>DTRAC</td>
<td>If DBFPLVL=0, no DEDB I/O read information is available for display in detail traces.</td>
</tr>
<tr>
<td>#I/O, AVG #I/O</td>
<td>If DBIO=NONE, the values in these fields are zero.</td>
</tr>
<tr>
<td>DTRAC DL/I CPU times</td>
<td>If CPU=DEPPGM, CPU=DEPDB2, or CPU=NONE, the values in these fields are zero.</td>
</tr>
<tr>
<td>DTRAC DB2 CPU times</td>
<td>If CPU=DEPPGM or CPU=NONE, the values in these fields are zero.</td>
</tr>
<tr>
<td>DBCTL thread tracing</td>
<td>If CICS=NO or CICS=OFFLINE, no trace data is collected for DBCTL threads.</td>
</tr>
<tr>
<td>DBCTL thread monitoring</td>
<td>If CICS=NO or CICS=OFFLINE, no data is collected for the Workload Monitor services.</td>
</tr>
</tbody>
</table>

**When changes to Event Collector parameters take affect**

If you change an Event Collector parameter in BBPARM member IMFECPOO, the change does not take effect until IMS restarts or an uninstall and install are done.

The Event Collector can be dynamically installed and uninstalled, a function that is typically used to implement the MainView for IMSOnline product maintenance while IMS is still up and running (see “Dynamic install and uninstall” on page 36.
Event Collector parameters

The Event Collector parameters are specified in BBPARM member IMFECP00. The parameters specify the data to be collected, the functions to be performed during error recovery, and the MainView for IMS product diagnostics to use. The parameters are read and processed by MainView for IMS at IMS initialization and remain in effect throughout the IMS session.

**Note**

Some of the available collection, recovery, and diagnostic parameters can be CPU-intensive and careful consideration should be given to their selection.

To see the current values of the parameters in the IMFECP00 member, use the IECPSUMR view.

Table 2 on page 73 lists the Event Collector parameters. Table 3 on page 76 lists the parameters by the section in which they appear in the IMFECP00 member.

### Table 2: Event Collector parameters alphabetically

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Brief description</th>
<th>Detailed information</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABCOUNT</td>
<td>The parameter defines how many times the Event Collector will try to recover from an abend before it shuts down.</td>
<td>“ABCOUNT parameter” on page 78</td>
</tr>
<tr>
<td>BACKOUT</td>
<td>The parameter defines the action that will be taken if the Event Collector encounters severe errors.</td>
<td>“BACKOUT parameter” on page 78</td>
</tr>
<tr>
<td>BHTO</td>
<td>The parameter controls whether IMS buffer handler activity is included with DL/I CPU time or if it is timed separately.</td>
<td>“BHTO parameter (buffer handler timing)” on page 78</td>
</tr>
<tr>
<td>BILLOVHD</td>
<td>The parameter determines whether dependent region CPU time is treated as overhead or as chargeable CPU time per user when the CPU time is spent in prior transaction termination, current transaction scheduling, program load, or schedule-to-first DL/I.</td>
<td>“BILLOVHD parameter (dependent region overhead CPU time)” on page 79</td>
</tr>
<tr>
<td>BMP</td>
<td>The parameter controls whether activity data for BMP and JBP transactions and programs is collected.</td>
<td>“BMP parameter (BMP and JBP data)” on page 80</td>
</tr>
<tr>
<td>BMPTTRQC</td>
<td>The parameter determines whether a transaction record is created for each BMP or JBP program MESSAGE GET UNIQUE.</td>
<td>“BMPTTRQC parameter” on page 81</td>
</tr>
<tr>
<td>CICS</td>
<td>The parameter controls whether records are collected for DBCTL thread data (which includes CICS and ODBA threads).</td>
<td>“CICS parameter (DBCTL threads)” on page 82</td>
</tr>
<tr>
<td>Parameter</td>
<td>Brief description</td>
<td>Detailed information</td>
</tr>
<tr>
<td>------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>CORE</td>
<td>The parameter determines whether the amount of storage allocated and used by a dependent region is calculated and recorded.</td>
<td>“CORE parameter” on page 83</td>
</tr>
<tr>
<td>CPIC1TRN</td>
<td>The parameter handles a special case in which two consecutive sync points occur without any intervening work.</td>
<td>“CPIC1TRN parameter” on page 83</td>
</tr>
<tr>
<td>CPICDB2</td>
<td>The parameter determines when a transaction record is written for CPI-C (explicit APPC) conversations.</td>
<td>“CPICDB2 parameter” on page 84</td>
</tr>
<tr>
<td>CPICDLI</td>
<td>The parameter determines when additional transaction records (in conjunction with the CPICDB2 parameter) are written for CPI-C (explicit APPC) conversations.</td>
<td>“CPICDLI parameter” on page 84</td>
</tr>
<tr>
<td>CPU</td>
<td>The parameter controls the level of CPU data collected by the Event Collector in all IMS regions, in just the dependent regions, or in no regions.</td>
<td>“CPU parameter (CPU data collection options)” on page 85</td>
</tr>
<tr>
<td>CPUOVHD</td>
<td>The parameter determines the type of overhead information that is collected.</td>
<td>“CPUOVHD parameter” on page 87</td>
</tr>
<tr>
<td>DBFFLVL</td>
<td>The parameter determines whether to collect full function database activity data.</td>
<td>“DBFFLVL parameter” on page 89</td>
</tr>
<tr>
<td>DBFP</td>
<td>The parameter determines whether to collect counts of NONKEY WRITES and NO I/O ALTERS.</td>
<td>“DBFP parameter” on page 89</td>
</tr>
<tr>
<td>DBFPLVL</td>
<td>The parameter determines whether DEDB activity data is collected and for which types of activity data is collected.</td>
<td>“DBFPLVL parameter” on page 90</td>
</tr>
<tr>
<td>DBIO</td>
<td>The parameter controls the level of database I/O data to be collected by the Event Collector.</td>
<td>“DBIO parameter (database I/O options)” on page 91</td>
</tr>
<tr>
<td>DBTNAME</td>
<td>The parameter determines whether to collect DL/I call counts only or to collect DL/I call counts and database I/O call counts.</td>
<td>“DBTNAME parameter” on page 92</td>
</tr>
<tr>
<td>DBTS</td>
<td>The parameter sets the maximum number of database trailers allowed per non-BMP/JBP region.</td>
<td>“DBTS parameter” on page 93</td>
</tr>
<tr>
<td>DBTS4BMP</td>
<td>The parameter sets the maximum number of database trailers allowed per BMP or JBP region.</td>
<td>“DBTS4BMP parameter” on page 94</td>
</tr>
<tr>
<td>Parameter</td>
<td>Brief description</td>
<td>Detailed information</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>DEPREC</td>
<td>The parameter controls whether recovery from additional abend conditions in dependent regions is enabled and performed as necessary.</td>
<td>“DEPREC parameter (extended recovery)” on page 94</td>
</tr>
<tr>
<td>DTSQLCPU</td>
<td>The parameter controls whether CPU time for individual DB2 SQL events is collected and recorded for detail traces.</td>
<td>“DTSQLCPU parameter” on page 95</td>
</tr>
<tr>
<td>DUMPS</td>
<td>The parameter determines whether a dump should be taken when the Event Collector abnormally terminates.</td>
<td>“DUMPS parameter” on page 95</td>
</tr>
<tr>
<td>ETIMEELP</td>
<td>The parameter determines whether to collect transaction elapsed time.</td>
<td>“ETIMEELP parameter” on page 96</td>
</tr>
<tr>
<td>ETIMEIWT</td>
<td>The parameter determines whether to collect transaction IWAIT time.</td>
<td>“ETIMEIWT parameter” on page 97</td>
</tr>
<tr>
<td>MAXDS</td>
<td>The parameter determines the maximum space in MB allocated to data space.</td>
<td>“MAXDS parameter” on page 98</td>
</tr>
<tr>
<td>MSCCLOCK</td>
<td>The parameter determines whether the Event Collector converts a transaction’s arrival date and time to local time.</td>
<td>“MSCCLOCK parameter” on page 98</td>
</tr>
<tr>
<td>MSGOQBUF</td>
<td>The parameter specifies the maximum number of internal work elements that will be allocated to send output queue information from IMS to the product address space.</td>
<td>“MSGOQBUF parameter” on page 99</td>
</tr>
<tr>
<td>MSGOQTME</td>
<td>The parameter determines whether message output queue time is collected for Fast Path and full function transactions.</td>
<td>“MSGOQTME parameter” on page 100</td>
</tr>
<tr>
<td>MXPASRQ</td>
<td>The parameter determines the number of FA request elements allocated in the data space.</td>
<td>“MXPASRQ parameter” on page 100</td>
</tr>
<tr>
<td>MXPASSQ</td>
<td>The parameter determines the maximum amount of space that can be used in the data space to queue an FA record.</td>
<td>“MXPASSQ parameter” on page 101</td>
</tr>
<tr>
<td>RGNIOPT</td>
<td>The parameter determines whether to abend the IMS dependent region if MainView for IMS initialization fails because of CSA shortage.</td>
<td>“RGNIOPT parameter” on page 101</td>
</tr>
<tr>
<td>SMFID</td>
<td>The parameter determines the ID of the SMF transaction record.</td>
<td>“SMFID parameter” on page 102</td>
</tr>
<tr>
<td>SUBSYS</td>
<td>The parameter determines the subsystem ID for system events.</td>
<td>“SUBSYS parameter” on page 102</td>
</tr>
</tbody>
</table>
## Event Collector parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Brief description</th>
<th>Detailed information</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSID</td>
<td>The parameter identifies the MainView for IMS system for the Performance Reporter and the Transaction Accountant.</td>
<td>“SYSID parameter” on page 102</td>
</tr>
<tr>
<td>TELON</td>
<td>The parameter determines support for the TELON application package.</td>
<td>“TELON parameter” on page 103</td>
</tr>
<tr>
<td>TIMERR</td>
<td>The parameter determines if a region abends if the time values are out of range.</td>
<td>“TIMERR parameter” on page 103</td>
</tr>
<tr>
<td>TRNSYNC</td>
<td>The parameter determines whether to write a transaction record at a BMP or JBP checkpoint.</td>
<td>“TRNSYNC parameter” on page 104</td>
</tr>
<tr>
<td>WLMSC</td>
<td>The parameter determines whether the WLM service class is reported on the TRN and PGM records.</td>
<td>“WLMSC parameter” on page 104</td>
</tr>
<tr>
<td>ZTIME</td>
<td>The parameter determines whether to monitor zIIP and zAAP usage within IMS dependent regions and to record usage amounts in the FA record.</td>
<td>“ZTIME parameter” on page 105</td>
</tr>
</tbody>
</table>

### Table 3: Event Collector parameters by section

<table>
<thead>
<tr>
<th>Section in IMFECPO0</th>
<th>Detailed information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Collection Parameters</td>
<td>“BHTO parameter (buffer handler timing)” on page 78</td>
</tr>
<tr>
<td></td>
<td>“BILLOVHD parameter (dependent region overhead CPU time)” on page 79</td>
</tr>
<tr>
<td></td>
<td>“CORE parameter” on page 83</td>
</tr>
<tr>
<td></td>
<td>“CPU parameter (CPU data collection options)” on page 85</td>
</tr>
<tr>
<td></td>
<td>“CPUOVHD parameter” on page 87</td>
</tr>
<tr>
<td></td>
<td>“DBFFLVL parameter” on page 89</td>
</tr>
<tr>
<td></td>
<td>“DBFP parameter” on page 89</td>
</tr>
<tr>
<td></td>
<td>“DBFPLVL parameter” on page 90</td>
</tr>
<tr>
<td></td>
<td>“DBIO parameter (database I/O options)” on page 91</td>
</tr>
<tr>
<td></td>
<td>“DTSQLCPU parameter” on page 95</td>
</tr>
<tr>
<td></td>
<td>“ETIMEELP parameter” on page 96</td>
</tr>
<tr>
<td></td>
<td>“ETIMEIWT parameter” on page 97</td>
</tr>
<tr>
<td></td>
<td>“MSGOQBUF parameter” on page 99</td>
</tr>
<tr>
<td></td>
<td>“MSGOQTME parameter” on page 100</td>
</tr>
<tr>
<td></td>
<td>“WLMSC parameter” on page 104</td>
</tr>
<tr>
<td>Section in IMFECPO0</td>
<td>Detailed information</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Data Collection Storage Parameters</td>
<td>“MAXDS parameter” on page 98</td>
</tr>
<tr>
<td></td>
<td>“MXPASRQ parameter” on page 100</td>
</tr>
<tr>
<td></td>
<td>“MXPASSQ parameter” on page 101</td>
</tr>
<tr>
<td>Miscellaneous Parameters</td>
<td>“DUMPS parameter” on page 95</td>
</tr>
<tr>
<td></td>
<td>“MSCCLOCK parameter” on page 98</td>
</tr>
<tr>
<td></td>
<td>“SMFID parameter” on page 102</td>
</tr>
<tr>
<td></td>
<td>“SUBSYS parameter” on page 102</td>
</tr>
<tr>
<td></td>
<td>“SYSID parameter” on page 102</td>
</tr>
<tr>
<td></td>
<td>“TELON parameter” on page 103</td>
</tr>
<tr>
<td>Parameters for Controlling Thread Type Monitoring and Frequency and Granularity of FA Records</td>
<td>“BMP parameter (BMP and JBP data)” on page 80</td>
</tr>
<tr>
<td></td>
<td>“BMPTRQC parameter” on page 81</td>
</tr>
<tr>
<td></td>
<td>“CICS parameter (DBCTL threads)” on page 82</td>
</tr>
<tr>
<td></td>
<td>“CPICITRN parameter” on page 83</td>
</tr>
<tr>
<td></td>
<td>“CPICDB2 parameter” on page 84</td>
</tr>
<tr>
<td></td>
<td>“CPICDLI parameter” on page 84</td>
</tr>
<tr>
<td></td>
<td>“DBTNAME parameter” on page 92</td>
</tr>
<tr>
<td></td>
<td>“DBTS parameter” on page 93</td>
</tr>
<tr>
<td></td>
<td>“DBTS4BMP parameter” on page 94</td>
</tr>
<tr>
<td></td>
<td>“TRNSYNC parameter” on page 104</td>
</tr>
<tr>
<td>Parameter for controlling collection of IBM System z Integrated Information Processor (zIIP) and IBM System z Application Assist Processor (zAAP) usage statistics</td>
<td>“ZTIME parameter” on page 105</td>
</tr>
<tr>
<td>Recovery Parameters</td>
<td>“ABCOUNT parameter” on page 78</td>
</tr>
<tr>
<td></td>
<td>“BACKOUT parameter” on page 78</td>
</tr>
<tr>
<td></td>
<td>“DEPREC parameter (extended recovery)” on page 94</td>
</tr>
<tr>
<td></td>
<td>“RGNIOPT parameter” on page 101</td>
</tr>
<tr>
<td></td>
<td>“TIMERR parameter” on page 103</td>
</tr>
</tbody>
</table>
**ABCOUNT parameter**

The parameter defines how many times the Event Collector will try to recover from an abend before it shuts down.

**Syntax**

ABCOUNT=02 | \(nn\)

**Options**

The valid option value is 01 through 99.

The default is 02.

---

**BACKOUT parameter**

The parameter defines the action that will be taken if the Event Collector encounters severe errors.

**Syntax**

BACKOUT=YES | NO

**Options**

Valid options are:

- **YES** *(default)*
  
  The Event Collector is backed out and IMS does not abend.

- **NO**
  
  The Event Collector is not backed out and IMS abends.

---

**BHTO parameter (buffer handler timing)**

The parameter controls whether IMS buffer handler activity is included with DL/I CPU time or if it is timed separately.

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 MainView...
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 MainView for IMS CPU usage by 20 to 40 percent.

**Syntax**

BHTO=OFF|ON

**Options**

Valid options are:

- **OFF (default)**
  The buffer handler time (BHT) data is included with the DL/I CPU time data. The OFF option is forced if the DBIO option is not BFALTERS. DEDB and MSDB BHT for IMS Fast Path is always included in DL/I CPU time data.
  The Performance Reporter and the Transaction Accountant report BHT CPU=0. The Performance Reporter, the Transaction Accountant, and the Workload Analyzer trace services report increased DL/I CPU time (DL/I CPU = DL/I + BHT), which produces increased chargeable CPU time in the Transaction Accountant. Workload Analyzer trace services report increased DL/I CPU time (DL/I CPU = DL/I + BHT).

- **ON**
  The BHT data is collected separately. BHT data is available for the Performance Reporter and the Transaction Accountant.

**BILLOVHD parameter (dependent region overhead CPU time)**

The parameter determines whether dependent region CPU time is treated as overhead or as chargeable CPU time per user when the CPU time is spent in prior transaction termination, current transaction scheduling, program load, or schedule-to-first DL/I.

**Syntax**

BILLOVHD=NO|YES|SCHEDDLI
Options

Valid options are:

- **NO (default)**
  
  Time for prior transaction termination, current transaction scheduling, program load, and schedule-to-first DL/I are treated as dependent region overhead CPU time.

  The NO option results in less chargeable CPU time in the Transaction Accountant.

- **YES**
  
  Time for prior transaction termination, current transaction scheduling, program load, and schedule-to-first DL/I are treated as dependent region chargeable CPU time.

  The YES option might cause variations in the usually stable message program CPU time, especially between the first and subsequent transactions processed in a single program scheduling.

  Overhead information cannot be collected for DBCTL regions.

- **SCHEDDLI**
  
  Time for prior transaction termination and current transaction scheduling are treated as dependent region overhead CPU time. Time for program load and schedule-to-first DL/I are treated as dependent region chargeable CPU time.

  The SCHEDDLI option might cause variations in the usually stable message program CPU time, especially between the first and subsequent transactions processed in a single program scheduling.

  Overhead information cannot be collected for DBCTL regions.

BMP parameter (BMP and JBP data)

The 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 MainView for IMS monitoring adds. However, this option is viable only if the MainView for IMS BMP or JBP data is not required for accounting or IMS performance analysis. In general, you will want to collect BMP and JBP data.

The effect of this parameter on MainView for IMS CPU usage depends on the number and activity of all BMPs and JBPs, but usage is higher when specifying YES rather than NO or NOCPU.
Syntax

BMP=YES|NO|NOCPU

Options

Valid options are:

- **YES** *(default)*
  BMP and JBP transaction and program activity data is collected. The collected data is available for Performance Reporter, Transaction Accountant, and workload trace.

- **NO**
  BMP and JBP transaction and program activity data is not collected. No data is available for Performance Reporter, Transaction Accountant, and workload trace.

- **NOCPU**
  BMP and JBP transaction and program activity data is collected, but no CPU time usage is collected. No CPU data for BMPs or JBPs is available for Performance Reporter, Transaction Accountant, and workload trace. All other data, such as DL/I counts and database accesses, is available for the Performance Reporter, the Transaction Accountant, and workload trace.

**BMPTRQC parameter**

The parameter determines whether a transaction record is created for each BMP or JBP program MESSAGE GET UNIQUE.

Syntax

BMPTRQC=NO|YES

Options

Valid options are:

- **NO**
  No transaction record is written for a MESSAGE GET UNIQUE that has a status code of QC.
YES
A transaction record is written for a MESSAGE GET UNIQUE that has a status code of QC.

CICS parameter (DBCTL threads)

The parameter controls whether records are collected for DBCTL thread data (which includes 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.

Syntax

CICS=YES|ONLINE|OFFLINE|NO

Options

Valid options are:

- **YES (default)**
  Combines online and offline functions. Data is available for offline batch report products, Workload Analyzer wait and trace services, and Workload Monitor services.

- **ONLINE**
  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.

- **OFFLINE**
  Records data for CICS and ODBA threads and writes it to the IMS log for batch report processing. Data is available for Offline batch products, such as Performance Reporter and Transaction Accountant, but not for Workload Analyzer wait or trace services and Workload Monitor services.

- **NO**
  Does not measure CICS and ODBA thread activity data. No data is available for Performance Reporter and Transaction Accountant, Workload Analyzer wait and trace services, and Workload Monitor services.
CORE parameter

The parameter determines whether the amount of storage allocated and used by a dependent region is calculated and recorded.

Syntax

CORE=YES|NO

Options

Valid options are:

■ YES (default)
  The amount of storage allocated and used by a dependent region is calculated and recorded.

■ NO
  The amount of storage allocated and used by a dependent region is not calculated and recorded.

CPIC1TRN parameter

The parameter handles a special case in which two consecutive sync points occur without any intervening work.

This parameter affects all transactions types.

Syntax

CPIC1TRN=NO|YES

Options

Valid options are:

■ NO (default)
  Use the NO option if you are not experiencing duplicate records.

■ YES
  Use the YES option if you are experiencing duplicate records.
— A transaction record is written only if a DL/I or ESS call was processed since the last transaction record was written.

— A program record is written only if a DL/I or ESS call was processed since the last program record was written.

**CPICDB2 parameter**

The parameter determines when a transaction record is written for CPI-C (explicit APPC) conversations.

**Syntax**

CPICDB2=TERM|SYNC

**Options**

Valid options are:

- **TERM** *(default)*
  
  For CPI-C (explicit APPC) conversations, writes a single transaction record for each conversation.

  The CPICDB2 option is in effect for a conversation until the application does an APSB call. When an APSB call is issued, the CPICDLI option is then in effect for the remainder of the application.

  Workload Analyzer, Workload Monitor, Performance Reporter, and Transaction Accountant report one transaction for each CPI-C conversation.

- **SYNC**
  
  For CPI-C (explicit APPC) conversations, writes a transaction record for each Commit call (SRRCMIT).

  Workload Analyzer, Workload Monitor, Performance Reporter, and Transaction Accountant report one transaction for each sync point.

**CPICDLI parameter**

The parameter determines when additional transaction records (in conjunction with the CPICDB2 parameter) are written for CPI-C (explicit APPC) conversations.
## Syntax

```plaintext
CPICDLI=APSB|SYNC
```

## Options

Valid options are:

- **APSB (default)**
  
  Writes a transaction record each time a new PSB is allocated by an APSB call. The CPICDLI option overrides the CPICDB2 option if and when the application does an APSB call.

  Workload Analyzer, Workload Monitor, Performance Reporter, and Transaction Accountant report one transaction for each APSB call.

- **SYNC**
  
  Writes a transaction record for each Commit call (SRRCMIT).

  Workload Analyzer, Workload Monitor, Performance Reporter, and Transaction Accountant report one transaction for each sync point.

---

## CPU parameter (CPU data collection options)

The parameter controls the level of CPU data collected by the Event Collector in all IMS regions, in just the dependent regions, or in no regions.

**Note**

This field includes user-attributable CPU time incurred in DB2 through the IMS Attach Facility if the parameter FEATURE=NODB2 is specified in PARMLIB member IMFSYS00 (see PARMLIB member IMFSYSBB for more information).

## Syntax

```plaintext
CPU=DEP|DEPPGM|DEPDB2|ALL|NONE
```

## Options

Valid options are:
- **DEP (default)**
  Collects CPU time data for transaction processing from dependent regions only. The data is recorded as application, DL/I, and DB2 time. Most chargeable CPU time is still collected unless LSO=Y or BMPs and JBPs are run with nonparallel DL/I.
  
  CPU fields for CONTROL and DLISAS records are zero. The Performance Reporter and the Transaction Accountant scheduling and open/close CPU fields are zero.
  
  When using the DEP option, the Event Collector attributes chargeable CPU application program or DL/I processing CPU time to a specific transaction and user. Event Collector CPU usage increases 25 to 35 percent (compared to CPU=NONE), depending on the amount of DL/I activity.

- **DEPPGM**
  Collects all CPU time data, including from dependent regions. Data is recorded as application time and includes DL/I and DB2 time. Most chargeable CPU time is still collected.
  
  CPU fields for CONTROL and DLISAS records are zero. Application program CPU contains all dependent region chargeable CPU. Message DL/I, buffer, and DB2 CPU contain zeros. This data is reflected in all CPU data shown by the Performance Reporter, the Transaction Accountant, and workload trace.
  
  When using the DEPPGM option, the Event Collector times all DB2 activities, including 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.
  
  Using the DEPPGM option 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.

- **DEPDB2**
  Collects CPU time data from dependent regions only. The data is recorded as application and DB2 time. The DL/I CPU time is included in the CPU time.
  
  CPU fields for CONTROL and DLISAS records are zero. Application program CPU contains all dependent region chargeable CPU except DB2 CPU. Message
DL/I and buffer CPU contain zeros. This data is reflected in all CPU data shown by the Performance Reporter, the Transaction Accountant, and workload trace. The DEPDB2 option functions like DEPPGM, except that the Event Collector separates the dependent region DB2/SQL time from the application program CPU time.

When using the DEPDB2 option, the Event Collector times 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.

**ALL**

Collects all CPU time data.

All CPU time data is available. Reported DL/I CPU time is approximately 5 to 15 percent higher than when using the DEP option, depending on the amount of DL/I and DB2 activity and the LSO option.

Using the ALL option 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 MainView for IMS CPU usage by 3 to 12 percent over the DEP option. Using the ALL option is the best choice if 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.

**NONE**

Does not collect CPU data. All CPU time fields in records will contain zeroes.

Using the NONE option uses the least amount of CPU time, but has the greatest loss of data.

### CPUOVHD parameter

The parameter determines the type of overhead information that is collected. The parameter applies only to CPU overhead; it does not affect chargeable CPU time.

Valid options are:
Syntax

CPUOVD=\texttt{YES}\mid\texttt{REFCPU}

Options

- **YES** (default)  
  Collects control region overhead and DLISAS region overhead in the MainView for IMS program log record (X’F9’), even if the CPU parameter is not set to ALL.  
  Control region overhead and DLISAS region overhead are always collected when the CPU parameter is set to ALL; therefore, when the CPUOVD parameter is set to YES, there is no effect when the CPU parameter is set to ALL. When the CPUOVD parameter is set to YES, there is no effect on dependent region overhead.  
  Overhead CPU fields in the records include control region overhead and DLISAS region overhead options

- **REFCPU**  
  Collects overhead CPU options in the MainView for IMS program log record (X’F9’) based on the CPU parameter specification.  
  The data collected is determined by the use of the CPU and CPUOVD parameters together.

  — If you do not want to collect CPU overhead data, use these options:  
    \begin{verbatim}
    CPUOVD=REFCPU  
    CPU=NONE
    \end{verbatim}
  
  — If you want to collect all CPU overhead data from control, DLISAS, and dependent regions, use these options:  
    \begin{verbatim}
    CPUOVD=REFCPU  
    CPU=ALL
    \end{verbatim}
  
  — If you want to collect CPU overhead data from the dependent region, without collecting DLISAS or control region CPU overhead data, use these options:  
    \begin{verbatim}
    CPUOVD=REFCPU  
    CPU=DEP
    \end{verbatim}

The CPU overhead options are set to zero when no CPU timing is done. In other words, control region overhead CPU is set to zero when the CPU parameter is set to NONE or DEP. Message region overhead CPU is also set to zero when the CPU parameter is set to NONE.
**DBFFLVL parameter**

The parameter determines whether to collect full function database activity data.

**Syntax**

\[
\text{DBFFLVL} = 1 \mid 0
\]

**Options**

Valid options are:

- **1** *(default)*
  - Full function database activity data is collected.
  - Full function DL/I database call and I/O statistics are collected for display in the database activity views.
  - The 1 option uses more CPU than the 0 option.

- **0**
  - Full function database activity data is not collected.
  - Full function DL/I database call and I/O statistics are not collected for display in the database activity views.
  - The 0 option does not affect CPU usage.

**DBFP parameter**

The parameter determines whether to collect counts of NONKEY WRITES and NO I/O ALTERS.

**Syntax**

\[
\text{DBFP} = \text{NO} \mid \text{YES}
\]

**Options**

Valid options are
- **NO (default)**
  Collects counts of NONKEY WRITES and NO I/O ALTERS and reports them in the database trailer.
  Counts of NONKEY WRITES and NO I/O ALTERS are collected and are available for performance analysis and billing.

- **YES**
  Does not collect counts of NONKEY WRITES and NO I/O ALTERS.
  NONKEY WRITES and NO I/O ALTERS counts are not reported in the database trailer.
  Use the YES option if statistics are not required for performance analysis or billing.
  Using the YES option reduces CPU usage for Fast Path transactions.

**DBFPLVL parameter**

The parameter determines whether DEDB activity data is collected and for which types of activity data is collected.

**Syntax**

```
DBFPLVL=3|2|1|0
```

**Options**

Valid options are:

- **3 (default)**
  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.
  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 database activity views.
  The 3 option uses more CPU than the other options.

- **2**
  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.
  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 database activity views.
  The 2 option uses less CPU than the 3 option, but more than the 1 and 0 options.
DBIO parameter (database I/O options)

The parameter 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.

---

**Note**

The DBIO parameter does not affect Fast Path databases.

---

**Syntax**

DBIO=IOWAITS | BFALTERS | NONE

**Options**

Valid options are:

- **IOWAITS (default)**
  
  Collects reads for each database; collects writes at the transaction level. Sets the BHTO parameter to OFF. NO-I/Os are not collected. The IOWAITS option does not apply to Fast Path.

  All database I/O data is collected at I/O IWAIT. Performance Reporter, Transaction Accountant, and the workload trace services I/O counts per transaction are very close to values with the BFALTERS option. Performance Reporter NO I/O counts are zero. Performance Reporter database report shows READs as usual; most WRITEs, which occur during sync point, are collected for the transaction in a special database trailer, ALLDBS, instead of per database.

  The IOWAITS option 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.

Using the IOWAITS option 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.

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

- **BFALTERS**
  Collects all I/O data for each database.
  Database I/O and NO-I/O data is collected in the buffer handler interface (during the DL/I call).
  With using 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, can also 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.

- **NONE**
  Does not collect any I/O data. Sets the BHTO parameter to OFF. The NONE option does not apply to Fast Path.
  DL/I calls are still available by database.

### DBTNAME parameter

The parameter determines whether to collect DL/I call counts only or to collect DL/I call counts and database I/O call counts.

**Syntax**

```
DBTNAME=DB | DD
```
Options

Valid options are:

- **DB** *(default)*
  Collects DL/I call counts and database I/O call counts at the database level (DBPCB name). A database trailer (DBT) is created for each DBPCB name that contains DL/I and I/O calls.
  When using the DB option, there is a minimal increase in CPU usage.

- **DD**
  Collects DL/I call counts at the database level. Collects database I/O counts at the data set level (ddname). The DD option does not apply to Fast Path.
  A DBT is created for each DBPCB name containing DL/I calls. A DBT is created for each ddname containing I/O calls.
  The DD option uses more database trailers than the DB option.
  When using the DD option, there is a larger increase in CPU usage than when using the DB option.

DBTS parameter

The parameter sets the maximum number of database trailers allowed per non-BMP/JBP region.

If a transaction accesses more than this number of databases, resource data is collected in an overflow trailer named OTHERS.

Using the DBTS parameter does not affect CPU usage; it does, however, affect ECSA requirements for each region.

**Syntax**

```
DBTS=nnn
```

**Options**

The valid value is 2 through 500. 20 is recommended; and 10 is the default.
**DBTS4BMP parameter**

The parameter sets the maximum number of database trailers allowed per BMP or JBP region.

If a transaction accesses more than this number of databases, resource data is collected in an overflow trailer named OTHERS.

Using the DBTS4BMP parameter does not affect CPU usage; it does, however, affect ECSA requirements for each region.

**Syntax**

```plaintext
DBTS4BMP=nnn
```

**Options**

The valid option value is 2 through 500 (30 is recommended). If a value is not specified, the value specified for the DBTS parameter is the default.

---

**DEPREC parameter (extended recovery)**

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

When using the YES option, MainView for IMS CPU usage may be increased 10 to 30 percent compared to using the 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 option should remain set until MainView for IMS is thoroughly tested and stabilized 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.

**Syntax**

```plaintext
DEPREC=YES|NO
```

**Options**

Valid options are:
- **YES** *(default)*
  Performs extended recovery. Performs recovery for additional abend conditions.

- **NO**
  Performs basic recovery. Some potential abend conditions cannot be recovered.

### DTSQLCPU parameter

The parameter controls whether CPU time for individual DB2 SQL events is collected and recorded for detail traces. This parameter has no affect unless a detail trace is active.

#### Syntax

DTSQLCPU=YES|NO

#### Options

Valid options are:

- **YES** *(default)*
  
  CPU time used by each SQL event is calculated and recorded.
  Elapsed time and CPU time used by each SQL call is available.
  Using the YES option, can add a significant increase in DB2 transaction CPU and elapsed times when detail tracing is active.

- **NO**
  
  CPU time is not calculated for individual SQL events, but total DB2 CPU time is still recorded.
  Elapsed time used by each SQL call is available.
  Using the NO option can significantly reduce CPU and elapsed times for DB2 transaction when detail tracing is active.

### DUMPS parameter

The parameter determines whether a dump should be taken when the Event Collector abnormally terminates.
Note
The DUMPS parameter is provided for compatibility with MainView for IMS Online version 4.1.00 and earlier. For MainView for IMS Online version 4.2.00 and later, a dump is taken for any Event Collector problem.

**Syntax**

DUMPS=YES|NO

**Options**

Valid options are:

- **YES (default)**
  A dump is taken when the Event Collector encounters a problem.

- **NO**
  A dump is not taken when the Event Collector encounters a problem.

**ETIMEELP parameter**

The parameter determines whether to collect transaction elapsed time.

**Syntax**

ETIMEELP=YES|NO

**Options**

Valid options are:

- **YES (default)**
  Collects transaction elapsed time and stores it in the following categories:
  - Application
  - DL/I TM calls
  - DL/I DB calls
  - DB2 calls
— MQSeries calls
— Other ESS calls
— Database open and close
— Sync point
— Schedule-to-first DL/I calls
— Last DL/I call

Transaction elapsed times are collected and categorized and are available for display in the summary trace service (STRAC) and in the region views and Summary Trace Display dialog.
Using the YES option minimally increases CPU usage.

■ NO

Does not collect transaction elapsed time.
Transaction elapsed times are not available for display in the STRAC nor in the region views and Summary Trace Display dialog.
Using the NO option has no affect on CPU usage.

**ETIMEIWT parameter**

The parameter determines whether to collect transaction IWAIT time.

**Syntax**

ETIMEIWT=YES|NO

**Options**

Valid options are:

■ YES *(default)*

Collects transaction IWAIT time and stores it in the following categories:

— Sync point: latch, VSAM, OSAM, and unclassified
— DL/I TM calls: latch and unclassified
— Database open and close calls: DBRC, I/O, and unclassified
— DL/I DB calls: latch, lock, VSAM, OSAM, DEDB, and unclassified
— Unclassified

Collected data is available for display in the summary trace service (STRAC) and in the region views and Summary Trace Display dialog.
The amount of CPU usage depends on the amount of transaction IWAIT time. The increase is probably minimal.

■ NO
  Bypasses collection of transaction IWAIT time.
  No data is available for display in the STRAC nor in the region views and Summary Trace Display dialog.
  Using the NO option does not affect CPU usage.

MAXDS parameter

The parameter determines the maximum space in MB allocated to data space.

The actual minimum amount of space allocated is dependent on the size of the pools allocated by other parameters. If the specified size is not large enough, it will be adjusted upward.

Syntax

```
MAXDS=n
```

Options

The valid value is 15 through 9999 (50 is the default).

MSCCLOCK parameter

The parameter determines whether the Event Collector converts a transaction’s arrival date and time to local time.

Syntax

```
MSCCLOCK={DEFAULT, NOTSYNC|DEFAULT, SYNC}|{nnnn, NOTSYNC|nnnn, SYNC}
```
Options

Valid options are:

■ DEFAULT,NOTSYNC | DEFAULT,SYNC

If a transaction originates from an MSC system that is defined to have a synchronized clock, the Event Collector will convert its arrival date and time (set by the originating MSC system) to local time. The converted time is then stored in the transaction arrival date and time fields of the transaction log record (X'FA'). The MSC systems can be in the same time zone as the local IMS or in a different time zone.

If a transaction originates from an MSC system that is not defined as synchronized with the local IMS clock, no conversion is done and the transaction arrival date and time are set to the start date and time on the local IMS.

Exceptions to the DEFAULT option can be specified with xxxx,NOTSYNC or xxxx,SYNC option.

Only one MSCCLOCK=DEFAULT,NOTSYNC | DEFAULT,SYNC record is accepted (the first one specified); if another DEFAULT record is encountered, it will be ignored.

The MSCCLOCK Event Collector parameter works in conjunction with the monitor TERM parameter operands SYNCLOCK and MSCCLOCK, which are used to qualify workload monitor and trace requests. TERM=SYNC LOCK defines a workload monitor or trace request to include all transactions from systems synchronized to the clock of the local IMS. TERM=MSCCLOCK defines the request to include only nonlocal transactions from MSC systems synchronized to the clock of the local IMS.

■ nnnn,NOTSYNC | nnnn,SYNC

Used to specify exceptions to the MSCCLOCK DEFAULT setting.

The value nnnn represents an MSC ID, which can range from 0001 to 2036. An MSC exception specification is valid whether it precedes or follows the DEFAULT,NOTSYNC or DEFAULT, SYNC record.

MSGOQBUF parameter

The parameter specifies the maximum number of internal work elements that will be allocated to send output queue information from IMS to the product address space. The MSGOQBUF parameter applies when the MSGOQTME parameter is set to YES.

If the PAS is unable to keep up with data arrival, you can increase this value to ensure that information is not lost.

CPU usage is minimal if the MSGOQTME parameter is set to YES. CPU usage is not affected if the MSGOQTME parameter is set to NO.
Syntax

MSGQQBUF=nnnnn

Options

The minimum option value is 1000. The default value is calculated as four times the value specified for the IMS MAXPST parameter.

MSGQQTME parameter

The parameter determines whether message output queue time is collected for Fast Path and full function transactions.

Syntax

MSGQQTME=YES|NO

Options

Valid options are:

- **YES** *(default)*
  Message output queue time is collected for Fast Path and full function transactions. Output queue time is also collected for MSC, message switch, and shared queues.
  Output queue time data is available for display in the total response time views.
  When using the YES option, there is a small increase (four to five percent) in CPU usage.

- **NO**
  Message output queue time is not collected.
  Output queue time data is not available for display in the total response time views.
  When using the NO option, CPU usage is not affected.

MXPASRQ parameter

The parameter determines the number of FA request elements allocated in the data space.
The request elements are used for sending transaction data to the product address space upon transaction completion.

**Syntax**

\[ \text{MXPASRQ}=n\text{n}\text{n}\text{n}\text{n} \]

**Options**

The valid option value is 100 through 99999. The default value is calculated as two times the value specified for the IMS MAXPST parameter.

**MXPASSQ parameter**

The parameter determines the maximum amount of space that can be used in the data space to queue an FA record.

**Syntax**

\[ \text{MXPASSQ}=n\text{n}\text{n}\text{n}\text{n} \]

**Options**

The valid option value is 1 through 99999.

**RGNIOPT parameter**

The parameter determines whether to abend the IMS dependent region if MainView for IMS initialization fails because of CSA shortage.

**Syntax**

\[ \text{RGNIOPT}=\text{ABEND|CONTINUE} \]

**Options**

Valid options are:
- **ABEND (default)**
  Abends the IMS dependent region if MainView for IMS initialization fails because of CSA shortage.
  Complete data recording is ensured.

- **CONTINUE**
  Continues the IMS dependent region if MainView for IMS initialization fails.
  No data recording is done for the region.

**SMFID parameter**

The parameter determines the ID of the SMF transaction record.

**Syntax**

SMFID=nnn

**Options**

The valid option value is 1 through 255.

**SUBSYS parameter**

The parameter determines the subsystem ID for system events.

**Syntax**

SUBSYS=nnnn

**Options**

The valid option value is a 1- to 4-character subsystem ID.

**SYSID parameter**

The parameter identifies the MainView for IMS system for the Performance Reporter and the Transaction Accountant.
**Syntax**

SYSID=x

**Options**

The valid option value is 1 through 9 and A through Z. 1 is the default value.

---

**TELON parameter**

The parameter determines support for the TELON application package.

**Syntax**

TELON=NO|YES

**Options**

Valid options are:

- **NO** *(default)*
  
  Disables specific data collection for the TELON application package.

- **YES**
  
  Supports the TELON application package by replacing the AGN field in the transaction and program records with the internal TELON transaction name carried in the transaction input message (for other TELON support options, see “Modification to support TELON” on page 119.

---

**TIMERR parameter**

The parameter determines if a region abends if the time values are out of range.

**Syntax**

TIMERR=NO|U4089

**Options**

Valid options are:
- NO *(default)*
  A region does not abend if the time values are out of range.
  The timer error flag and timer error dispatcher type fields are set in the X'FA' log record.

- U4089
  A region does abend with a U4089 if the time values are out of range.
  The timer error flag and timer error dispatcher type fields are set in the X'FA' log record.

**TRNSYNC parameter**

The parameter determines whether to write a transaction record at a BMP or JBP checkpoint.

**Syntax**

TRNSYNC=NO|YES

**Options**

Valid options are:

- NO *(default)*
  Does not write a transaction record at a BMP or JBP checkpoint. A record is written only per each successful MESSAGE GET UNIQUE or at program end.

- YES
  Writes a transaction record at a BMP or JBP checkpoint.
  CPU usage is minimally affected.

**WLMSC parameter**

The parameter determines whether the WLM service class is reported on the TRN and PGM records.

**Syntax**

WLMSC=NO|YES
Options

Valid options are:

- **NO** *(default)*
  The WLM service class is not reported on the TRN and PGM records.

- **YES**
  The WLM service class is reported on the TRN and PGM records.

ZTIME parameter

The parameter determines whether to monitor zIIP and zAAP usage within IMS dependent regions and to record usage amounts in the FA record.

**Syntax**

ZTIME=NO|YES

**Options**

Valid options are:

- **NO** *(default)*
  Does not monitor and record zIIP and zAAP usage. The FA field TRNPGCPU includes any zIIP and zAAP time used by the dependent region, along with general purpose CP usage, but separate zIIP and zAAP statistics are not recorded.

- **YES**
  Monitors zIIP and zAAP usage by the dependent region. The usage is recorded in the FA extension area mapped by DSECT TRXZ, in BBSAMP member IMETRNX. Additional statistics are recorded in the F9 record, mapped by IMEPGM. CPU usage is minimally affected.
Define target systems

The MainView for IMS and MainView for DBCTL products operate 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 Customization Reference*, BBPARM member BBIJNT00 is used to define all eligible target systems and associate them with the subsystem IDs of a BBI-SS PAS.

**Note**  
Defining target systems is only applicable for MainView for IMS Online and MainView for DBCTL.

To define a target system:

- Use the jobname of the 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=1n10
  ```
Customize user exit routines

BMC provides samples of several user exit routines for the Event Collector, for access to the IMS log file records and to IRUF data, and for changing the IRUF file during creation.

Event Collector user exit routines

The Event Collector user exit routines are skeleton programs in BBSAMP that you can customize to build more extensive routines to meet your specific needs.

Transaction record user exit routine (IMRUTRNL)

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 IMRUTRNL3.

For special considerations about using this exit, see “IMRUTRNL and IMRUPGM cross-memory mode considerations” on page 110.

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.

For special considerations about using this exit, see “IMRUTRN and IMRUPGM cross-memory mode considerations” on page 110.

**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 the operating system 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 IMFECPO0. 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 IMFECPO0, the MainView for DBCTL product takes one of the following actions.

- If BACKOUT=NO, MainView for DBCTL abends the IMS control region
- If BACKOUT=YES, MainView for DBCTL disables the Event Collector

The BACKOUT parameter, specified in IMFECPO0, has a default of YES (see “Event Collector parameters” on page 73).

**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.
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).

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

**WARNING**

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.

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

---

**Log edit user exit routines**

User exit routines supplied by BMC, PRSLEXIT and PRSCEEXIT, provide user access to the IMS log file records and to IRUF data. These routines are written in Assembler language and are distributed in BBSAMP members PRSLEXIT and PRSCEEXIT.

**Note**

The log edit user exit routines are used only for MainView for IMS Offline.

PRSLEXIT is a dummy exit routine for the IMS log file records.

PRSCEEXIT is a default exit routine for IRUF data.

If you write your own user exit routines, you should write them in Assembler language for performance reasons. You can use the BBSAMP members for reference.

If you are going to use user-written exit routines for PRSLEXIT, PRSCEEXIT, or both, instead of the routines supplied by BMC, you must link the user-written routines with a front-end module (PRSLEXTB or PRSCEXITB), using the link edit statements from BBSAMP members LNKCEXIT and LNKLEXIT.

Program IMFLEDIT calls PRSCEEXIT and PRSLEXIT as subroutines. Control is passed to PRSLEXIT at the start of processing, each time a record is read from the log file, and at the end of log file processing. Control is passed to PRSCEEXIT at the start of processing, at the end of IRUF processing, and before each record is written to the IRUF data set.
If user exit routines are written in Assembler, two additional user fields are available in the common parameter area. The fields can be used to save the address for intercommunication between the log user exit and the customer user exit. At entry to each exit routine, Register 3 points to the common parameter area. The two user fields are four bytes long and are located at offsets X’27C’ and X’280’ into the common area.

**Exit routine to IMS log file data (PRSLEXIT)**

The distributed version of PRSLEXIT is a dummy exit routine provided to satisfy linkage requirements. PRSLEXIT provides user access to the IMS log file records.

You can incorporate other log processing functions into the IMS log edit to minimize the number of passes required against the log files.

PRSLEXIT receives control through the following entry points:

<table>
<thead>
<tr>
<th>Entry point</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEXITEP1</td>
<td>For initialization</td>
</tr>
<tr>
<td>LEXITEP2</td>
<td>For log record processing</td>
</tr>
<tr>
<td>LEXITEP3</td>
<td>For exit termination</td>
</tr>
</tbody>
</table>

The initiating (LEXITEP1) and terminating (LEXITEP3) entry points are given control without any parameter list. The record processing (LEXITEP2) entry point is given control with a pointer to the record area containing the log record just read by the edit.

The exit routine can be coded in any operating system language that supports standard subroutine linkage.

**Exit routine to IRUF data (PRSCEXIT)**

PRSCEXIT is an exit routine you can use to access and modify the contents of each IRUF record before it is written out. With PRSCEXIT, you can:

- Build your own customer identification (CI) field
  You can then make your own logical grouping of users/LTERMS before the offline reports are written. For example, you can group information by department or by logical function, instead of by LTERM name (the default). This option is particularly useful for charge-back facilities.

- Exclude a record from being written to the IRUF
- Associate BMP and JBP transaction accounting records with their appropriate LTERM and transaction codes

The defaults for the distributed version of PRSCEXIT are to:

- Give reports by user/LTERM
  The customer identification field is built by moving the LTERM name to the first eight positions and moving spaces to the remaining ten positions of the field. The field is built only for terminal and transaction accounting records. Program records cannot be identified by user because multiple transactions may have been processed by one program.

- Permit all records to be written to the IRUF

- Build the transaction category field by moving the numeric value of the transaction class to the transaction category

- Construct the transaction code for a pseudo-BMP or pseudo-JBP transaction record by moving the program name to the transaction code field in the transaction accounting record
  This transaction code is created only for a BMP or JBP that has not processed any transactions. A pseudo-transaction record is written to account for all processor resources used and database I/O activity incurred. However, the log record is written with only a program identification, while LTERM and transaction code are blank. The transaction code is set to the program name in this exit, while LTERM is set to ??-BLANK in IMFLEDIT.

PRSCEXIT receives control through the following entry points:

<table>
<thead>
<tr>
<th>Entry point</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEXITPEP1</td>
<td>For initialization</td>
</tr>
<tr>
<td>CEXITEOF</td>
<td>For exit termination</td>
</tr>
<tr>
<td>CEXITTER</td>
<td>For terminal accounting record processing</td>
</tr>
<tr>
<td>CEXITPSB</td>
<td>For program accounting record processing</td>
</tr>
<tr>
<td>CEXITTRN</td>
<td>For transaction accounting record processing</td>
</tr>
</tbody>
</table>

The initiating (CEXITEP1) and terminating (CEXITEOF) entry points are given control without any parameter list. The other entry points (CEXITTER, CEXITPSB, and CEXITTRN) are given control with a parameter list that contains pointers to the IRUF record and the NEGATE RECORD SW field.

The NEGATE RECORD SW switch indicates to Log Edit whether or not the record should be written to the IRUF. Before control passes to the exit routine, this one-byte switch is set to a value of C'0'. If the exit routine is to signal Log Edit to negate the writing of the record, a value of C'1' must be moved to the NEGATE RECORD SW.
When this indicator is set by the exit routine, Log Edit bypasses writing the record to the IRUF data set.

The exit routine can be coded in any operating system language if standard subroutine linkage is supported.

## Summarized IRUF exit

TASCOSTR supports a user exit that allows you to change the IRUF file during creation. A sample Assembler exit is provided in BBSAMP member TASEXIT.

This exit must be named TASEXIT and is link-edited into module TASCOSTR.

You may have to relink your user exit after maintenance is applied.

### Note

The summarized IRUF exit is used only for MainView for IMS Offline.

## Exit invocation points

The entry points during which control is passed to the user-written exit routine are shown in the following table:

<table>
<thead>
<tr>
<th>Entry point</th>
<th>Entry time</th>
</tr>
</thead>
</table>
| TEXITEP1    | When the system is initialized  
There is no call parameter list. |
| TEXITEP2    | As a summary transaction record is about to be written to the IRUF identified by the DETCOSTS DD statement  
The call parameter list points to the IRUF record about to be written. |
| TEXITEP3    | When the system is terminating  
There is no call parameter list. |
| TEXITEP4    | After an IRUF record is read from the file identified by the RESUTIL DD statement and before costing  
All fields in the transaction accounting record (TAR) and the program accounting record (PAR) are available at this time. See BBSAMP members CIMPAR01, CIMTAR03, CIRUFR03, and SASIRUF for information about the individual fields. |
Linkage considerations

The following linkage editor statements are required to include a modified version of this user exit in the composite load module:

```
INCLUDE USERLIB(TASEXIT)
INCLUDE IMFLOAD(TACCOSTK)
ENTRY TACCOSTK
NAME TACCOSTR(R)
```

These statements are in BBSAMP member TACCOSTR.
Optional modifications

Optional modifications are available for the MainView for IMS product.

**Note**
Optional modifications only pertain to MainView for IMS Online and MainView for IMS Offline.

Modification to support SAP

LINKSAP in the BBSAMP distribution data set is a sample job that supports SAP application programs.

This support replaces the IMS transaction code with the SAP identifier in the Event Collector transaction record (X'FA'). The SAP identifier is usually the SAP report ID. If the report ID is not available, the SAP transaction code is used as the SAP identifier.

This support allows you to analyze your SAP workload by using:

- Performance Reporter and Transaction Accountant batch reports that provide the SAP identifier instead of the IMS transaction name
- Workload Monitor selection criteria with a service request to limit data collection based on the SAP identifier instead of the IMS transaction name
- The PRSPRINT report which indicates whether the SAP exit was invoked
- The PRSSELEC utility to select SAP-related records

To activate this support, you must:

- Install SAP 5.0c or later
  Contact SAP Corporation of Germany for SAP installation information.
- Make a copy of the distributed BBSAMP member LINKSAP and modify and run the JCL as instructed in this member.

  LINKSAP links a user exit, called SAPEXIT, into the SAP program library. This exit is called by the SAP routines for each transaction to pass the SAP identifier to the Event Collector.

## Modifications to reconcile response time monitoring data

Certain transactions (such as message switch transactions, which have the same arrival time as the original transaction) can distort the average response times in Workload Analyzer and Workload Monitor monitor data by indicating a large input queue time.

Use the MSGSWA2S parameter in BBPARM member IMFBEX00 to set message switch transaction arrival time. Set MSGSWA2S to:

- **Y** to use the transaction start time as the arrival time
- **N** (the default) to use the arrival time of the first transaction in the message switch chain

To see the current values of the parameters in the IMFBEX00 member, use the IBEXSUMR view.

**Note**

Response time modifications are only applicable for MainView for IMS Online.

## SMP-applied modifications

The IBM System Modification Program (SMP) should be used to apply these modifications. Using SMP ensures that these modifications are not regressed when a MainView for IMS service is applied.

If a MainView for IMS service is applied to any of the user-modified CSECTs, the user modifications must be restored with an SMP RESTORE and APPLY after the service is applied.

Use the IBM service aid utility AMBLIST to locate the offsets for the entry points in the CSECTs. Sample JCL and a control statement for using the AMBLIST utility are as follows:
**Example**

```plaintext
// JOB
// EXEC PGM=AMBLIST
//SYSPRINT DD SYSOUT=* 
//SYSLIB DD DSN=&prefix.BBLINK,DISP=SHR <=== BBLINK LIBRARY 
//SYSIN DD *
LISTLOAD OUTPUT=XREF, MEMBER=(IMPSRV,IMECSAU1,IMECSAU9)
/*
```

---

**Modification to support TELON**

This modification supports the TELON application development package. Activate MainView for IMS support for the TELON in one of the following ways:

- Specify TELON=YES in BBPARM member IMFECPP00.
  
  TELON=YES causes the AGN field in the transaction and program records to be replaced with the internal TELON transaction name carried in the transaction input message.

  If the first three bytes of the IMS transaction code match the internal TELON transaction code name, the internal TELON transaction code replaces the IMS transaction code and PSB name in the MainView for IMS log records.

  Specifying TELON=YES makes the information available to the IMFLEDIT user exit routines.

- ZAP the unconditional branches (X’47F0’) at the locations shown in the following table to NOPs (X’4700’). If more than one release of IMS is in use, SUPERZAP the appropriate CSECT.

**Note**

If you want only the internal TELON transaction code to be placed in the AGN field and you do not want to replace the transaction code and PSB name, take these steps:

- Specify TELON=NO in BBPARM member IMFECPP00.

- Use the TELON1 ZAP, rather than the TELON2 ZAP.

<table>
<thead>
<tr>
<th>Load module</th>
<th>CSECT</th>
<th>Entry point</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMECSAUx</td>
<td>IMEDL3Ex</td>
<td>TELON1</td>
</tr>
<tr>
<td>IMECSAUx</td>
<td>IMELTRNx</td>
<td>TELON2</td>
</tr>
</tbody>
</table>

The variable x can be:

- 5 suffix for IMS 11.1
6 suffix for IMS 12.1
7 suffix for IMS 13.1

**Note**
If the MainView for IMS modules are already copied to the IMS library (see “Modifying the IMS control region JCL” on page 29), the modified version of IMECSAUx must be copied to IMS.

### Modification to suppress IMFLEDIT warning messages

The **CPU TIME EXCEEDS TRAN ELAPSED** warning message may not indicate a true error. Instead, it may indicate that a transaction has continued processing after receiving a QC status code from an MGU call.

The Event Collector uses the timestamp from the QC-MGU to set the stop time for the transaction, resulting in a small transaction elapsed time. If the transaction continues processing and collecting CPU time, it causes the CPU time to exceed the small transaction elapsed time.

**Note**
Suppressing IMFLEDIT warning messages is only applicable for MainView for IMS Offline.

To suppress warning messages issued for BMPs and JBPss only or for BMPs and JBPss and MPPs and JMPs, zap the one-byte flag at this entry point.

<table>
<thead>
<tr>
<th>Load module</th>
<th>CSECT</th>
<th>Entry point</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRSPARM</td>
<td>PRSPARM</td>
<td>ENMSGFLG</td>
</tr>
</tbody>
</table>

The one-byte flag at ENMSGFLG contains a value of X'00'. Replace it with one of the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'01'</td>
<td>To suppress warning messages issued for BMPs and JBPss only</td>
</tr>
<tr>
<td>X'02'</td>
<td>To suppress warning messages issued for BMPs, JBPss, MPPs, and JMPs</td>
</tr>
</tbody>
</table>
IMS dump analysis

Use this information to analyze an IMS dump.

Note

IMS dump analysis is only applicable for MainView for IMS Online and MainView for DBCTL.

AO exit dispatch

Except for MainView AutoOPERATOR for IMS MTO message capture for the Journal log, all AO code merely passes control to the specified routines during operation. During initialization, special protection exists while the Event Collector is being set up.

MainView AutoOPERATOR for 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 that pertain to MainView for IMS Online and MainView for DBCTL (products):

- The product register save areas are in the product 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 product module at entry saves the registers of an IMS module in the next IMS prechained save area pointed to by R13.

Product module registers are always saved in product save areas, which are dynamically assigned as required. Normally, one of several preallocated save areas per region is used.

Each active IMS region has a product data area acquired for it at region initialization. This data area is in ECSA and is named IMERDnnn, 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 product save area, whether in IMERDnnn or IMFSP000, is identified with SAR or ISA.

In most cases, only a save area backward pointer exists, pointing from the product save area to the previous IMS save area. While a product module is in control, the current R12 is its base register and the current R13 points to its SAR.

In some cases, a product 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 product module is an R14 value in an IMS save area that 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 product SAR, 4 bytes in front of the address pointed to by the R14 in the IMS save area.

---

**Note**

When a product interfaces between two IMS modules, the product 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.
BBSAMP data set members

To help you understand and use your BMC 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.

Descriptions of several BBSAMP members

The following table describes several of the BBSAMP members used for the MainView for IMS and MainView for DBCTL products. The table does not list all of the members available in BBSAMP.

<table>
<thead>
<tr>
<th>BBSAMP member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCHCTL1</td>
<td>Sample SYSIN for the log archive utility (see ARCHJCL) that selects only MainView for IMS log records</td>
</tr>
<tr>
<td>ARCHCTL2</td>
<td>Sample SYSIN for the log archive utility (see ARCHJCL) that selects all MainView for IMS log records and some IMS log records</td>
</tr>
<tr>
<td>ARCHJCL</td>
<td>Sample JCL for the IMS log archive utility, which produces an MainView for IMS user file</td>
</tr>
<tr>
<td>BLKDBTW</td>
<td>Sample starter set of monitors for MainView for DBCTL</td>
</tr>
<tr>
<td>BLKIMFW</td>
<td>Sample starter set of monitors for MainView for IMS</td>
</tr>
<tr>
<td>CIMLAR01</td>
<td>Sample COBOL layout for IRUF terminal (LTERM) accounting record (LAR)</td>
</tr>
<tr>
<td>CIMPAR01</td>
<td>Sample COBOL layout for IRUF program accounting record (PAR)</td>
</tr>
<tr>
<td>CIMTAR03</td>
<td>Sample COBOL layout for IRUF transaction accounting record (TAR) for MainView for IMS version 4.6 and later</td>
</tr>
<tr>
<td>CIRUFR03</td>
<td>Macro to map all IRUF records in Assembler for MainView for IMS version 4.6 and later</td>
</tr>
<tr>
<td>FPORUN</td>
<td>Sample JCL to read the IMS log and create an IRUF with response option and produce several reports</td>
</tr>
<tr>
<td>GTFIMF</td>
<td>Diagnostics; sample JCL (see GTFUSE)</td>
</tr>
<tr>
<td>BBSAMP member name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>GTFIMFP</td>
<td>Diagnostics; sample JCL (see GTFUSE)</td>
</tr>
<tr>
<td>GTFIMF00</td>
<td>Diagnostics; sample parameters (see GTFUSE)</td>
</tr>
<tr>
<td>GTFUSE</td>
<td>Diagnostics; how to use GTF trace facility for MainView for IMS Event Collector</td>
</tr>
<tr>
<td>ICOPYn</td>
<td>Sample JCL to copy BMC modules from the BBLINK library to a site-authorized library, where n applies to an IMS release</td>
</tr>
<tr>
<td>IMEDBT</td>
<td>Macro to map the database trailer (DBT) for the MainView for IMS transaction log record in Assembler</td>
</tr>
<tr>
<td>IMEPGM</td>
<td>Macro to map the MainView for IMS program log record in Assembler</td>
</tr>
<tr>
<td>IMETMEQU</td>
<td>Macro used within IMETRN</td>
</tr>
<tr>
<td>IMETRN</td>
<td>Macro to map the MainView for IMS transaction log record in Assembler</td>
</tr>
<tr>
<td>IMFACTIV</td>
<td>IMS PR; sample JCL to create a general activity analysis for all LTERM and databases</td>
</tr>
<tr>
<td>IMFARB</td>
<td>Macro referenced when assembling some user-written services</td>
</tr>
<tr>
<td>IMFASYDS</td>
<td>Macro referenced when assembling some user-written services</td>
</tr>
<tr>
<td>IMFCLNDR</td>
<td>Performance Reporter; sample JCL to produce sample calendar reports One sample of each type of calendar report is produced.</td>
</tr>
<tr>
<td>IMFCOSTR</td>
<td>Sample JCL to sort an IRUF into customer ID sequence and summarize it (program TASCOSTR)</td>
</tr>
<tr>
<td>IMFNSUM</td>
<td>Transaction Accountant; sample JCL to create a financial summary analysis from an IRUF</td>
</tr>
<tr>
<td>IMFFPRPT</td>
<td>Sample JCL to create Fast Path transaction processing statistics</td>
</tr>
<tr>
<td>IMFLEDIT</td>
<td>Sample JCL to create an IRUF by editing the IMS system log</td>
</tr>
<tr>
<td>IMFMVSXA</td>
<td>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.</td>
</tr>
<tr>
<td>IMFPROG</td>
<td>Performance Reporter; sample JCL to produce program processing statistics</td>
</tr>
<tr>
<td>IMFREGUT</td>
<td>Performance Reporter; sample JCL to produce message region utilization analysis</td>
</tr>
<tr>
<td>IMFRESP</td>
<td>Performance Reporter; sample JCL to produce a response-time distribution report</td>
</tr>
<tr>
<td>IMFRPTS</td>
<td>Sample JCL to read the IMS log, create an IRUF with response option, and produce several reports</td>
</tr>
<tr>
<td><strong>BBSAMP member name</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>IMFSELEC</td>
<td>Sample JCL to create an extracted IRUF (select a subset of IRUF records) with which to generate reports</td>
</tr>
<tr>
<td>IMFSETAM</td>
<td>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.</td>
</tr>
<tr>
<td>IMFTARC</td>
<td>Sample job to archive a trace log data set</td>
</tr>
<tr>
<td>IMFTRAN</td>
<td>Performance Reporter; sample JCL to produce transaction processing statistics</td>
</tr>
<tr>
<td>IMFTRLOD</td>
<td>Sample job to restore an archived trace log data set</td>
</tr>
<tr>
<td>IMFTRND2</td>
<td>Performance Reporter; sample JCL to produce DB2® transaction processing statistics</td>
</tr>
<tr>
<td>IMFTRNFP</td>
<td>Performance Reporter; sample JCL to produce Fast Path transaction processing statistics</td>
</tr>
<tr>
<td>IMFVT</td>
<td>Macro referenced when assembling some user-written services</td>
</tr>
<tr>
<td>IMRUDLI3</td>
<td>Sample Event Collector user exit routine for DL/I user exit routine CALL/END</td>
</tr>
<tr>
<td>IMRUPGM3</td>
<td>Sample Event Collector user exit routine to access program records for evaluation</td>
</tr>
<tr>
<td>IMRUTRN3</td>
<td>Sample Event Collector user exit routine to access transaction records for evaluation</td>
</tr>
<tr>
<td>JASEXIT</td>
<td>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</td>
</tr>
<tr>
<td>JRNLMSG</td>
<td>Macro referenced when assembling some user-written services</td>
</tr>
<tr>
<td>JXT001</td>
<td>Sample job to set up and maintain a trace log directory</td>
</tr>
<tr>
<td>JXT003</td>
<td>Sample job to verify existence of trace log data set in the system catalog</td>
</tr>
<tr>
<td>JXT011</td>
<td>Sample job to define a trace log data set using IDCAMS</td>
</tr>
<tr>
<td>LNKCEXIT</td>
<td>Linkage editor statements to replace default MainView for IMS Log Edit user exit routine that defines the customer ID with a user-written routine</td>
</tr>
<tr>
<td>LNKLEXIT</td>
<td>Linkage editor statements to replace default MainView for IMS Log Edit user exit routine that accesses the log file with a user-written routine</td>
</tr>
<tr>
<td>LINKSAP</td>
<td>Sample job to link SAPEXIT into SAP program library</td>
</tr>
<tr>
<td>LOGREC</td>
<td>Sample JCL to print the software LOGREC records in SYS1.LOGREC.LOGREC</td>
</tr>
<tr>
<td>PRSCEXIT</td>
<td>Sample Assembler user exit routine to define the customer ID field in the IRUF</td>
</tr>
<tr>
<td>BBSAMP member name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>PRSCEXITC</td>
<td>Sample COBOL user exit routine to define the customer ID field in the IRUF</td>
</tr>
<tr>
<td>PRSLEXIT</td>
<td>Sample Assembler user exit routine to access a log record read from the IMS system log</td>
</tr>
<tr>
<td>PRSLEXTC</td>
<td>Sample COBOL user exit routine to access a log record read from the IMS system log</td>
</tr>
<tr>
<td>PRSLEXTA</td>
<td>Sample user exit routine to access a log record read from the IMS system log and write the record to an external file</td>
</tr>
<tr>
<td>PRSPRINT</td>
<td>Sample JCL to select and print IRUF records</td>
</tr>
<tr>
<td>RARGEN</td>
<td>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.</td>
</tr>
<tr>
<td>RARGFN</td>
<td>Resource Analyzer; macro to generate code for functions selected in macro RARGEN</td>
</tr>
<tr>
<td>RAUSR00</td>
<td>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 operating system control blocks can be accessed.</td>
</tr>
<tr>
<td>RMUSR01</td>
<td>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 operating system control blocks can be accessed.</td>
</tr>
<tr>
<td>RTOPTCH</td>
<td>Macro to generate a patch area for MainView for IMS modules. This macro should not be issued more than once in a module.</td>
</tr>
<tr>
<td>SASIRUF</td>
<td>SAS definition of IRUF terminal, program, and transaction accounting records</td>
</tr>
<tr>
<td>TACCOSTR</td>
<td>Transaction Accountant; sample linkage editor statements to link a user exit (TASEXIT) into the Transaction Accountant version of program TASCOSTR</td>
</tr>
<tr>
<td>TASxxxxx</td>
<td>Sample layouts for the IRUF records used in Transaction Accountant</td>
</tr>
<tr>
<td>TASEXIT</td>
<td>Sample user exit routine for TASCOSTR IRUF summarization</td>
</tr>
<tr>
<td>WATBTRAC</td>
<td>Sample batch JCL to print history trace data</td>
</tr>
</tbody>
</table>
Specify product keys

Product keys are used to authorize older implementations of the products.

Specifying product keys involves:

- Determining your internal CPU model and version number
- Obtaining password keys from BMC
- Creating and modifying BBPARM member BBKEYS

Determining the internal CPU model and version number

Use the following procedure to determine the internal model and version number of the CPUs at your site:

1. Display your existing password keys and the current CPU ID by performing one of the following actions:
   - Enter the following BBI control command from your MainView terminal session:
     `.D KEYS`
     The command output is written to the BBI-SS PAS journal log.
   - Enter the following MVS MODIFY command:
     `/F ssid,.D KEYS`
     `ssid` is the BBI-SS PAS job name or started task ID. The command output is written to the SYSLOG or console.

Output like the following example is produced:

`/F 19APAS,.D KEYS`
`COPY PROTECTION KEY STATUS: P49A`
2 Extract the CPU information by using the BMC SYSINFO utility:

   a On the system image where the products will execute, enter the following command:

   TSO CALL 'hlq.BBLINK(SYSINFO)'

   hlq is the high-level data set qualifier.

   b Record the following information from the SYSINFO output:

      ▪ CPU IDENTIFIER
      ▪ CPU TYPE
      ▪ CPU MODEL

3 Use TSO TEST to display storage on the active system by performing the following steps:

   a From the TSO READY prompt, type the command TEST
      ‘SYS1.LINKLIB(IEFBR14)’ and press Enter.

   b Type L 208.%+4 L(12) C

      The following output is displayed:

      aaaaaaaaa. XXnYYYYZZZZ

      The output variables areas follows:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaaaaaaaa</td>
<td>Hexadecimal characters (not needed for password key)</td>
</tr>
<tr>
<td>XX</td>
<td>CPU model version</td>
</tr>
<tr>
<td>n</td>
<td>Numeric value (not needed for password key)</td>
</tr>
<tr>
<td>YYYYY</td>
<td>CPU serial number (CPU ID)</td>
</tr>
</tbody>
</table>
Creating and modifying BBPARM member BBKEYS

Perform the following steps to create and modify the BBKEYS member:

1. Create a member called BBKEYS in the BBPARM data set.
2. Edit BBKEYS to add a key parameter for each product key.

**Note**
Each key parameter must begin on a separate line. During key validation, MainView validates each key and selects the key that is most qualified for your CPU.

Specify the key parameter as follows:

```
KEY=ppp -cccc-tt-nnnnn-yyddd -xxxx ssss[-]
```

Specify the variables according to the following table:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZZZZ</td>
<td>CPU type</td>
</tr>
<tr>
<td><strong>PPP</strong></td>
<td>Three-character product ID:</td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>IPX</td>
<td>IMSplex System Manager</td>
</tr>
<tr>
<td>IRA</td>
<td>IMS Resource Analyzer</td>
</tr>
<tr>
<td>IRM</td>
<td>IMS Resource Monitor</td>
</tr>
<tr>
<td>IWA</td>
<td>IMS Workload Analyzer</td>
</tr>
<tr>
<td>IWM</td>
<td>IMS Workload Monitor</td>
</tr>
<tr>
<td>IAD</td>
<td>IMS Workload Analyzer Extension for DB2</td>
</tr>
<tr>
<td>IMD</td>
<td>IMS Workload Monitor Extension for DB2</td>
</tr>
<tr>
<td>ILA</td>
<td>IMS Resource Analyzer Extensions for IRLM and IMS Resource Monitor Extensions for IRLM</td>
</tr>
<tr>
<td>IPF</td>
<td>IMS Performance Reporter</td>
</tr>
<tr>
<td>IPD</td>
<td>IMS Performance Reporter Extension for DB2</td>
</tr>
<tr>
<td>ITA</td>
<td>IMS Transaction Accountant</td>
</tr>
<tr>
<td>ITD</td>
<td>IMS Transaction Accountant Extension for DB2</td>
</tr>
</tbody>
</table>

| **cccc** | Four-character CPU type (for example, 3090) |
| **tt**   | Two-character CPU internal model group type (for example, 23) |
| **nnnnn**| Four- to five-digit CPU serial number in hexadecimal format |
| **yyddd**| Julian expiration date |
| **xxxx** | Four-character authorization key that BMC provided |
| **ssss** | SMF ID of the system where the product should be activated |

**Note**

- You can use a generic * qualifier for the **ppp**,**cccc**, **tt**, and **nnnnn** values; however, the password key that you received from BMC must have been generated with the same qualifier.

- Specifying **ssss** helps prevent an IPL failure. With **ssss** specified, an invalid password key can be identified, and a WTOR message can be issued so that the operator can specify another key immediately. If the SMF ID of the target system (**ssss**) is not specified, no external warning messages are issued for invalid keys.
Message changes

See the following topics to determine the messages that were added, changed, and removed with recent product releases:

- “Version 5.1.00 messages” on page 131
- “Version 4.6.00 message changes” on page 132
- “Version 4.5.00 message changes” on page 133

Version 5.1.00 messages

The following tables list additions or changes to messages in version 5.1.00 that are issued:

- During initialization or termination of the product in the product address space
- By the product in the IMS address spaces

These messages are ones that you might want to use for your automation processes. Documentation for all messages is in the BMC Documentation Center. You can also use the MainView online message facility to see information about any MainView product message.

Table 4: Messages added with version 5.1.00

<table>
<thead>
<tr>
<th>ID</th>
<th>Message text</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBFPBD58</td>
<td>Enclave setup failed RC=rc</td>
</tr>
<tr>
<td>BBFPSS24</td>
<td>Tran arrival rate table init error, RC=nn, RE=nn, RV=nn</td>
</tr>
<tr>
<td>BBFPSS25</td>
<td>Sampler task failed to join Sampler Enclave</td>
</tr>
<tr>
<td>BBFPSS26</td>
<td>Tran arrival rate table was initialized</td>
</tr>
<tr>
<td>BBFPSS27</td>
<td>Tran arrival rate table detail, RC= RE= RV=</td>
</tr>
</tbody>
</table>
**Version 4.6.00 message changes**

The following tables list the messages that were added or changed in version 4.6.00:

**Table 6: Messages added with version 4.6.00**

<table>
<thead>
<tr>
<th>ID</th>
<th>Message text</th>
<th>Subsystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBFDXP04</td>
<td>Failed to get program data, reason: reason</td>
<td>IMS</td>
</tr>
<tr>
<td>BBFDXP05</td>
<td>The IMS target is not active: imsid</td>
<td>IMS</td>
</tr>
<tr>
<td>BBFPD223</td>
<td>Action only allowed for buffer pools defined in a DFSDF xx member.</td>
<td>IMS</td>
</tr>
<tr>
<td>BBFPD901</td>
<td>Unable to EDIT: ' xxxxxxx(mmmmmmm)'</td>
<td>IMS</td>
</tr>
<tr>
<td>IM5412E</td>
<td>LOAD FAILED FOR LGBSORT - CHECK STEPLIB</td>
<td>IMS</td>
</tr>
<tr>
<td>PRS242E</td>
<td>IRUF VERSION IS NOT SUPPORTED</td>
<td>IMS</td>
</tr>
</tbody>
</table>

**Table 7: Message changed with version 4.6.00**

<table>
<thead>
<tr>
<th>ID</th>
<th>Message text</th>
<th>Subsystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>IM0202W</td>
<td>MVIMS/EC DD FUNCTION DISABLED - reason RC=returnCode RE=reasonCode</td>
<td>IMS</td>
</tr>
</tbody>
</table>
Version 4.5.00 message changes

The following table lists the messages that were added in version 4.5.00:

Table 8: Messages added with version 4.5.00

<table>
<thead>
<tr>
<th>ID</th>
<th>Message text</th>
<th>Subsystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBFP2514I</td>
<td>Detail trace buffer shortage in the control region</td>
<td>IMS</td>
</tr>
<tr>
<td>BBFP2515I</td>
<td>Trace start request is in progress, no data recorded</td>
<td>IMS</td>
</tr>
<tr>
<td>BBFP2516I</td>
<td>Trace stop request is in progress, no data recorded</td>
<td>IMS</td>
</tr>
<tr>
<td>BBFP2517I</td>
<td>Trace is inactive for the selected unit of work</td>
<td>IMS</td>
</tr>
<tr>
<td>BBFP2518I</td>
<td>No detail trace entries to display or the unit of work has ended</td>
<td>IMS</td>
</tr>
<tr>
<td>BBFP7126</td>
<td>Filters enforced, specify a pattern for the cross-reference</td>
<td>IMS</td>
</tr>
<tr>
<td>BBQQ2110W</td>
<td>cmd request failed, see related messages</td>
<td>IMS</td>
</tr>
<tr>
<td>BBQQ2117W</td>
<td>No trace log data set entries found to delete</td>
<td>IMS</td>
</tr>
<tr>
<td>BBQQ2213W</td>
<td>SET Request failed - xxxxxxxxx</td>
<td>IMS</td>
</tr>
<tr>
<td>BBQZTA02W</td>
<td>TRACE LOG PROCESSING FAILURE - ENTRY STILL IN USE</td>
<td>IMS</td>
</tr>
</tbody>
</table>
Index

A

ABCOUNT parameter 78, 110
activating line commands 28
activating MainView (for DBCTL) 27
activating MainView (for IMS) Offline 27
activating MainView (for IMS) Online 27
AGN field 19
allocating a BBPARM data set 29
AMBLIST service aid utility 118
analyzing dumps 121
AO exit routines
  changing default processing 32
  DFSAOE00 23, 30, 36
  DFSAOUE0 30, 36
  enabling 30
  parameters 23
  removing 14
  return code 32
AOEEXIT parameter 23, 30
AOEINIT parameter 23, 30
AOIEXIT parameter 23, 30, 32
application program CPU timing category 64
archiving trace log data sets 49
ASID field 19
authorization, product 27
AutoOPERATOR. See AO

B

BACKOUT parameter
  description 78
  usage 110
batch message processing data. See BMP data
BBIISPxX member
  trace storage 43, 45
  workload history traces 51
BBIJNT00 member

  subsystem name 23
  target DBCTL systems 28, 107
  target IMS systems 28, 107
BBIISP00 member
  customizing parameters for an IMS 23
  product initialization 27
BBKEYS member 127
BBLINK data set
  deleting modules 18
  dynamic install and uninstall 36
  IMS Event Collector access 29
  product authorization 27
BBPARM data set
  allocation 29
  BBKEYS member 127
  customizing members 23
  setting up 21
BBSAMP data set
  jobs for deleting modules 18
  sample members 123
BBSAMP members
  PRSCEXIT 111
  PRSLEXIT 111
  TACCOSTR 114
BFALTERS option
  control buffer CPU 66
  message buffer CPU 66
BHTO parameter
  control buffer CPU 66
  description 78
  message buffer CPU 66
BILLOVHD parameter
  application program CPU 64
  description 79
  message region overhead CPU 68
BMCPSWD statement 27
BMP
data 80
parameter 23, 80
region, database trailers 94
traces 71
BMP data
  control buffer CPU 66
  control DL/I CPU 65
  message buffer CPU 66
  message DL/I CPU 65
overview 69
BMPTRQC parameter 81
buffer handler activity
  control buffer CPU timing category 66
  message buffer CPU timing category 66
buffer handler timing 78
buffers, detail trace
  buffer size (TRBUFF) 45
  default display size 42
  information level considerations 43
  number of buffers (TRBUFF) 45

C

changing Event Collector parameters 72
CI (customer identification) 112
CICS parameter
  DBCTL threads 71
  description 82
  recommendation 23
CLASS field 19
compatibility
  downward 13
  previous releases 13
  upward 13
components, deactivating 23
control buffer CPU timing category 66
control DL/I CPU timing category 65
control OPEN/CLOSE CPU timing category 67
control region
  Event Collector access 29
  overhead 87
  overhead timing category 68
conventions, documentation 12
CORE parameter 83
CPIC1TRN parameter 83
CPICDB2 parameter 84
CPICDLI parameter 84
CPU
  data collection options 85
  dependent region overhead 79
  parameter 15, 85
  parameter, workload trace data 71
  times 64, 68
  usage, Event Collector 15
  usage, samplers 15
CPU timing categories
  application programming CPU 64
  control buffer CPU 66
  control DL/I CPU 65
  control OPEN/CLOSE CPU 67
  control region overhead 68
  DB2 CPU 66
  message buffer CPU 66
  message DL/I CPU 65
  message OPEN/CLOSE CPU 67
  message region overhead CPU 68
  program scheduling CPU 67
CPU, changing or adding 129
CPUOVHD parameter 87
CQS data query
  refresh rate 57
  time out 56
CQS information views (ISQ*) 57
creating trace log data sets 50
cross-memory mode, user exit routines 110
cross-reference views, limiting records processed 59
CSA (common service area) usage 101
customer ID
  default value 19
  description 19
  IRUF summarization 19
  recommended structure 19
  setting it up 19
customer identification (CI) 112
customizing
  BBPARM members 23
  Event Collector parameters for an IMS 21
  Event Collector user exit routines 109
  system parameters for an IMS 21, 23
  trace facility 41
cut-off time for IMS data 21

D
daily cut-off time for IMS data 21
data collection
Event Collector 23, 61
shared message queues 57
data query
  refresh rate settings 57
time out setting 56
data sets
  ibbparm 21
  BBPARM 21, 23, 29
  UBBPARM 21
database I/O options 91
DB2 CPU timing category 66
DBCTBL3P product authorization table 27
DBCTL target systems, defining 28, 107
DBCTL threads
  controlling collection 82
  monitoring 71
  tracing 71
DBFLVNL parameter 89
DBFP parameter 89
DBFPLVNL parameter 90
DBIO parameter 91
DBTNAME parameter 92
DBTS parameter 93
DBTS4BMP parameter 94
DCIOAL detail trace parameter 42
deactivating MVDBC components 23
defining
  target systems 28, 107
  trace log data sets 49
deleting modules 18
dependent region overhead CPU time 79
dependent regions
  abend at initialization failure 101
  CPU time data collection 15
  detail trace buffer allocation 45
DEPREC parameter
  description 94
  user exits 110
detail trace
  buffer size (TRSIZE) 45
  information level considerations 42
  number of buffers (TRBUFF) 45
DFSAOE00 exit routine 23, 30, 36
DFSAOUE0 exit routine 30, 36
directory trace 41
directory, trace 47
disabling the Event Collector 15
DL/I-CALL-END user exit routine 110
DLISAS region
control buffer CPU 66
control DL/I CPU 65
control region overhead 68
  overhead, CPUOVHD parameter 87
documentation information 11
DTRAC, DB2 and DL/I CPU times 71
DTSQLCPU parameter 95
dump analysis 121
DUMPS parameter 95
dynamic install and uninstall 36

electronic documentation 11
EMIOAL detail trace parameter 42
Energizer (for IMS) Connect, using with MainView
  (for IMS) Online 33
ETIMEELP parameter 96
ETIMEIWT parameter 97
Event Collector
  imsidECP member 21
  imsidSYS member 21
  BMP data collection 69
  CPU timing 64
  CPU usage 15
  customizing parameters 21
  data collection 23, 61
  data collection parameters 63
  disabling 15
  dump analysis 121
  exit routines, customizing 109
  IMS control region access 29
  overview 61
  parameters. See Event Collector parameters
SAPEXIT user exit routine 117
starting and stopping 36
TELON support 103, 119
user exit routines, customizing 109
Event Collector parameters
imsidECP member 21
imsidSYS member 21
BHTO (buffer handler timing) 78
BILLOVHD (dependent reion overhead CPU time) 79
BMP (BMP and JBP data) 80
changing parameters 72
CICS (DBCTL threads) 82
CPU (CPU data collection options) 85
customizing 21
data collection 23, 61
DBIO (database I/O options) 91
DEPREC (extended recovery) 94
option sets 69
specifying 73
EXCSxxx parameters 55
exit routines
AO 14
changing default processing 32
cross-memory mode considerations 110
DFSAOE00 23, 30, 36
DFSAOUFE0 30, 36
Event Collector, customizing 109
IELOAD 32
IMFLEDIT, TELON support 119
IMRUDLI 110
IMRUPGM 109
IMRUTRN 109
IRUF data 112
Log Edit 111
MainView AutoOPERATOR (for IMS) 33
program record 109
return code parameter 32
SAPEXIT 117
summarized IRUF 114
transaction record 109
extended recovery 94

F
FFIOAL detail trace parameter 42
FP ROUTE CODE field 19
FPIOAL detail trace parameter 42
full option set, Event Collector parameters 70

H
Help, online 11

ibbparm data set 21
IBEXSUMR IMFBEX summary view 41, 42, 52, 55, 59
IBINSTL member 37
IBUNINST member 39
ICOPYAOI job 36
ICOPYn jobs 18
identification fields in IRUF. See IRUF identification fields
IELOAD exit routine 32
IIOFTBL3P product authorization table 27
IMFBEX00 member
customizing parameters for an IMS 23
shared message queue data collection 57
trace defaults, specifying 41
trace defaults, updating 45
transaction trace view limits 52
workload threshold options 55
IMFBEX00 member, viewing values in 55
IMFECPO0 member
customizing parameters for an IMS 23
data collection parameters 23, 25
IMFEDP00 member 23
IMFLOGTR member 50
IMFSYS00 member
AOEEXIT parameter 30
AOIEXIT parameter 30
customizing parameters for an IMS 21, 23
deactivating components 23
disable the Event Collector 15
IMFTARC member 49
IMFRLOD member 50
IMRUDLI user exit routine 110
IMRUPGM program record user exit routine 109
IMRUTRN transaction record user exit routine 109
IMS control region
Event Collector access 29
overhead 87
IMS dump analysis 121
IMS target systems, defining 28, 107
IMSID field 19
imsidECP member 21
imsidSYS member 21
initialization messages 131
initializing MainView (for DBCTL) 27
initializing MainView (for IMS) Offline 27
initializing MainView (for IMS) Online 27
input queue time, for message switch transactions 118
install, dynamic 36
installation
  control region JCL 29
dynamic 36
  options 29
IOFBBL3P product authorization table 27
IONBL3P product authorization table 27
IONTBL3P product authorization table 27
IRUF
  creation procedure 21
daily cut-off time 21
  identification fields. See IRUF identification fields
IRUF identification fields
  AGN 19
  ASID 19
  CLASS 19
  CUSTOMER ID 19
  FP ROUTE CODE 19
  IMSID 19
  LTERM 19
  MSC DESTINATION ID 19
  MSC ORIGIN ID 19
  MVIMS SYSID 19
  PERFORMANCE GROUP 19
  PLINE 19
  PROGRAM 19
  PROGRAM TYPE 19
  PTERM 19
  REGION 19
  SMFID 19
  TRANCODE 19
  USERID 19
  VTAM NODE 19
IRUF summarization
  customer ID 19
  process overview 19
  user exit 114
ISQQUERY parameter 57
ISQTIME parameter 56

J

JBP
  BMP parameter 23
data 80
  region, database trailers 94
  traces 71
JBP (Java batch message processing) data
  control buffer CPU 66
  control DL/I CPU 65
  message buffer CPU 66
  message DL/I CPU 65
  overview 69
JCL
  AMBLIST service aid utility 118
  BBPARM data set allocation 29
  dynamic install 37
  dynamic uninstall 39
  IMS Event Collector access 29
  LINKSAP member 117
  product authorization table access 27
  samples 123
  trace log data set printing 50
JXT001 member 47
JXT003 member 48

L

LAST SEG LOST 33
LEXITEPn PRSLEXIT entry points 112
line commands, activating 28
LINKSAP member 117
Log Edit utility parameters 23
logging trace data 41
logspace requirement 33
LSO option
  control buffer CPU 66
  control DL/I CPU 65
  message buffer CPU 66
  message DL/I CPU 65
LTERM field 19
LTRAC traces 71

M

MainView (for IMS Online, DBCTL, and IMS Offline)
BBKEYS authorization 127
MainView AutoOPERATOR (for IMS) dump analysis 121
user exit considerations 33
MainView Customization, data sets 21
MainView Logger logspace requirement 33
MAXDS parameter 98
message buffer CPU timing category 66
message changes 131
message DL/I CPU timing category 65
message level system parameter 23
message OPEN/CLOSE CPU timing category 67
message region overhead CPU timing category 68
minimum option set, Event Collector parameters 71
modifying MVIMS
  response time monitoring 118
  SAP support 117
  TELON support 119
modules, deleting 18
monitoring DBCTL threads 71
MSC clock synchronization 98
MSC DESTINATION ID field 19
MSC ORIGIN ID field 19
MSCCLOCK parameter 98
MSCCLOCK TERM operand 98
MSGLVL1 parameter 23
MSGOQBUF parameter 99
MSGOQTME parameter 100
MTO message capture 23, 30
MVDBC parameter 27
MVIMS parameter 27
MVIMS, SYSID field 19
MXPASRQ parameter 100
MXPASSQ parameter 101

N
NEGATE-RECORD-SW switch 112
NOEC option, IMFSYS00 FEATURE parameter 15

O
online Help 11

P
parameters

AO exit routines 23, 30
AOEEXIT 23, 30
AOEINIT 23, 30
AOIEXIT 23, 30, 32
cross-reference views 59
data collection 63
detail trace I/O area data 42
Event Collector 15
Log Edit 23
option sets 69
product initialization 27
system 21, 23
trace 43
workload threshold 55
XREFLIM 59
PARM parameter, JXT003 member 48
PERFORMANCE GROUP field 19
PLINE field 19
primary commands, SETOPTS set product level options command 59
printing trace log data sets 50
product
  authorization table 27
  authorization, enabling 27
  initialization, enabling messages 131
  initialization, enabling 27
  installation 29, 36
product keys 127
product level options, setting 52, 59
PROGRAM field 19
program record user exit routine (IMRUPGM) 109
program scheduling CPU timing category 67
PROGRAM TYPE field 19
PRSCEXIT
  entry points 112
  user exit routine 112
PRSCEXIT user exit routine
  customer IDs 19
  description 112
  overview 111
PRSLEXIT user exit routine
  LEXITEPn entry points 112
  overview 111
PRSPARM load module 120
PTERM field 19
publications, related 11
RC parameter 32
REGION field 19
related publications 11
RESET command 45, 52, 55
response time monitoring, modifying 118
restoring trace log data sets 50
return code parameter, exit routines 32
RGNIOPT parameter 101

S
sample JCL 123
samplers 15
SAP program support 117
SAPEXIT user exit routine 117
serial number, changing CPU 129
SETOPTS primary command 52, 59
setting up BBPARM data sets 21
shared message queues, controlling data collection 56
SMFID field 19
SMFID parameter 102
SMP-applied modifications
  response time monitoring 118
  TELON support 119
standard option set, Event Collector parameters 69
starting and stopping the Event Collector 36
STORAGE parameter 43
STRAC traces 71
SUBSYS parameter 21, 23, 102
SYNCLOCK TERM operand 98
SYSID parameter 102
SYSINFO utility 127
system parameters, customizing 21, 23

T
target systems, defining 28, 107
TASCOSTR
  BBSAMP member 114
  linkage considerations 114
TASEXIT, TEXITEPn initialization entry points 114
TELON support 103, 119
TEXITEPn, TASEXIT initialization entry points 114
thresholds, workload 55
TIMERR parameter 103
timing categories, CPU. See CPU timing categories trace
  archiving log data sets 49
  buffer display storage size 43
  DBCTL threads 71
  defaults, specifying 41
  defaults, updating 45
  defining log data sets 49
  directory, setting up and maintaining 47
  duration 44
  JXT001 member 47
  JXT003 member 48
  logging options 44
  logging, continuous 51
  managing log data sets 49
  parameters 43
  PARM parameter 48
  printing log data sets 50
  restoring log data sets 50
  transaction trace views, limiting 52
  using IMS logs to create log data sets 50
  verifying directory entries 48
  workload history traces 51
See also detail trace
TRancode field 19
transaction record
  summarization 19
  user exit routine (IMRUTRN) 109
TRBUFF parameter 45
TRCYL parameter 44
TRDATA parameter 42
TRDIR parameter 47
TRMAXRD parameter 52
TRMAXWR parameter 52
TRSYNC parameter 104
TRPREFIX parameter 44, 51
TRREUSE parameter 44, 51
TRSIZES parameter 45
TRSMDSCL parameter 44
TRSMSCCL parameter 44
TRSMSSCL parameter 44
TR_SUFFIX parameter 44
TRTIME parameter 44
TRVOLS parameter 44

U
UBBPARAM data set
imsidECP member 21
imsidSYS member 21
setting up 21
uninstall, dynamic 36
user exit routines. See exit routines
USERID field 19
USEZIIP parameter 23

V
verifying trace directory entries 48
VTAM NODE field 19

W
WATBPRNT member 50

X
XREFLIM parameter, view value of 59

Z
zIIP processing parameters 23
ZIIPWFL parameter 23
ZTIME parameter 105